International Journal of Computer Applications Technology and Research
Volume 6–Issue 4, 172-174, 2017, ISSN:-2319–8656

CPU Performance in Data Migrating from Virtual Machine to Physical Machine in Cloud Computing

Lochan.B
CMR University
Bangalore, India

Dr. Divyashree B A
Dept of CSE,BNMIT
Bangalore, India

Abstract: Cloud computing has a massive use of virtual machines to permit isolated workload to be used from one resource to the another and resource usages to be controlled. Migrating from one operating system to the other operating system is difficult. The virtual machines mainly deals with the live migration process. In this paper, we present the Performance of CPU in Virtual Machine with various features like Cluster, CPU, Live migration, Data Centers, Hosts, Storage, Disks, Templates. The multiprocessor is mainly used in the host machine which allow the features of guest operating system. There are various performance anomalies, which overheads for the infrastructure for the cloud. They are various implication for the results in the future architecture for the cloud infrastructure. Both the container and virtual machine support for the input output intensive application from future cloud allocated to the different application. The large number of the storage and network activity has to served for challenges on the platform. Cloud Computing in the virtual machine has high consumption of memory and CPU resource for inefficient virtualization software.

Keywords: Live Migration, Cluster, Virtualization, Host, Data Centers

1. INTRODUCTION

Virtual Memory are used massively in the cloud computing. It performs the platform like Hypervisor (Hyper-V) that the virtual machines are available for the customers to run the service like Platform as a Service(Paas) that make the available for customer to run inside the workloads running inside the virtual memory[1]. It has a various challenge for preventing and diagnose the performance anomalies in virtualized environment. The Application running inside black box will have infrastructure as the environment. Cloud Management has to work automatically that prevents the performance anomaly for the management for Virtualized Cloud System. There are two main objective to achieve the system level metrics i) A performance anomaly that can have raise detection alerts. ii) The system metrics that cause inference grained faculty virtual machine to the performance anomaly [2]. Virtual Machine in the multiprocessor allows the guest operating system for the use of the symmetric multiprocessor that are used in the host machine. In these, it mainly consists of virtual machine monitor and performance of the overhead of the virtualization. It has various types of workloads for the physical host resources for the encapsulated in a virtual machine run on a single physical host. They are various Virtualization and Symmetric multiprogramming technologies are having opposite goals. Some of the example for Software Virtualization are Microsoft Hyper-V, Xen, VirtualBox, VMware ESX/ESXi. In Virtualization multiple operating system are done in concurrently as shown in Fig.1.

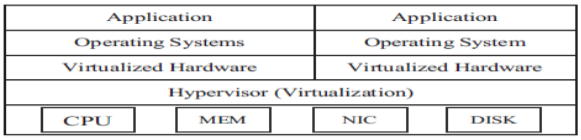

Fig. 1. Virtualization

Several techniques are mainly used for the reduce of virtual machine monitor intervention. The multiple operation system has to share a single utilization of system resources like Input/Output device. There are many approaches are used for the enabling virtualization. Full Virtualization and Para Virtualization. Full Virtualization is mainly used for modifications that are not used in the hardware support such as technology used for the virtualization. Para Virtualization has operating system to work with modification of virtual machine monitor. Input or output needs some effectively handling virtualization to be improved for the scalability such as guest access for the I/O devices. The virtual machine used to overhead the saturate the high throughput situation such as virtual machine. Single Root Virtualization has Direct I/O virtualization for the major device resource to achieve both sharing and high performance[3].

2. BACKGROUND

2.1 Requirement and Motivation for Cloud Virtualization:

Every user and program should operate using the privileges to complete the job. The least Common mechanism has strongly implement the principles of least privileges. Shared library can be especially have modern application require different version of the library. The requirements are mainly used to have a resource isolation for suitable infrastructure or configuration between application. It improves the I/O concurrency for enhanced storage

performance. It has hardware support for unit virtualization for memory

2.2 Scientific Computing:

The Scientific Computing has heterogeneous CPU, Memory and Network as the bottleneck resource. The average of the Scientific Cluster is found to be around many nodes to be stable over the past many years. The Network, Memory, I/O, CPU can have Heterogeneous as the bottleneck resources. The Workloads are in scientific and users in performance of cloud computing service. It mainly depends on the performance in which the focus of the work to be carried out[4]. The Characteristics Performance of four commercial cloud computing service has been executing parallel application of up to many processor.

2.3 Live Migration

It is the process of moving a guest virtual machine from one host physical machine to another in Fig.2. It is mainly used for load balancing it can be moved from host physical machine with lower usage the physical machine it has overload to add, upgrade or remove hardware device in the host physical machine it can have an hardware independence in the hardware improvements. Energy Saving helps for guest virtual machine should be powered off to save energy and low cost in usage of periods. Geographic Migration is the guest virtual machine that can be moved from one location to the lower latency or in serious circumstances. The main features is to work for sending the state of the guest virtualized device or the machine memory it has to be migrated. It is also recommended to use live migration from migration from virtual machine[5].

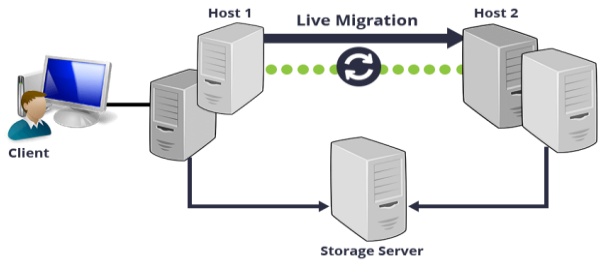

Fig2: Live migration from one host to the another host

3. SYSTEM DESIGN

Hypervisor has its own policy for what a virtual machine have its CPU by default. It has to decide the some hypervisor which CPU host Physical machine for the available for the guest virtual machine. The host has its own filtering, Classifying the Physical CPU model for each group has its own virtual machine[6]. It has a safe migration between the host physical machine that provided the physical CPU that classify the same group for a suitable guest virtual machine model is successful between host physical machine. It has to emulate features that is aware of the features that were created after the hypervisor was released may not. It has the grouping and allocation of resources on per guest virtual machine[7].

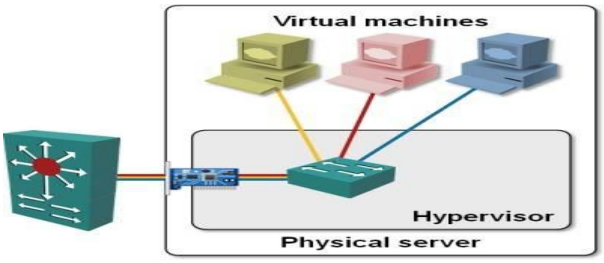

Fig.3 : Live migration networking

The System Administrator to in the either turntables against the guest operating system. It mainly consists of i) Memory: The memory controller allows for setting limits on RAM and swap usage and querying cumulative usage of all processes in the group. ii) CPU**:** The CPU scheduler controller controls the prioritization of processes in the group. It is similar to grant level privileges. iii) Freezer: The freezer controller pauses and resumes execution of processes in the group[8]. This is similar to SIGSTOP for the whole group. iii) Devices **:** The devices controller grants access control lists on character and block devices in fig.3

4. EVALUATION

The numerous aspects can be done in the performance issue on the non-virtualized execution because it has various resources available for the productive work. When one or more resource are fully utilized and have a workload metrics like throughput and the latency for the overhead virtualization[9]. All the test are performed on the system with 8GB RAM. The two system are needed for calculating the performance of live migration from one operating system to the other operating system[10]. In Virtualization technology, while deploying it must ensure that Operating System and physical machine cannot be compromised. The guest Virtual Machine, Network, Memory, Devices system have security on system using virtualization using the deployment plan. There are various works are carried for the state the guest virtual machine's memory and device for a destination from host physical machine. The main features is to use shared, networked storage for the guest virtual machine to be migrated. In these shared storage, while migration can be performed live or not. In a live migration, the guest virtual machine continue to run in the host physical machine for which it can have memory pages from the host physical machine its destination host physical machine[11]. During migration the source for any changes it can be already transferred for the initial pages to be transferred. A migration is not performed live, it has to suspends for the already transferred and to transfer the initial pages to been transferred. If the network is experiencing the use or low bandwidth, the migration will take more time. The destination host physical machine it has the offline migration to be used as the live migration will be complete such as migration depends on the network bandwidth and latency[12].

System	PowerEdge servers
CPU	2*Quad Core Intel
Memory	8 GB RAM
RAM Per VM	1.5 GB
Operating System	Hyper-V
Disk Drive	4*146 GB 10,000 RPM

Table 1: System Configuration

4.1 HARDWARE

Table 1. shows the system configuration for client hardware as Single-socket, quad core server,2.50GHZ Intel E420 Processor. 8GB Memory. All Experiments are conducted in the number of System for a CPU Virtual Machine having the number of pCPUs used equals to the number of vCPUs configured in the virtual machine[13]. Fig.4 shows RHEV-H Networking configuration. Most of the virtualization platforms has interface that allows the guest virtual machine for the hypervisor as guest virtual machine system level metric from the usage from domain0[14].

4.2 SOFTWARE

The software is mainly used as Virtual Box. In these, Hypervisor is mainly used. The scripting language is used i.e python.

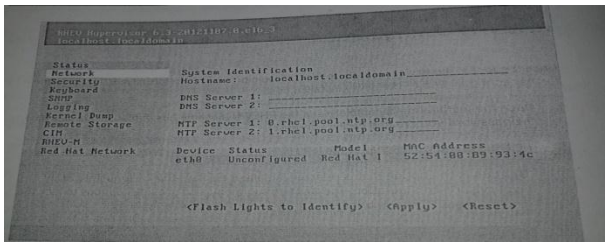

Fig4. RHEV-H Networking Configuration

5. CONCLUSION AND FUTURE WORK

Virtualization is important technology because it has more efficient use of resource. It differs from a physical system that leads to degradation for application that is running on a virtual Machine. The Performance of the data migration from one operating system to the other operating system to carried out is difficult. The virtual machine has the performance of input output to negligible when it is has a large rates. It contains the tradeoff started with the overhead of high packet rates. It represent a tradeoff between of management and performance should be case-by-case basis. It has the limited of each workload to single socket performance of turning and analysis. This leads to several isolation of multiple workload on the same server, live resizing of containers and virtual machine of scale out and tradeoffs between restarting and live migration. The innovation of the Virtualization has to bring the performance of the more data in the Para Virtualization in the future.

REFERENCES

[1] Wes Felter, Alexandre Ferreira, Ram Rajamony, Juan Rubio by IBM Research, Austin, Tx by " An updated Performance Comparison of Virtual machine and Linux Containers" Vol.4 2014

[2] Yongmin Tan, Hiep Nguyen, Zhiming Shen, Xiahui Gu North Crohina State Univerity "Prepare:Predictive Performance Anomaly Prevention for Virtualized Cloud System" page 241-254 2013

[3] Yaozu Dong, Xiaowei Yang,Xiaoyong by intel china software center "High Performance Network Virtualization with SR-IOV "

[4] Alexandru Losup "Performance Analysis of Cloud Computing Services for many tasks scientific computing"

[5]http://www.vmware.com/pdf/virtualization.pdf "Virtualization Overview"

[6] Advanced multi layered unification file system. http://aufs.sourceforgenet,2014

[7] "Virtual computing lab," http://vcl.ncsu.edu/.

[8] Ari Balogh, Google Compute Engine is now generally available with expanded OS support, transparent maintenance and lower price. http://googledevelopers.blogsport.com /2013/12/ google-compute-engine-is-now-generally.html Dec-2013

[9]Jerome H.Saltzer and Micheal D.Schroeder The protection of information in computer system. In proceeding of the IEEE,Vol 63 Sep 2013

[10] "Amazon Elastic Compute Cloud," http://aws. amazon. com/ec2/

[11] "Rubis Online Auction System,"http://rubis.ow2.org/.

[12] C.Wang, V.Talwar, K.Schwan and P.Ranganathan, "online detection of utility cloud anomalies using metric distribution," in proc. of NOMS,2010

[13] D.Chisnall, "The definition guide to the XEN"

[14] G.J.Popek and R.P.Goldberg, Formal requirements for virtualizable third generation architecture communications of the ACM-July 1974

International Journal of Computer Applications Technology and Research
Volume 6–Issue 4, 175-182, 2017, ISSN:-2319–8656

Software Quality Measure

Eke B. O.
Department of Computer Science
University of Port Harcourt
Port Harcourt, Nigeria

Musa M. O.
Department of Computer Science
University of Port Harcourt
Port Harcourt, Nigeria

Abstract: Modern gadgets and machines such as medical equipments, mobile phones, cars and even military hardware run on software. The operational efficiency and accuracy of these machines are critical to life and the well being of modern civilization. When the software powering these machines fail it exposes life to danger and can cause the failure of businesses. In this paper, software quality measure is presented with the emphasis on improving standard and controlling damages that may result from badly developed application. The research shows various software quality standards and quality metrics and how they can be applied. The application of the metrics in measuring software quality in the research produced results which shows that the code metrics performance is better than the design metrics performance and points to a new way of improving quality by refactoring application code instead of developing new designs. This is believed to ensure reusability and reduced failure rate when software is developed.

Keywords: Software, quality, reusability, metrics, measure

1. INTRODUCTION

Software quality measures how well software is designed (*quality of design*), and how well the software conforms to that design (*quality of conformance*), although there are several different definitions. It is often described as the 'fitness for use for the purpose' of developing a piece of software. Whereas *quality of conformance* is concerned with implementation, *quality of design* measures how valid the design and requirements are in creating a worthwhile product. But what exactly is software quality? It's not an easy question to answer, since the concept means different things to different people.

Software quality may be defined as the degree of conformance to explicitly stated functional and performance requirements, explicitly documented development standards and implicit characteristics that are expected of all professionally developed software (Ho-Won, et al. 2014). In the definition, it is clear that software requirements are the foundations from which quality is measured. It is then believed that lack of conformance to requirement is lack of quality. Specified standards define a set of development criteria that guide the management of software engineering. Hence, if criteria are not followed during software development, lack of quality will usually result.

A set of implicit requirements often goes unmentioned, for example ease of use, maintainability, usability and other software quality concerns. If software confirms to its explicit (clearly defined and documented) requirement but fails to meet implicit (not clearly defined and documented, but indirectly suggested) requirements, software quality is suspected.

As with any definition, the definition of 'software quality' is also varied and debatable. Some even say that 'quality' cannot be defined and some say that it can be defined but only in a particular context. Some even state confidently that 'quality is lack of bugs'. Whatever the definition, it is true that quality is something we all aspire to have when developing software .

The Institute of Electrical and Electronics Engineers (IEEE) defines software quality as ***t***he degree to which a system, component, or process meets specified requirements and the degree to which a system, component, or process meets customer or user needs or expectations.

Similarly, International Software Testing Qualifications Board (ISTQB) defines software quality as the degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations. The totality of functionality and features of a software product that bear on its ability to satisfy stated or implied needs (Stephen, 2012).

2. SOFTWARE DEVLOPMET LIFE CYCLE

When developing software of high quality, it is crucial to have a good understanding and knowledge of the various phases or stages of Software Development Life Cycle (SDLC). Software Development Life Cycle, or Software Development Process, defines the steps/ stages/ phases in the building of quality software (McConnell, 2015).

There are various kinds of software development models like:

i) Waterfall model
ii) Spiral model
iii) Iterative and incremental development (like 'Unified Process' and 'Rational Unified Process')
iv) Agile development (like 'Extreme Programming' and 'Scrum')

Models are evolving with time and the development life cycle can vary significantly from one model to the other. However, each model comprises of all or some of the core phases/ activities/ tasks involved in software development.

2.1 The Basic Model of SDLC

The basic model of the Software development Life Cycle starts out with the requirement analysis and moves into the design phase, the implementation phase, testing phase, the release phase and cycles back to the requirement phase (Scott, 2005).

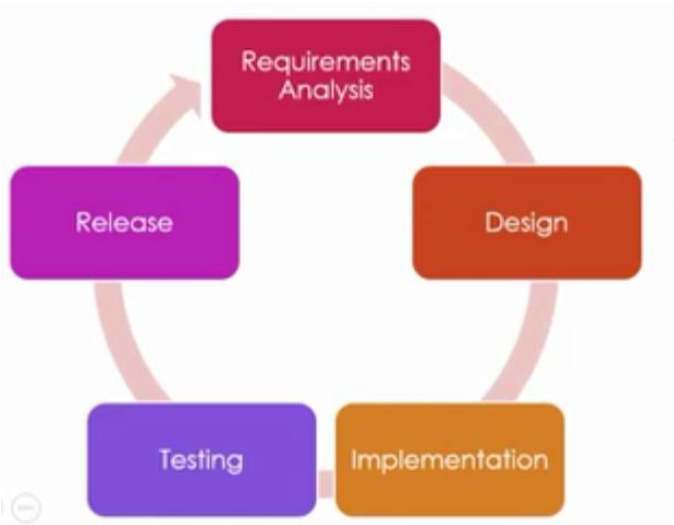

Figure 1: SDLC Basic Model

The phases specified in figure 1 is basic and its arrangement may vary from one methodology to another, however the activities carried out in the phases are similar.

Activities in the life cycle:

Requirement : In requirement activity, developers work directly with customer(s) and identifies the problem to be solved. It focuses on "what" the software is intended to do and not "how". It is important to note that often what a client or customer actually need is often not very clearly expressed and often vary within the development period.

Analysis and Design: This activities focuses on how the programme will achieve the software requirements. The activities start from analysis by making sure that the problem is broken down into smaller pieces called components and then the design is carried out when components are used in synthesizing the system to work together to make the whole programme work.

Implementation : In this phase the code is written according to design specifications. The implementation language may be a barrier to the development of the system. The developers must select a programming language that will be able to handle all of the concerns in requirement captured in the design. This is important due to the fact that certain compiler restrictions or implementation may not allow easy development of certain components according to specification. The selection of programming language of development is therefore an important consideration when quality of application is considered at the implementation stage of software development life cycle. Some time if a reusable software component is available it is preferable to reuse it if it had been previously tested to be working efficiently.

Testing: During the testing activity the code or the design is verified by using different ways in checking if the design or the code met certain the design specification or code functional specification. It is a strong view held by software engineers that if proper testing is carried out at the design stage of a software development life cycle then the coding testing will only be a confirmatory test that the system is working properly as expected. This view have been researched upon to even check whether it is better to carry-out design testing before code testing and which of the two is capable of revealing development error. A similar verification was carried out by Jiang in one of the researches (Jiang et. al., 2007).

Release: When software is released certain concern and requirement may be omitted by the developer or the customer. When the requirement is omitted by the customer it may be released in the next version of the software and it is often not held as a quality issues rather it is an upgrade issue. However, when the requirement omission is from the developer it is a serious quality concern issue. It is at the release phase that the software is closely examined by the staff of the customer(s) and validates that the programme meets the customer's expectations.

There may still be many other activities/ tasks which have not been specifically mentioned above depending on the software design methodology. But it is essential that the key activities within a software development life cycle be understood even if it is at a review level.

2.2 Who Cares About Software Quality?

With software or anything else, assessing quality means measuring value. Something of higher quality has more value than something that's of lower quality (IOS, 2010). Yet measuring value requires answering another question: value to whom? In thinking about software quality, it's useful to focus on three groups of people who care about its value, as Figure 2 shows.

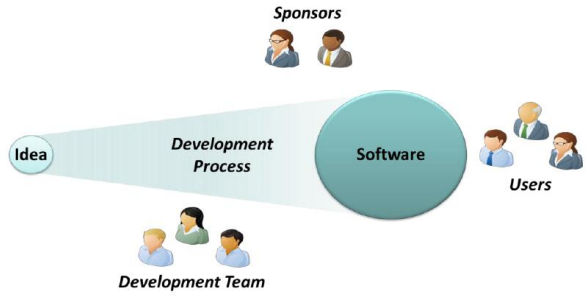

Figure 2: Those who care about software quality

As the figure illustrates, a development process converts an idea into usable software. The three groups of people who care about the software's quality during and after this process are:

1. The software's *users*, who apply this software to some problem.
2. The *development team* that creates the software.
3. The *sponsors* of the project, who are the people paying for the software's creation. For software developed by an organization for its own use, for example, these sponsors are commonly business people within that organization.

All three of these groups are stakeholders of software quality. The aspects of quality that each finds most important aren't the same, however. Understanding these differences requires dissecting software quality to really see the detail structure.

3. ANALYSIS OF SOFTWARE QUALITY

Analysis involves the decomposition of the system into its component parts to identify the part that can be combined in forming a new system. Hence it is useful to think about the software quality by dividing it into three aspects: functional quality, structural quality, and process quality. Doing this helps us see the big picture, and it also helps clarify the trade-offs that need to be made among competing goals (Basili, et, al.,1996). . Figure 4 illustrates this idea.

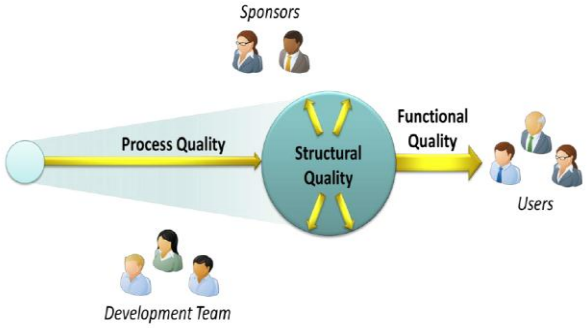

Fig.3: Software quality decomposed into three aspects: functional quality, structural quality, and process quality.

The three aspects of software quality are *functional* quality, *structural* quality, and *process* quality.

3.1 Functional Quality

Functional quality reflects how well the software complies with or conforms to a given design, based on functional requirements or specifications. This attribute also ensures that the software correctly performs the tasks it's intended to do for its users. Among the attributes of functional quality are:

1. ***Meeting the specified requirements.*** Whether they come from the project's sponsors or the software's intended users, meeting requirements is the *sine qua non* of functional quality. In some cases, this might even include compliance with applicable laws and regulations. Requirements commonly change throughout the development process, achieving this goal requires the development team to understand and implement the correct requirements throughout, not just those initially defined for the project.
2. ***Creating software that has few defects.*** Among these are bugs that reduce the software's reliability, compromise its security, or limit its functionality. Achieving zero defects is too much to ask from most projects, but users are rarely happy with software they perceive as buggy.
3. ***Good enough performance.*** Users often perceive slow system as not been well designed or to be outrightly a bad software. The thing that may be causing the low performance might be very simple but that is not actually what the user sees. It is the end product of the software that the user interacts with.
4. ***Ease of learning and ease of use.*** To its users, the software's user interface *is* the application, and so these attributes of functional quality are most commonly provided by an effective interface and a well-thought-out user workflow. The aesthetics of the interface—how beautiful it is—can also be important, especially in consumer applications.

Software testing commonly focuses on functional quality. All the characteristics just listed can be tested, at least to some degree, and so a large part of ensuring functional quality boils down to testing.

3.2 Structural Quality

The second aspect of software quality, structural quality, means that the code itself is well structured. Unlike functional quality, structural quality is hard to test for (although there are tools to help measure it) (Robert, 1992). The attributes of this type of quality include:

1. ***Code testability.*** Checking if the developed code is organized in a way that makes testing easy or whether testing the code will be fell based on the style of code development.

2. ***Code maintainability*.** High level modularity is also checked to make sure that it is easy to add new code or change existing code without introducing bugs in other part of the program.
3. ***Code understandability*.** Is the code readable? Is it more complex than it needs to be? These have a large impact on how quickly new developers can begin working with an existing code base.
4. ***Code efficiency*.** It also check if the program consumes a lot of system resources in execution, and writing efficient code can be critically important in making the application to execute in old and newer machines. Users often do not need to upgrade their hardware or to buy new system just to be able to run a program, when similar app can also run in their machine.
5. ***Code security*.** Does the software allow common attacks such as buffer overruns and SQL injection? Is it insecure in other ways?

3.3 Process Quality

Process quality, is also critically important. The quality of the development process significantly affects the value received by users, development teams, and sponsors, and so all three groups have a stake in improving this aspect of software quality(Antoniol, et, al.,2002)..

The most obvious attributes of process quality include these:

1. ***Meeting delivery dates*.** Was the software delivered on time?
2. ***Meeting budgets*.** Was the software delivered for the expected amount of money?
3. ***A repeatable development process that reliably delivers quality software*.** If a process has the first two attributes—software delivered on time and on budget—but so stresses the development team that its best members quit, it isn't a quality process. True process quality means being consistent from one project to the next.

4. SOFTWARE QUALITY ASSURANCE

Software Quality Assurance (SQA) is a set of activities for ensuring quality in software engineering processes (that ultimately result in quality in software products). These activities include:

Process definition and implementation, Auditing, and Training

Processes could be:

1. Software Development Methodology
2. Project Management
3. Configuration Management
4. Requirements Development/Management
5. Estimation
6. Software Design
7. Testing, etc.

Once the processes have been defined and implemented, Quality Assurance has the following responsibilities:

1. identify weaknesses in the processes
2. correct weakness to continually improve the process

The quality management system under which the software system is created is normally based on one or more of the following models/standards which are the most popular models:

1. CMMI 2. Six Sigma 3. ISO 9000

There are many other models/standards for quality management but the ones mentioned above are the most popular. Software Quality Assurance encompasses the entire software development life cycle and the goal is to ensure that the development and/or maintenance processes are continuously improved to produce products that meet specifications/requirements. The process of Software Quality Control (SQC) is also governed by Software Quality Assurance (SQA). SQA is generally shortened to just QA.

4.1 Software Quality Control

Software Quality Control (SQC) is a set of activities carried out to ensure quality in software products (Antoniol, et, al.,2001).

It includes the following activities:

i) *Reviews:* The review of the activities carried out must be done at all stages of the life cycle based on the methodology selected for the system development. In the sample model we are using in this paper it may include:

1. Requirement Review: Review carried out when the initial requirements have been done, to check if all the requirements needed in the system are captured.
2. Design Review: When the design of the system is completed, the review re-examine the design to see if there are certain omissions that needed to be corrected.
3. Code Review: This involve the checking of the coding pattern to see if it satisfies the principles required for quality program.
4. Deployment Plan Review : The review is carried out to make sure that there are no omissions in the plans for the deployment of the system.
5. Test Cases and Test Plan Review involve the checking of the test conditions required to execute the system.

ii). *Testing:* Testing varies from one methodology to another but one issue is common to them all, which is that testing need to be done. Some methodology reserve a specific time for testing phase while other encourage progressive testing throughout the life cycle. Whichever process that is used some of the testing carried out include:

1. Unit Testing: This involve the testing of single modules or program units and to make sure that it is working according to the expected goal.
2. Integration Testing: Once the units are working according to the expectation they can be brought

together and tested to make sure they are working well as a whole unit.

3. System Testing: once the entire system is ready for deployment it can still be tested with varying example data to make sure that various input data will work up to the expectation of the system.
4. Acceptance Testing: In this stage the customers can use the real life data set to test the system before it is finally deployed for usage.

Software Quality Control is limited to the Review/Testing phases of the Software Development Life Cycle and the goal is to ensure that the products meet specifications/requirements. The process of Software Quality Control (SQC) is governed by Software Quality Assurance (SQA). While SQA is oriented towards prevention, SQC is oriented towards detection. Some people assume that QC means just Testing and fail to consider Reviews; this should be discouraged (Schröter, et al., 2006).

Differences between Software Quality Assurance (SQA) and Software Quality Control (SQC)

Criteria	**Software Quality Assurance (SQA)**	**Software Quality Control (SQC)**
Definition	SQA is a set of activities for ensuring quality in software engineering processes (that ultimately result in quality in software products). The activities establish and evaluate the processes that produce products.	SQC is a set of activities for ensuring quality in software products. The activities focus on identifying defects in the actual products produced.
Focus	Process focused	Product focused
Orientation	Prevention oriented	Detection oriented
Breadth	Organization wide	Product/project specific
Scope	Relates to all products that will ever be created by a process	Relates to specific product
Activities	i) Process Definition and Implementation ii) Audits iii) Training	i) Reviews ii) Testing

5. SOFTWARE TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Statistics had been used over the year in test (Siegel, 1956) and it is still been used in certain parameter test even in software metrics. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest (Diomidis, 2006).. In general, these properties indicate the extent to which the component or system under test:

1. meets the requirements that guided its design and development,
2. responds correctly to all kinds of inputs,
iii) performs its functions within an acceptable time,
iv) is sufficiently usable,
v) can be installed and run in its intended environments, and
vi) achieves the general result its stakeholders desire.

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding software bugs (errors or other defects). The job of testing is an iterative process as when one bug is fixed, it can illuminate other, deeper bugs, or can even create new ones.

5.1 Software Testing Levels

There are four levels of software testing: *Unit >>Integration >>System >>Acceptance*.

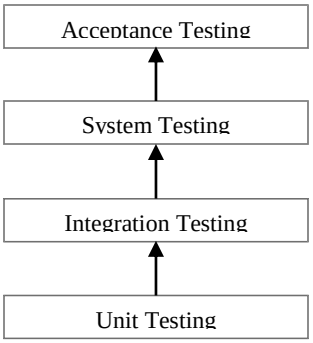

Fig. 4: A Software Testing level

1. Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.
2. Integration Testing is a level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.
3. System Testing is a level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.
4. Acceptance Testing is a level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Note: Some tend to include Regression Testing as a separate level of software testing but that is a misconception. Regression Testing is, in fact, just a type of testing that can be performed at any of the four main levels.

5.2 Techniques of Software Testing

Below are some methods / techniques of software testing:

1. Black Box Testing is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. Test design techniques include: *Equivalence partitioning, Boundary Value Analysis, Cause Effect Graphing.*
2. White Box Testing is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. Test design techniques include: *Control flow testing, Data flow testing, Branch testing, Path testing.*
3. Gray Box Testing is a software testing method which is a combination of Black Box Testing method and White box Testing method.
4. Agile Testing is a method of software testing that follows the principles of agile software development.
5. Ad Hoc Testing is a method of software testing without any planning and documentation.

6. SOFTWARE ENGINEERING STANDARDS TEST MEASURE

According to the IEEE Comp. Soc. Software Engineering Standards Committee, a standard can be: An object or measure of comparison that defines or represents the magnitude of a unit. It can also be a characterization that establishes allowable tolerances or constraints for categories of items, or a degree or level of required excellence or attainment.

6.1 Software Standards Legal Implications

Comparatively few software products are forced by law to comply with specific standards, and most have comprehensive non-warranty disclaimers. However, for particularly sensitive applications (e.g. safety critical) software will have to meet certain standards before purchase.

1. Adherence to standards is a strong defence against negligence claims (admissible in court in most US states).
2. There are instances of faults in products being traced back to faults in standards, so
3. Standards writers must themselves be vigilant against malpractice suits.

When standards are released, it is also important to subject the so call standard to QA testing to make sure that serious fault will not arise by adhering to those standards.

6.2 Quality Assurance Standards

Differing views of quality standards: taking a systems view (that good management systems yield high quality); and taking an analytical view (that good measurement frameworks yield high quality). Examples:

1. Quality management: ISO 9000-3 Quality Management and Quality Assurance Standards - Part 3: Guidelines for the application of 9001 to the development, supply, installation and maintenance of computer software
2. Quality measurement: IEEE Std 1061-1992 Standard for Software Quality Metrics Methodology

6.2.1 Product Standards

These focuses on the products of software engineering, rather than on the processes used to obtain them. Perhaps surprisingly, product standards seem difficult to obtain. Examples:

1. Product evaluation: ISO/IEC 14598 Software product evaluation
2. Packaging: ISO/IEC 12119:1994 Software Packages - Quality Requirements and Testing

6.2.2 Process Standards

A popular focus of standardization, partly because product standardization is elusive and partly because much has been gained by refining process. Much of software engineering is in fact the study of process. Examples:

1. Life cycle: ISO/IEC 12207:1995 Information Technology - Software Life Cycle Processes
2. Acquisition: ISO/IEC 15026 System and software Integrity Levels
3. Maintenance: IEEE Std 1219-1992 Standard for Software Maintenance
4. Productivity: IEE Std 1045-1992 Standard for Software Productivity Metrics.

7 EXPERIMENTAL USE CASE

In the experimental system a project program was used to examine the concepts provided in this paper to empirically find out what will be the result using a given set of data and a varying metrics.

Condition1 : i) Model development using design metrics
ii) Model development using design metrics only

Condition 2: i) Testing the models with data set utilizing code metrics
ii) Testing models with data set utilizing design metrics

Data set: The data set was randomly generated so that various type of data will be covered and the data that may be considered none applicable will also be tested. When the data is not within area that the system should handle the system need to graciously handle such challenge without an outright crash.

Metrics Used: The metrics used include selected design metrics and selected code metrics.

Design Metrics: The design metrics used in the experimentation (Subramanyam et,al., 2003) include: The design complexity of a module, Design_Density, Essential_Complexity Module, Essential_Density and Maintenance_Severity. All the design metrics were calculated as a factor of cyclomatic complexity of a module ($e - n + 2$) where n could be Number of calls to other functions in a module and e effort matrics of the module.

Code Metrics: The design metrics used in the experimentation include: The halstead length content of a module $\mu = \mu 1 + \mu 2$,
The halstead length metric of a module $N = N1 + N2$,
The halstead level metric of a module $L = (2*\mu_2) /(\mu_2*N_2)$
The halstead difficulty metric of a module $D = 1/L$
The halstead volume metric of a module $V = N * \log2(\mu 1 + \mu 2)$
The halstead effort metric of a module $E = V/L$
The halstead programming time metric of a module $T = E/18$
The halstead error estimate metric of a module B = $E^{2/3}/1000$

Method: In the experimental use case six specific instruments or programs are selected for quality examination and the design metrics as well as the code metrics was used to test the outcome using the various data set. A fault tool checker was also deployed to compare the result from the metrics to the result from the tool checker is statistically obtained via the internal system of the tool and the percentage fault was also extracted.

7.1 Results

The result of the quality test is clearly displayed on the table. The result show a note of the specific intrument of program module used in the test. It also show the performance of the metrics using their fault levels. The instruments are different and that variation is also reflected in the data sets used in testing the system table 1 clearly illustrate all these values.

Table 1: Result of percentage fault from the tests

Data set	Test No	% Fault		Note
		Design metrics	Code metrics	Specific Instrument
DT1	0001	2.1	0.7	Simple number computation
DT2	0002	5.4	1.2	Input and output processing
DT3	0003	2.5	0.9	A database system
RD1	0004	14.2	7.3	A combustion experiment
RD2	0005	7.8	3.2	Multimedia system
RD3	0006	16.3	2.8	A Recursive procedure

In figure 5 it is clear from the graph the percentage fault of each of the metrics used. The design metrics performance of the code metrics appears to be better than the performance of the *design* metric groups. In the 6 data sets, the instruments where clearly varied from simple input-output processing which is a very simple program that can be easily tested for performance to recursive procedures which if not controlled could result to a forever executing system that can exhaust the processing resources and memory of the system if continuous output is generated.

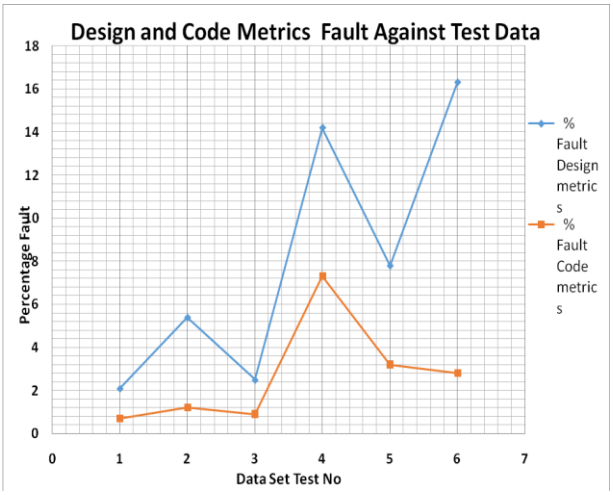

Fig. 5: A plot of the testing result

The fault was lowest on simple number computation which is understandable but on the contrary instead of recursive procedure showing the highest fault level on the code metrics it was the combustion engine classical program that was not written in a highly modular format that showed the highest fault level for the code metrics. It is clear therefore that it is not only the metrics that are contributing to the fault that the process type used in the development of the system that also contribute a great part to the system efficiency.

8.CONCLUSION

Organizations that develop low-quality software, whether for internal use or for sale, are always looking backward, spending time and money on fixing defects in "finished" products. In contrast, an organization that builds in product quality from the beginning can be forward-looking and innovative; it can spend its resources on pursuing new innovations instead of spending the time on maintenance.

In the use case test it is clear that the instrument (type of problem solved by a program) alone does not determine the performance of the system. The process or methodologies used clearly contribute much in the performance; a poor process can result to bad program both at design and implementation. The benefits of including quality-oriented activities in all phases of a software development lifecycle are both broad and deep. These measures not only facilitate innovation and lower costs by increasing predictability, reducing risk, and eliminating rework, but they can also help to differentiate an quality product from its competitors. Most important, continuously ensuring quality will always cost less than ignoring quality considerations.

9.RECOMMENDATION

In most of the instruments, it is clear that those that have very bad fault at design also had corresponding higher fault at implementation. The paper did not correlate the two but from the plot of the result the relationship of the two plot is obvious. It is therefore recommended the a combined effort at improving both design fault and coding fault can be a target that can be easily realizable if good process and programming practice is imbibed. Further research is also recommended to find out correlation between the metrics to see the effect or level of fault relationship. This will enable a valuable discuss on the regression test of the design fault with the coding fault. This work is recommended as launch pad to such research so that the quality issues raised and discussed in this work will be used in handling such cases.

ACKNOWLEDGEMENT

Oyol Computer Consult Inc Port Harcourt, Nigeria is acknowledged for providing the facility and tool used in carrying out the experimental tests. We also thank them for offering some of the instruments used.

REFERENCES

Ho-Won J., Seung-Gweon K., and Chang-Sin C. (2014). Measuring software product quality: A survey of ISO/IEC 9126. *IEEE Software*, 21(5):10–13, September/October 2014.

Stephen H. K (2012). *Metrics and Models in Software Quality Engineering*. Addison-Wesley, Boston, MA, second edition.

McConnell, S. (2015), *Code Complete* (Fifth ed.), Microsoft Press Pressman,

Scott M. (2005), *Software Engineering: A Practitioner's Approach* (Sixth, International ed.), McGraw-Hill Education

Jiang Y., Cukic, B. and Menzies T. (2007) Fault prediction using early lifecycle data. pages 237–246. Software Reliability. ISSRE ’07. The 18th IEEE International Symposium on, Nov. 2007.

International Organization for Standardization.(2010) *Software Engineering—Product Quality—Part 1: Quality Model*. ISO, Geneva, Switzerland, 2010. ISO/IEC 9126-1:2010(E).

Antoniol, G. Canfora, G. Casazza, A. D. Lucia, and E. Merlo.(2002) Recovering tracebility links between code and documentation. *IEEE Transactions on Software Engineering*, 28(10):970–983.

Antoniol, G., Casazza, G., Penta, M. and Fiutem, R. (2001) Object-oriented design patterns recovery. *Journal of Systems and Software*, 59(2):181–196.

Basili, V. R., Briand L. C. and Melo W. L. (1996). A validation of object-oriented design metrics as quality indicators, 1996.

Breiman. L (2001) Random forests. *Machine Learning*, 45:5– 32, 2001.

Diomidis S.(2006). *Code Quality: The Open Source Perspective*. Addison Wesley, Boston, MA, 2006.

Robert L. Glass.(1992) *Building Quality Software*. Prentice Hall, Upper Saddle River, NJ, 1992.

Roland Petrasch, (1999) "The Definition of, Software Quality’: A Practical Approach", ISSRE, 1999

Schröter, A, Zimmermann, T and Zeller A. (2006) Predicting component failures at design time. In *ISESE ’06: Proceedings of the 2006 ACM/IEEE international symposium on International symposium on empirical software engineering*, pages 18–27, New York, NY, USA, 2006. ACM Press.

Siegel. S. (1956) *Nonparametric Satistics*. New York: McGraw- Hill Book Company, Inc., 1956.

Subramanyam R. and M. S. Krishnan. M S. (2003) Empirical analysis of ck metrics for object-oriented design complexity: Implications for software defects. *IEEE Trans. Softw. Eng.*, 29(4):297–310,

International Journal of Computer Applications Technology and Research
Volume 6–Issue 4, 183-189, 2017, ISSN:-2319–8656

An Approach to Face Recognition Using Feed Forward Neural Network

Sajetha T
Dept. of CSE, UVCE,
Bangalore University,
Bangalore, India

Samyama Gunjal G H
Dept. of CSE, UVCE,
Bangalore University,
Bangalore, India

Abstract: Many approaches have been proposed for face recognition but there are major constraints like illumination, lightning, pose etc., when taken into consideration, results in poor recognition rate. We propose a method to improve the recognition rate of the face recognition system which uses various methods like homogeneity, energy, covariance, contrast, asymmetry, correlation, mean, standard deviation, entropy, kurtosis to extract the facial features for a better recognition rate. Also the extracted features are trained and it is associated with a feed forward back propagation neural network used for classification to render better results.

Keywords: Face recognition, Neural network, Features extraction.

I. INTRODUCTION

Face recognition is an active research area in last 30 years. Criminal immigrant detection, Passport authentication, participant identification, system access control, enterprise security, scanning criminal persons, telecommunication are some of the applications of face recognition system [1]. Although there is remarkable progress in the face recognition system, it remains a challenging problem, mainly because of the complexities involved in variations in illumination, where because of different light conditions the face may appear differently. This leads in the misclassification of the input face images. Another problem is posing, where the face recognition system becomes unable to recognize the different poses of the same person. The different poses may include face images with smiling, not smiling, wearing glasses or without glasses and so on. Therefore it becomes necessary to develop an efficient face recognition system.

The approaches for face recognition system can be dealt as analytical and holistic approaches. For analytical approaches, the face outline forms a feature vector which represents a face [2].

Holistic method uses the whole face information. *Principal Component Analysis* (PCA) is a well known holistic method which uses eigenface for face recognition [3]. PCA can achieve the minimum error but it has limitations where it is hard to decide suitable thresholds automatically and to decide upon on how many eigenvectors are required to recover an original face. *Linear Discriminant Analysis* (LDA) is another holistic method which is applied for the fisher face methods. It lags behind where it needs large training sample sets for good generalization. PCA is normally adopted to reduce the feature dimension before LDA can be applied. In the proposed method, we are detecting various features of face like homogeneity, energy, variance, contrast, asymmetry, correlation, mean, standard deviation, entropy, kurtosis each of these features has its own characteristics and overall gives better recognition rate

II. ORGANIZATION OF PAPER

The paper is organized as follows. Section I deals with the brief description of basics of face recognition system, its advantages, why the face recognition system is required. Section III deals with reviews of background of face recognition system followed by the description of early and some new techniques of face recognition system. Section IV deals with proposed method. Section V deals with the simulation and results of the proposed work with classification using feed forward back propagation neural network. Section VI gives the conclusion.

III. RELATED WORK

The face recognition methods are divided into template based method and geometry based methods. Template matching method uses the whole face information.

R. Brunelli and T. Poggio developed the template based method [4], where an array with intensity values is used to represent the image and compared using Euclidean distance.

Sirovich and Penev used PCA method to obtain reduced dimensions of the human faces [6]. PCA is one of the statistical approaches used for the compression of data [5] and is also known as eigenface method. This method was considered robust and produced a good recognition rate on various database. However, PCA lags behind if the parameter range exceeds. It also removes the relationships between neighborhood pixels.

The Linear Discriminant Analysis or the fisher face method is another statistical approach which is considered as better than eigenface method in classifying

face images better than the eigenface method. It uses interclass and intraclass relationships to classify face images. It is robust against noise, occlusion, and illumination [12].

Independent Component Analysis (ICA) is a new statistical technique which extracts independent variables [7]. The technique of ICA is originated from signal processing. However, the training is considered slow and computational cost is high.

Geometry feature based methods were mainly focused on finding the angles, size, and distance between eyes, nose, ears, head outline and mouth. Wavelets is a technique used in the geometry method to decompose complex signals into basis functions, this is similar to Fourier decomposition. For several computer vision applications Gabor filters are used [8].

IV. PROPOSED METHOD

The proposed method uses a well known face database, ORL database from the AT&T laboratories, Cambridge [9]. The architecture of the proposed system is as shown in Fig 1. A typical face recognition system has pre-processing, feature extraction and classification steps. In the proposed method pre-processing is done where the images are converted to matrix form, Feature extraction is done using homogeneity, energy, covariance, contrast, asymmetry, mean, entropy, kurtosis, standard deviation, and covariance. The classification is carried out using feed forward neural network.

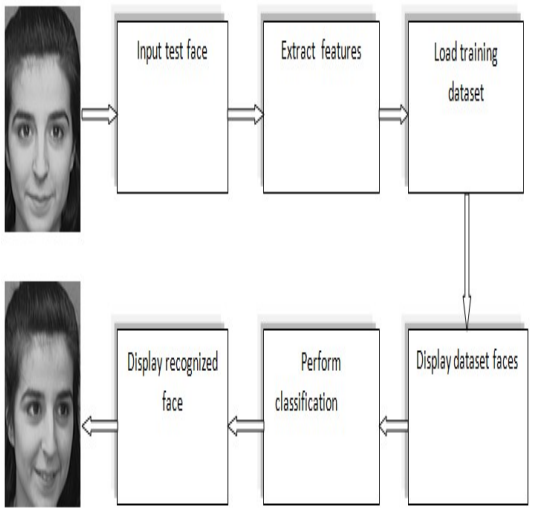

Fig 1: Architecture of proposed face recognition system

A. Pre-processing Stage

The first phase of face recognition system would be collecting the face images. The images may contain unnecessary factors like noise, blurriness, improper illumination effects, distortions etc.

Pre-processing of images is used to enhance the image features which might be important for further processing, it optimize the images to get the satisfactory results on output.

Several pre-processing methods like enhancing brightness of the images, smoothing of images to reduce noise, contrast enhancement, edge enhancement, normalizing intensity of image pixels, removing image reflections, histogram equalization which distributes the brightness levels normally can be applied to the selected input images. In the proposed method images are collected from the ORL database. It is transformed into matrix form in order to ease the operation.

B. Feature Extraction Factors

- **Homogeneity** is used to see the similarity of the pixels, to check the uniformity of the pixels.
- **Energy** is defined as a factor that would capture the desired solution and perform gradient-descent to compute its lowest value, resulting in a solution for the image segmentation.
- **Covariance** is a measure of how much the two pixels change together, the greater values of one pixel mainly correspond with the greater values of the other pixel, and the same holds for the smaller values, i.e., the pixels tend to show similar behavior, the covariance is positive, in the opposite case, when the greater values of one pixel mainly correspond to the smaller values of the other, i.e., the pixels tend to show opposite behavior, the covariance is negative. The covariance shows the linear relationship between the pixels.
- The **Contrast** function enhances the contrast of an image. It defines the intensity of the image; image may have high intensity or the low intensity.
- The **Asymmetry** is without symmetry means, it is the analysis of the abnormal distribution of the pixels in the specified window. It means that there are no mirror images in a composition. It involves the measure of the asymmetry of the data. If asymmetry is negative then the data are spread more to the left of the mean. If it is negative the data are spread more to the right. The asymmetry of the normal distribution is zero.

[1] The **Correlation** defines how much the pixels are closely related to each other. Properties of correlation are:

- Single objects usually have a higher correlation value within them than between adjacent objects.
- Pixels are usually more highly correlated with pixels nearby than with more distant pixels. Smaller window sizes will usually have a higher correlation value than larger windows.
- Correlation is calculated for successively larger window sizes, the size at which the Correlation value declines may be taken as the size of definable objects within an image, which works only if all objects in the image are of same size.

- Correlation uses a different approach in calculation than the other texture measures mentioned above. As a result, it provides different information and can often be used in combination with other texture measures.
- If all the pixels have identical values, then the variances are zero and the correlation equation would give an undefined result.
- If the variance is zero, then the equation is not used, and the correlation is set to 1, to reflect identical pixels.

[2] **Mean** returns the mean of image. It finds out the mean of the row pixels, the mean of the column pixels and the mean together with row and column pixels returning mean of the image.

[3] **Standard deviation** returns deviation between images. It computes the standard deviation of each row and column of the input.

[4] **Entropy** measures the randomness of the pixels, which is used to characterize the texture of the input image [11].

[5] **Kurtosis** describes the peakness of image, frequency distribution of the pixels. If the distribution is normal then the kurtosis has *zero* value. If the kurtosis value is positive its peak distribution will be sharper. Conversely, if kurtosis has negative value lesser the peak distribution will be. The extracted features from test and training images are classified using feed forward neural network.

C. Feature Extraction

In various applications feature extraction is a process of dimensionality reduction, which is required if the input data to be processed is too large and redundant, then the input data is transformed into a reduced dimension set of features, which are also called as features vectors.

In the proposed method the features of the face are extracted and fed as input to the neural network for the further process. Extraction of the relevant information is necessary for the proper recognition to be made.

Rather than working with the entire image it is simple to analyze the relevant extracted features. In the proposed method the features like Homogeneity, Energy, Covariance, Contrast, Asymmetry, Correlation, Mean, Standard Deviation, Entropy, and Kurtosis. The following equations are used to extract the particular features [10].

Features	Expression
Homogeneity	$H=\sum_{i,j}\frac{p(i,j)}{1+\lvert i-j\rvert}$
Energy Range = [0 1] Energy is 1 for a constant image.	$E=\sum_{i,j}p(i,j)^2$ p (i, j) is the pixel value at the point (i, j)
Covariance	$Var=\sqrt{SD}$
Contrast Is 0 for a constant image	$C=\sum_{i,j}\lvert i-j\rvert^2\,p(i,j)$
Asymmetry	$A=\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(\frac{p(i,j-\mu)}{\sigma}\right)^3$ p (i, j) is the pixel value at point *(i,j),* μ and σ are the mean and standard deviation respectively.
Correlation **1** if the image is positively correlated **-1** if the image in negatively correlated	$Corr=\sum_{i,j}\frac{(i-\mu_i)(j-\mu_j)\,p(i,j)}{\sigma_i\sigma_j}$ p_i and p_j, the partial probability density functions.
Mean	$\mu=\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}p(i,j)$ p (i, j) is the pixel value at point (i, j) of an image of size mXn.
Standard Deviation (SD)	$\sigma=\sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(p(i,j)-\mu)^2}$
Entropy	$Ent=-\sum_{k=0}^{L-1}pr_k(\log_2 pr_k)$ pr_k is the k^{th} grey level probability, L total number of grey levels.
Kurtosis	$K=\left\{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left[\frac{p(i,j)-\mu}{\sigma}\right]^4\right\}-3$ p (i, j) is the pixel value at point (i, j), For normal distribution -3 becomes zero.

D. Neural network

The Feed Forward Back propagation neural network is a technique with the network structure, having series of layers, where the inputs flows forward in the network. The network is a collection of nodes connected together. Nodes represent neurons and arrows represent links between them. Input layer has input nodes which simply pass values to the processing nodes. The activity of hidden layer is hidden. The output layer is expected to give the output.

Back propagation neural network works in two phases. Feed forward and feed backward. Network without cycles are called a feed-forward networks, where the inputs flows only forward. In the feed backward network, weights can be used, if the expected output doesn't come then the weights can be changed in such a way that it will achieve expected output.

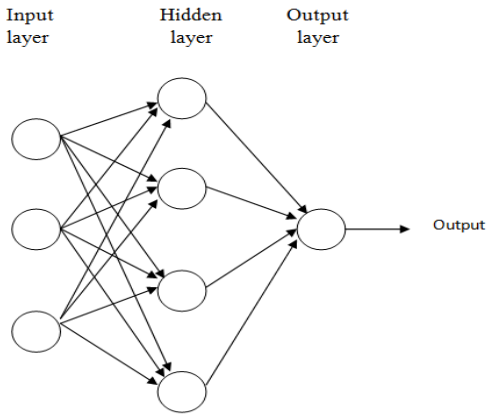

Fig 2: Feed Forward Back propagation neural network

In the proposed method feed forward neural network is used. The features extracted for the face is fed as an input to the neural network. It propagates forward and gives the expected result, which is in this case recognizing the input face. The features obtained for input face is compared with the features of the training dataset faces. The network is trained with the training dataset images. The feed forward back propagation neural network is as shown in Fig 2.

The advantage of using neural network in classification stage is, there is no need to train the network with known inputs repeatedly. Once it is trained it is expected to work efficiently to analyze the data, and is easy to maintain.

V. SIMULATION AND RESULTS

In the proposed method 8 selected face images of different individuals from the ORL database are considered and training data consists of 40 images, each individual with 5 different facial expressions like open/close eyes, smiling or not smiling, glasses or no glasses. The sample dataset is as shown in Fig 4. The test faces are shown in Fig 3. For some faces, the images were taken at different times, varying the lighting. The system has been simulated in MATLAB.

Fig 3: Test data

Fig 4: Sample data set consisting each person image with different facial expression

Any face image among 8 different individuals is given as input to the face recognition system. The training data set is loaded and features are extracted for the training data set. With the extracted features classification is carried out using back propagation neural network. Then a decision is made upon recognizing the face. The test data with recognized face is as shown in Fig 5.

Fig 5: Test face and recognized face

The values obtained for all the features mentioned for the input face is given in Table 1.

TABLE I: THE FEATURE VALUES OBTAINED

Features	Values
Homogeneity (H)	0.2624
Energy (E)	3.8950
Covariance (Var)	3.5248
Contrast (C)	6.9221
Asymmetry (A)	0.2327
Correlation (Corr)	6.9662
Mean (μ)	0.3252
Standard deviation (σ)	131.98
Entropy (Ent)	-693.7
Kurtosis (K)	432051

VI. CONCLUSION

For any face recognition system feature extraction is an important step. If features are not extracted properly recognition becomes difficult. The proposed face recognition system has high recognition rate as it calculates various features like homogeneity, energy, correlation, covariance, mean, standard deviation, kurtosis, asymmetry, contrast, entropy.

Extracted features increase the robustness of the face recognition system. It is found that the recognition rate for the selected database followed by neural network gives high recognition rate and eliminates the variations due to pose.

REFERENCES

[1]. D. Osten, H.M.Carin, M.R.Arneson and B.L.Blan, "Biometric personal authentication system", U.S. Patent #571, 9950, Feb 1998.

[2]. Kin- Man Lam and Hong Yan,"An Analytic – to – Holistic Approach for Face Recognition Based on a Single Frontal View", IEEE Trans., on Pattern Analysis and Machine Intelligence, Vol.20, no. 7, July 1998.

[3]. K.I. Diamantaras and S.Y.Kung,"Principal Component Neural Network:Theory and Applications", John Wiley & Sons,Inc., 1996.

[4]. R. Brunelli, T. Poggio, "Face recognition: Features versus templates", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15(10), pp.1042-1052, 1998.

[5]. M.Turk and A.Pentland," Eigenfaces for recognition", J.Cognitive Neuroscience, vol.3, 71-86., 1991.

[6]. Sirovich, L, Penev, P.S., "The global dimensionality of face space", 4th IEEE International conference, 2000.

[7]. A. Hyvaerinen, E. Oja.," A fast fixed-point algorithm for independent component analysis", MIT Press, Neural Computation, vol. 9, pp.1483-1492, 1997.

[8]. Rahman , M.T, Bhuiyan, M.A, " Face Recognition using Gabor Filters", 11th International Conference, 24-27, Dec 2008.

[9].http://www.uk.research.att.com/facedatabase.html.

[10]. http://www.mathworks.in.

[11]. Fattah,S.A, "Spatial Domain Entropy based local feature extraction scheme for face recognition", 7th International Conference, 24-27, Dec 2012.

[12]. Shuiwang Ji, Jieping Ye, "A Unified framework for generalized linear Discriminant Analysis", IEEE International conference on Computer vision and Pattern Recognition, 2008.

International Journal of Computer Applications Technology and Research
Volume 6–Issue 4, 190-193, 2017, ISSN:-2319–8656

An Examination of the Bloom Filter and its Application in Preventing Weak Password Choices

Nancy Cheng
Department of Computer Science
Stanevagra University

Fabio Rocca
Department of Computer Science
Stanevagra University

Abstract: Choosing weak passwords is a common issue when a system uses username and password as credentials for authentication. In fact, weak passwords may lead to system compromising. Lots of approaches have been proposed to prevent user from selecting weak or guessable passwords. The common approach is to compare a selected password against a list of unacceptable passwords. In this paper we will explain space-efficient method, which is called Bloom Filter, of storing a dictionary passwords. The time complexity in this approach is constant time, no matter how many passwords we have in our dictionary.

Keywords: Weak passwords; Bloom filter; Authentication; Security; Web Security.

1. INTRODUCTION

Authentication is an absolutely essential element of having a secure environment in digital world. In this process the identity of the user (or in some cases, a machine) will be checked against the some credentials that he/she has provided before for a system[3,9, 12]. If they are matched the user will be authenticated and forwarded for the next step which is usually authorization. There are several approaches for authentication which are selected based on the system administrator policies and available infrastructures. One of the basic authentication mechanism is using user name and password to provide some fundamental security characteristics. This technique has been a part of security since long time ago[2,6,11]. However, now a days, systems administrators need to re-examine their password security policies to remain effective against modern programs and computers that can crack weak passwords in minutes. But it has always been a concern that how we can select a proper and strong password that cannot be predicted or cracked easily. Lots of approaches and techniques have been proposed in this area that try to generate a complicated combination of characters and numbers as password[14,15]. In the most cases it works but it is not always comfortable for the user to have or memorize a system generated password, instead if we could have a dictionary of weak passwords which are easy to recognize by attackers and check them against the entered password proposed by user we can easily decide the selected password by end user is strong enough or not. Having a dictionary of weak passwords to be checked against the entered password usually involves with reading the whole dictionary which needs extra facilities and hardware but there is a data structure model that can address this issue easily which is called Bloom Filter. Bloom filter is an efficient data structure in terms of space usage and lookup time that will be explained in details in the next section.

2. BLOOM FILTER

Bloom filter was introduced in “Space/time trade-offs in hash coding with allowable errors” paper by Burton H. Bloom[1]. In this paper he compared the space/time trade-offs of different types of hash-tables[12,14]. He found that a new type of hash-table, which is now known as a Bloom filter needed less time to reject elements that are not in the table and less space to store these elements[11]. Bloom filter as an efficient probabilistic data structure has been used in different areas such as checking for viruses in network packets, spam control in email, web caching, spell, password checking and etc [3,18]. It consists of a bit array of m bits which are all set to zero, adding the elements to the m bits of array can be done through the k different hash functions. Based on the output of each hash function the particular position in array will be changed to 1 and later on, the structure can be queried for the membership of elements. So the elements themselves are not stored in the Bloom filter, only their membership [5]. Figure 1 shows the

structure of bloom filter.

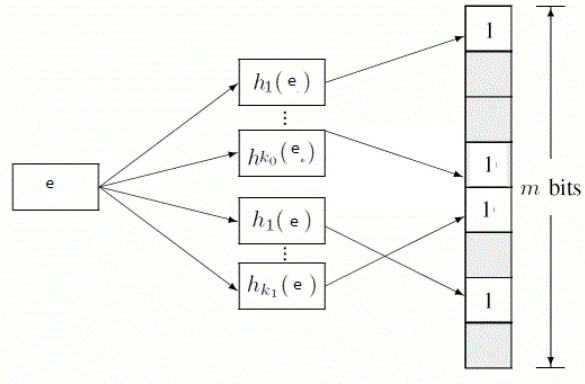

Figure 1: Bloom filter structure [5].

For the coming request after applying k hash functions, the bloom filter will be checked If one or more of mapped bits are still 0, the element is certainly not in the set. If all bits are 1, the element was probably in the set, although there is a small probability that the tested bits were set to 1 due to the addition of different elements. Then we have a false positive. There is a trade-off between the probability of false positives and the size of the Bloom filter (m), number of hash functions (k) and number of items in the set (n)[5]. The false-positive probability can be calculated from m and k in the following way. The probability p of one of the m bits still being zero after the addition of n elements is[5]

$$p = \left(1 - \frac{1}{m}\right)^{kn} \approx e^{-kn/m}.$$

And the probability of a false positive f is then equal to the probability that all the k bits that we test are equal to 1, which is equal to

$$f = (1-p)^k = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \approx \left(1 - e^{-kn/m}\right)^k.$$

From simple calculations it follows that for a given m and n, the value of k that minimizes the false- positive probability f is equal to:

$$k = \frac{m}{n} \ln 2$$

Which gives the probability of

$$f = \left(\frac{1}{2^{\ln 2}}\right)^{m/n} \approx 0.62^{m/n}$$

There are some main parameters that affects the accuracy in bloom filters, table 1 shows the key components in bloom filters [5].

Table 1: Key Bloom Filter Parameters

Parameter	Outcome
Number of hash functions (k)	More computation, lower false positive rate
Size of filter (m)	More space is needed, lower false positive rate
Number of elements in the set (n)	Higher false positive rate

3. EXPERIMENT AND EVALUATION

In the current scenario the dataset of popular weak password is needed, so the dataset of weak passwords with 300,000 passwords was selected. For this project two approaches experimental and theoretical will be calculated and compare together to have a good inside about the implemented version of bloom filter besides having some optimal values for the other parameters. The key practice in this project is to calculate the lowest false positive rate for the implementation through the adjusting the key parameters. To address that, the list of 30,000 strong password was used to calculate the false positive rate from empirical point of view.

The first experiment is to calculate the false positive rate based on different hash functions to have a basic idea about the some optimal numbers of hash functions. In bloom filter using more hash functions may be considered as a way to reach the lowest positive rate, it is true but at some points adding more hash functions to the system does not work and from those points the false positive rate again starts to grow. In practice, hash functions yielding sufficiently uniformly distributed outputs, such as MD5 ,CRC32 ,SHA1, SHA2, murmur, FNV which are useful for most probabilistic filter purposes like bloom filter [5].

For the first experiment the m/n = 6 was selected and different hash functions were applied. Figure 2 shows the details about two approaches

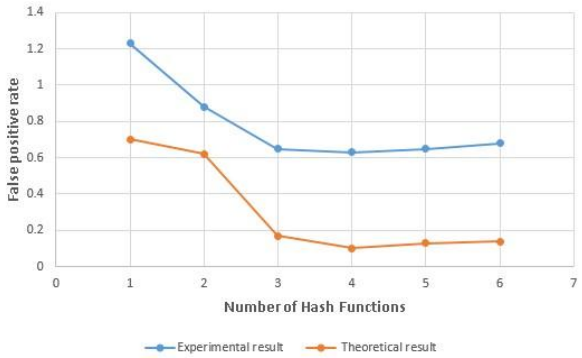

Figure 2: False positive rate through different number of hash functions

As figure 2 depicts the experimental result has a little higher error in comparison with the theoretical one but almost both are following the same trend. From the theoretical result point of view, K=4 plays an optimal solution for the given data and the experimental one confirms this result. Table 2 shows the exact result for both cases.

Table 2: False positive rate through different number of hash functions

K	Experimental result	Theoretical result
1	1.23	0.7
2	0.88	0.62
3	0.65	0.17
4	0.63	0.1
5	0.65	0.13
6	0.68	0.14

For the next step the false positive rate is calculated based on the different bloom filter sizes (m). Figures 3, 4 and 5 shows the different values for false positive rates while we are applying different number of hash functions

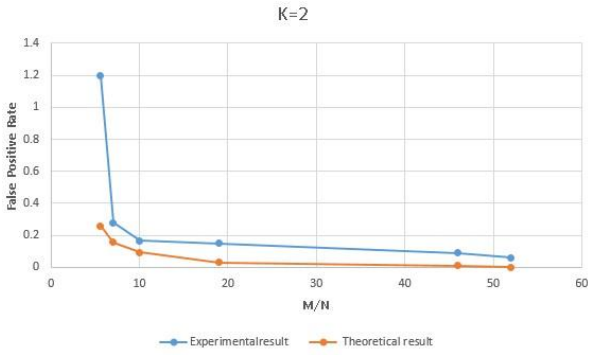

Figure 3: False positive rate with different size of bloom filter (K=2)

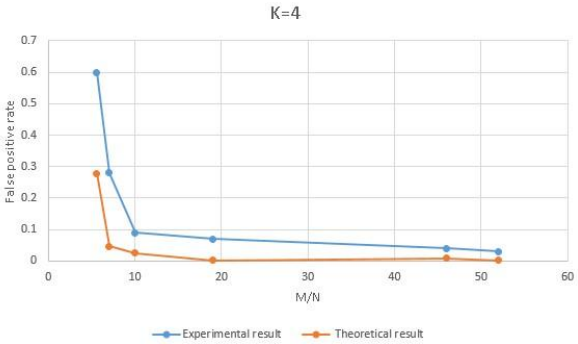

Figure 4: False positive rate with different size of bloom filter (K=4).

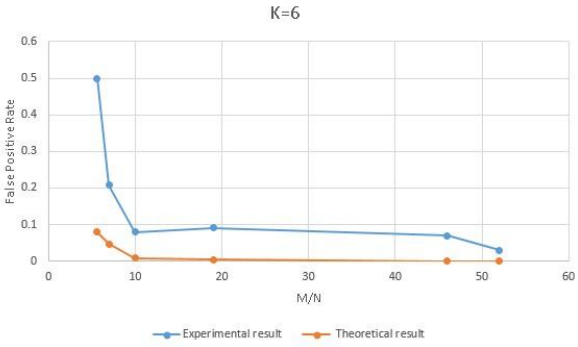

Figure 5: False positive rate with different size of bloom filter (K=6).

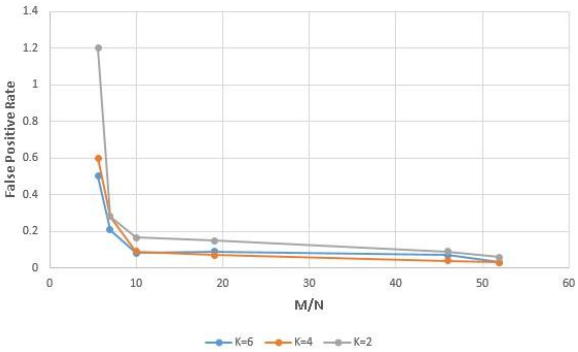

Figure 6: False positive rate with different size

of bloom filter and different number of hash functions. As the figure 6 shows the lowest false positive rate can be achieved with a bloom filter with m/n=50. Although, it can be considered as an optimal solution for this scenario but if we had some restrictions in regards of required space, we can consider the m/n=10 with number of hash functions equal to 4. In fact, it depends on the domain and how accurate we expect to receive the outcome from the bloom filter. In the current case as we do not have that much sensitivities we can take K=4 with m/n=10 as an optimal solution.

4. CONCLUSION

Bloom filter is a compact data structures for probabilistic representation of a set in order to support membership queries[7,8]. It has a strong space advantage over other data structures for representing sets and it also comes with lookup time efficiency in comparison with other approaches, but as may be expected, there is always a tradeoff and for this case the false positive is a tradeoff for space and time efficiency. Based on the domain and acceptable false positive rate, the bloom filter can be adjusted to achieve the optimal lookup time beside space efficiency. In this paper, based on the input data and admissible false positive error rate, 4 independent hash functions with m/n =10 were selected as an optimal solution.

REFERENCES

[1] Spafford, Eugene H. "Preventing weak password choices." (1991).

[2] Broder, Andrei, and Michael Mitzenmacher. "Network applications of bloom filters: A survey." *Internet mathematics* 1, no. 4 (2004): 485-509.

[3] Ganesan, Ravi, and Christopher I. Davies. "Method and system for proactive password validation." U.S. Patent 5,394,471, issued February 28, 1995.

[4] Mitzenmacher, Michael. "Distributed, compressed Bloom filter Web cache server." U.S. Patent 6,920,477, issued July 19, 2005.

[5] Porat, Ely. "An optimal Bloom filter replacement based on matrix solving." In *International Computer Science Symposium in Russia*, pp. 263-273. Springer Berlin Heidelberg, 2009.

[6] Pouriyeh, Seyed Amin, and Mahmood Doroodchi. "Secure SMS Banking Based On Web Services." In SWWS, pp. 79-83. 2009.

[7] Weirich, Dirk, and Martina Angela Sasse. "Pretty good persuasion: a first step towards effective password security in the real world." In *Proceedings of the 2001 workshop on New security paradigms*, pp. 137-143. ACM, 2001.

[8] Hao, Fang, Murali Kodialam, and T. V. Lakshman. "Building high accuracy bloom filters using partitioned hashing." In *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 1, pp. 277-288. ACM, 2007.

[9] Shanmugasundaram, Kulesh, Hervé Brönnimann, and Nasir Memon. "Payload attribution via hierarchical bloom filters." In *Proceedings of the 11th ACM conference on Computer and communications security*, pp. 31-41. ACM, 2004.

[10] Kirsch, Adam, and Michael Mitzenmacher. "Distance-sensitive bloom filters." In *2006 Proceedings of the Eighth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pp. 41-50. Society for Industrial and Applied Mathematics, 2006.

[11] Pouriyeh, Seyed Amin, Mahmood Doroodchi, and M. R. Rezaeinejad. "Secure Mobile Approaches Using Web Services." In SWWS, pp. 75-78. 2010.

[12] Zhong, Ming, Pin Lu, Kai Shen, and Joel Seiferas. "Optimizing data popularity conscious bloom filters." In *Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing*, pp. 355-364. ACM, 2008.

[13] Mitzenmacher, Michael. "Bloom filters." In *Encyclopedia of Database Systems*, pp. 252-255. Springer US, 2009.

[14] Allahyari, Mehdi, Krys J. Kochut, and Maciej Janik. "Ontology-based text classification into dynamically defined topics." In Semantic Computing (ICSC), 2014 IEEE International Conference on, pp. 273-278. IEEE, 2014.

[15] Tarkoma, Sasu, Christian Esteve Rothenberg, and Eemil Lagerspetz. "Theory and practice of bloom filters for distributed systems." *IEEE Communications Surveys and Tutorials* 14, no. 1 (2012): 131-155.

[16] Melicher, William, Blase Ur, Sean M. Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. "Fast, lean and accurate: Modeling password guessability using neural networks." In *Proceedings of USENIX Security*. 2016.

[17] Laufer, Rafael P., Pedro B. Velloso, and O. C. M. B. Duarte. "Generalized bloom filters." *Electrical Engineering Program, COPPE/UFRJ, Tech. Rep. GTA-05-43* (2005).

[18] Allahyari, Mehdi, and Krys Kochut. "Automatic topic labeling using ontology-based topic models." In *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on*, pp. 259-264. IEEE, 2015.

[19] Morris, Robert, and Ken Thompson. "Password security: A case history." *Communications of the ACM* 22, no. 11 (1979): 594-597.

[20] Kaufman, Charlie, Radia Perlman, and Mike Speciner. *Network security: private communication in a public world*. Prentice Hall Press, 2002.

[21] Rosenberg, Jothy, and David Remy. *Securing Web Services with WS-Security: Demystifying WS-Security, WS-Policy, SAML, XML Signature, and XML Encryption*. Pearson Higher Education, 2004.

[22] Rubin, Aviel D., Daniel Geer, and Marcus J. Ranum. *Web security sourcebook*. John Wiley & Sons, Inc., 1997.

International Journal of Computer Applications Technology and Research
Volume 6–Issue 4, 194-198, 2017, ISSN:-2319–8656

Cost-Efficient Task Scheduling with Ant Colony Algorithm for Executing Large Programs In Cloud Computing

Fatemeh Imani
Department of Computer Engineering, Ardabil Branch, Islamic Azad University, Ardabil, Iran

Shiva Razzaghzadeh
Department of Computer Engineering, Ardabil Branch, Islamic Azad University, Ardabil, Iran

Masoud Bekravi
Department of Computer Engineering, Ardabil Branch, Islamic Azad University, Ardabil, Iran

Abstract: The aim of cloud computing is to share a large number of resources and pieces of equipment to compute and store knowledge and information for great scientific sources. Therefore, the scheduling algorithm is regarded as one of the most important challenges and problems in the cloud. To solve the task scheduling problem in this study, the ant colony optimization (ACO) algorithm was adapted from social theories with a fair and accurate resource allocation approach based on machine performance and capacity. This study was intended to decrease the runtime and executive costs. It was also meant to optimize the use of machines and reduce their idle time. Finally, the proposed method was compared with Berger and greedy algorithms. The simulation results indicate that the proposed algorithm reduced the makespan and executive cost when tasks were added. It also increased fairness and load balancing. Moreover, it made the optimal use of machines possible and increased user satisfaction. According to evaluations, the proposed algorithm improved the makespan by 80%.

Keywords: cloud computing; formatting; task scheduling; makespan; load balancing

1. INTRODUCTION

With the development of information technology, it is necessary to do computing tasks everywhere at every time. It is also essential that people be able to perform their heavy computing tasks through some services without having expensive hardware and software requirements. Cloud computing is the latest solution provided by technology for these needs. The National Institute of Standards and Technology (NIST) defines cloud computing as a model to provide easy access based on user demand through the network for a group of modifiable and configurable computing resources such as networks, servers, storage space, applications and services. This access should be able to offer quickly-provided or free services without needing resource management or direct intervention. In this definition, the cloud can be described with five essential features including virtual computing resource sharing, WAN access, quick flexibility, requested services (based on order or demand), and measured services. The cloud environment is based on requests, and users can increase or decrease the use of resources. In other words, the results are related to usage in the cloud environment. Sharing the usable computing power among some tenants can improve the productivity rate because servers are not idle for nothing in this method. One reason is that computers are used more because cloud computing customers do not need to calculate and determine the maximum load [1]. Virtual machine scheduling problem has been investigated in many studies conducted on cloud computing environments. The main aim of scheduling in the cloud is to shorten the makespan, increase the system throughput, and establish load balancing in resource [2]. Running a program can be seen as the execution of different tasks in it. Different tasks can be executed simultaneously with several virtual machines. In this thesis, ACO was used with a fair and accurate resource allocation approach based on machine performance. As the number of natural resources should be proportionate to the performances of ants, the tasks performed by resources should be proportionate to the tasks assigned to them. An appropriate path for resource allocation is very important to perform tasks. Therefore, machines can process tasks at the maximum power and on the shortest path. As a result, resources are used optimally. In the cloud environment, providers want to make the most of their resources, and users want to minimize their costs [4]. However, they want to achieve their intended performance. The appropriate and optimal use of resources such as memory, processor and bandwidth is a challenge; therefore, the quality of task scheduling is regarded an important problem which has a great effect on the performance of cloud service providers. All of the scheduling algorithms are intended to minimize the makespan [4]. None of the previous studies dealt with task distribution in a cloud environment by considering the costs and optimal use of machines and increasing their idle times. Considering task scheduling algorithms, the ACO was adapted from social theories in this thesis to optimize the use of machines, reduce machine idle time, and decrease the makespan. It was also used to minimize executive costs by modifying the idle time (vms) to perform insensitive tasks.

2. LITERATURE

In this section, previous research works on network processing are reviewed. First, preliminary methods such as Dynamic Level Scheduling are described, then the most recent methods are reviewed.

2.1 Dynamic Level Scheduling

For this purpose, a specific model has been proposed. The main aim of this method is to decrease the processing time. In network processing environments, other scheduling algorithms do not emphasize the subtasks of an application which is run in the computing host or the virtual organization. The main aim is to perform scheduling in a way that all the input applications can use the available throughput. In the paper presented by Cathody and Caratasa, the heuristic technique was added to the abovementioned method to increase the system efficiency [5].

2.2 Allocating the Fast Process to the Largest Task

The FPLTF scheduling algorithm (Xoa et al.) determine the tasks based on the available resources in the system [6]. This method depends on speed of processor and resources and the size of tasks. In this method, the largest task is allocated to the fastest resource. If there are many large tasks, this method will not be efficient enough. The dynamic FPLTF (Chang et al.) algorithm was developed with respect to the static FPLTF algorithm. In this method, the highest priority is allocated to the largest task. It is also necessary to estimate the data which are required for processing [7].

2.3 WQR (Queue with Repeat)

This method is based on WQ. In this method, faster processors are allocated to large tasks (Young et al.) by using the FCFS and random scheduling methods. WQR iterates the tasks to transfer them to available resources. The iteration of tasks can be selected by the user. When one of these tasks is finished, the scheduling algorithm stops the iteration of other tasks. One of the problems of this method is that it spends too much time allocating resources to the iteration operations.

2.4 Balanced Ant Colony Optimization (BACO)

The main idea of this method is taken from ACO (Xoa et al.), and it is mainly intended to decrease the processing time and load balancing of each resource. This method changes the density of pheromone based on the positions of resources, something which can be possible by updating the pheromone locally and globally. In this method, the makespans are shortened at the same time as the system is kept in balance. In the architecture of this network process scheduling, there are four components: portal, information server, task scheduling algorithm, and resources required for processing. The portal is used as an interface for users [6].

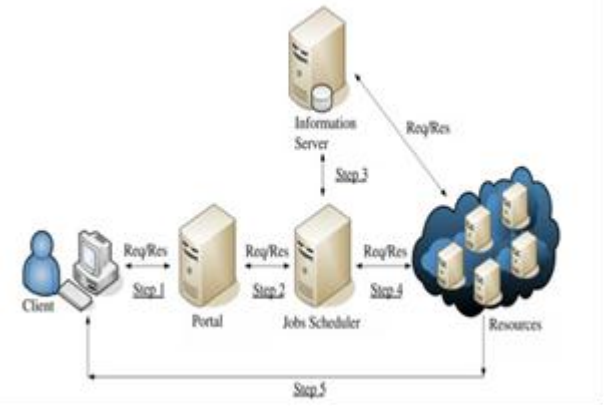

Figure 1. Structure of the system

2.5 Load Balancing

Load evaluation is usually mechanized for the continuity of a service when one or some components of the system fail. The components are constantly monitored. When one component does not respond, the load balancer comes up and prevents the traffic from being sent to it. With an appropriate load evaluation where resources are used, problems can usually be mitigated. Not only does this decrease costs and creates green computing, but it also keeps the pressure low on unique circuits whose lifetime will be potentially elongated. In fact, it can be stated that the load balancing is meant to find an appropriate map of tasks on the available processors in the system in a way that each processor runs an equal number of tasks until the total makespan is minimized as much as possible.

2.6 The Importance of Load Balancing

With load balancing, the load can be balanced through the dynamic transfer of local tasks from one machine to another one in a remote node or a machine which is used less often. This solution maximizes user satisfaction, minimizes response time, increases the exploitation of resources, increases the failure times and improves the system efficiency. Load balancing is also needed to achieve green computing in the clouds [8].

3. OPTIMIZED ACO

Cloud computing is the extended version of network computing which is done in a parallel and distributed way. It is also a new model for business computing. Compared with network computing, the new features of cloud computing include heterogeneous resources distributed and dispersed in large scale to include the datacenter. Moreover, the virtualization technology creates latent heterogeneous resources in cloud computing. Network computing is generally used in scientific computing to solve the limited domain problem. Cloud computing provides a user-oriented plan which offers various services to meet the needs of users. In cloud computing, resources are converted into virtual resources by using the virtualization packaging technology. This makes the resource allocation and interaction process be different from user tasks and network computing [9,10].

3.1 Designing the Optimized ACO

To design the optimized ACO, user tasks are allocated to resources which are the same as the machine output so that the machine can do the processing with full power, and there should not be any loads on resources. This increases the efficiency of resources. Now tasks should be classified and prioritized for the fair allocation of resources. This action was not possible in the normal ACO. Users, resource providers and the scheduler system are intended in cloud computing. The main part of scheduling computations includes user tasks and resources so that the fair distribution of resource allocation can be possible in cloud computing by using the optimized algorithm. The scheduling algorithm is optimized includes two main steps. Figure 2 shows the architecture of the optimized algorithm.

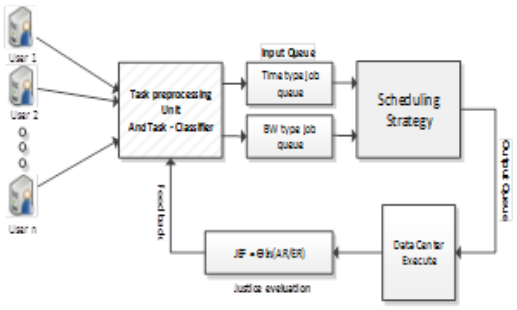

Figure 2. Architecture of the optimized algorithm

3.2 The Strategy of the Proposed Algorithm (Optimized Scheduling)

Figure 2 shows main mechanism of the algorithm in this thesis. It uses the local optimal method to allocate resources to a group of tasks and virtual machines. Now the algorithm is briefly described.

1. The tasks sent by the user are classified and prioritized with respect to the quality parameters in the computing unit.
2. Given the type of tasks, two lists are created. One of them is meant for processing tasks (runtime), and the other one is meant for the tasks needing bandwidth.
3. A group of virtual machines named VMlist is given to the system based on their normalized performances.
4. Given the number of virtual machine processors and the expected waiting time of each task, the virtual machine is selected.
5. Given the real bandwidth of virtual machine and the expected bandwidth of tasks, the desired machine is selected.
6. If tasks are equal to the performances of resources, they are assigned to them. In other words, fairness is considered in the allocation of tasks.
7. Finally, the virtual machine is freed after task processing is done.

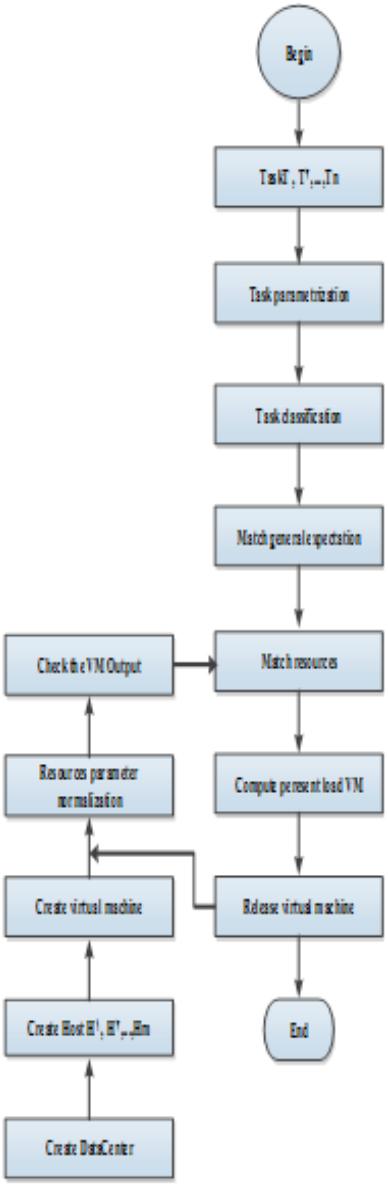

Figure 3. ACO algorithm

4. SIMULATION ENVIRONMENT

The experiments were implemented in the Cloud simulator using library functions including CloudSim, SimJava, and GridSim. This test is run in the CloudSim environment, and the application is run at the user layer code [10,11].

4.1 Evaluation Criteria

In the next sections, three tests are conducted in the form of different parameters such as makespan, the number of processors, bandwidth, and user satisfaction. Then the optimized algorithm is compared with different algorithms. In this evaluation, the optimized ACO algorithm is compared with other scheduling algorithms based on Berger's model and greedy model. In all the experiments, virtual machine parameters including machine ID, the number of processors, available memory, and bandwidth in Table (2-4) were used along with task parameters such as task ID, class ID, length, file size, output size, the expected time, and the expected bandwidth according to Table (1-4). Then the proposed algorithm is compared with other algorithms [12, 13].

Table 1. Jobs Parameter

Task Id	Class type	Length	File_size	Output_size	Expectation time	Expectation BW
0	**1**	**4000**	**2500**	**500**	**400**	**-**
1	**1**	**3000**	**2000**	**400**	**200**	**-**
2	**1**	**2000**	**800**	**300**	**150**	**-**
3	**1**	**5000**	**5000**	**2000**	**500**	**-**
4	**2**	**2000**	**800**	**300**	**-**	**2000**
5	**2**	**3000**	**2000**	**400**	**-**	**3000**
6	**2**	**800**	**300**	**300**	**-**	**1200**
7	**2**	**2500**	**1000**	**500**	**-**	**2000**

Table 2. Vm Parameter

VM ID	CPU	Memory	Bandwidth
0	4	2048	1200
1	2	1024	3000
2	2	1024	1000
3	1	512	1200

4.2 Comparing the Results of Simulating the Proposed ACO Method with Berger and Greedy

4.2.1 Makespan in Different Algorithm

In this experiment, the optimized algorithm was compared with the scheduling Berger algorithm and greedy base by considering the makespan. The tests were conducted in a highly heterogeneous environment. Figure 4 and Table 3 show the runtime of tasks in an analytic comparison.

Table 3. Vm Parameter

Task Id	0	1	2	3	4	5	6	7
ACo-O	400	500	200	1000	200	600	80	250
Berger	400	500	200	520	250	520	140	500
Greedy Base	400	500	200	600	200	300	90	500

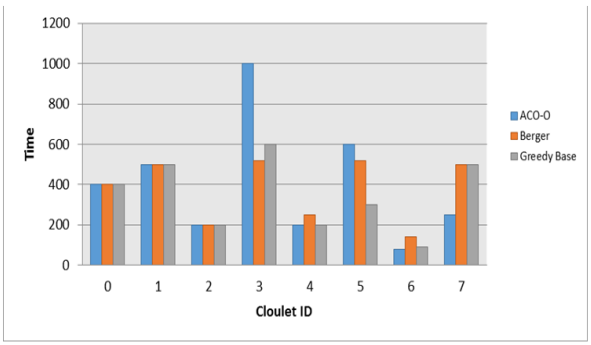

Figure 3. Comparing completing time

4.2.2 Comparing the Number of Processors in the Machines Allocated to Tasks

In this section, the proposed algorithm is compared to different scheduling algorithms by using improvement strategies with respect to quality. The tests are run in this range. The main research goal and necessity is to select resources and to achieve the best time as well as the appropriate cost. According to the datasets shown in Tables 2 and 3, the number of processors of machines used in the Berger and Greedy Base algorithms can be compared with the proposed algorithm.

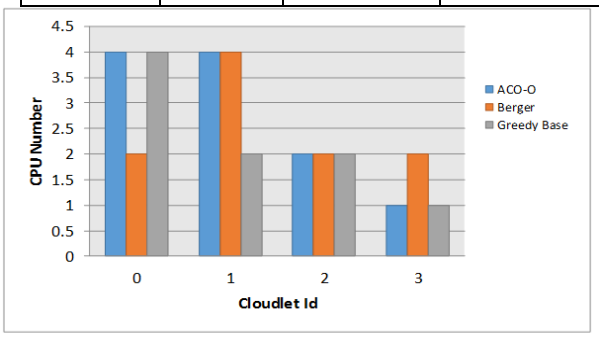

Figure 4. Comparing the Number of Processors in the Machines Allocated to Tasks

5. SUGGESTIONS

More studies can be conducted on resource scheduling. Regarding future works, some suggestions are made as follows:

• A fuzzy neural network of the service quality of tasks and resources can be used with an ACO approach in the future.

• The methods based on the genetic algorithm can be used to make significant improvements in the proposed algorithm because they are very helpful in optimization problems.

6. REFERENCES

[1] Danielson K. 2008. Distinguishing Cloud Computing from Utility Computing . Available from http:/ / www. ebizq. net/ blogs/ saasweek/ 2008/ 03/

distinguishing_cloud_computing/ . [Accessed 26 March 2008]

[2] Sran N. and Kaur N. 2013. Comparative Analysis of Existing Load Balancing Techniques in Cloud Computing, International Journal of Engineering Science Invention, Vol.2, Issue 1.

[3] Hamo A. and Saeed A. 2013. Towards a Reference Model for Surveying a Load Balancing, International Journal of Computer Science and Network Security , Vol. 13.

[4] Peng L. 2009. the definition of cloud computing and characteristics. http:// www.china cloud .cn/ [Accessed 25 February 2009]

[5] Gkoutioudi, H. D. Karatza, "Task cluster scheduling in a grid system", Simulation Modeling Practice and Theory, Vol. 18, pp. 1242-1252, 2010.

[6] Chang and et al, "An ant algorithm for balanced job scheduling in grids", Future Generation Computer Systems, Vol. 25, pp. 20-27, 2009.

[7] Goa Y. and et al, "Adaptive grid job scheduling with genetic algorithms", Future Generation Computer Systems, Vol. 21, pp. 151-161, 2005.

[8] Begum S and Prashanth C S R. 2013. Review of Load Balancing in Cloud Computing. International Journal of Computer Science Issues , Vol.10, Issue 1.

[9] Khetan A, Bhushan V and Chand Gupta S. 2013.A Novel Survey on Load Balancing in Cloud Computing International Journal of Engineering Research & Technology

[10] Sundararajan S, Raj B. 2011. IBM Center for High Performance On Demand Solutions. IBM Cloud Computing White Paper http://www.ibm.com/developerworks/Websphere /zones/hipods/.[Accessed2011].

[11] Lia J, Fenga L, Fang S. 2014. An Greedy-Based Job Scheduling Algorithm in Cloud Computing. JOURNAL OF SOFTWARE, VOL. 9, NO. 4, APRIL.

[12] Buyya R, Ranjan R and Rodrigo N. 2009. Calheiros, Modeling and simulation of scalable cloud c computing environments and the CloudSim Toolkit: challenges and opportunities. In: Proceedings of the seventh high performance computing and simulation conference (HPCS 2009, ISBN:978-1- 4244-49071), Leipzig, Germany. New York, USA: IEEE Press; June 21-24.

[13] Baomin X U C Z . 2011. Job scheduling algorithm based on berger model in cloud environment. Advances in Engineering Software, vol. 42, pp. 419–425.

International Journal of Computer Applications Technology and Research
Volume 6–Issue 4, 199-208, 2017, ISSN:-2319–8656

Significance and Application of Computer-Based Forecasting to Governance and Leadership

Fergus U. Onu
Department of Computer Science,
Ebonyi State University, Abakaliki - Nigeria

Michael O. Ezeji
Department of Computer Science,
Ebonyi State University, Abakaliki - Nigeria

Abstract: Forecast results, products and applications are crucial in different sectors of a nations's economy. Areas such as health, social, political, commerce and infact overall economy require the use of forecast techniques; but, success in governance and leadership seem to depend majorly on proper use of efficient forecasting techniques. This study focuses on the significance and application of computer-based forecasting in governance and leadership generally. The research being a descriptive type uses data from primary sources (observation, oral interview) and secondary sources (online publication) to locate, collect, analyze and summarize opinions on the significance and applications of computer-based forecasts to governance and leadership concerns. The work revealed that poor forecasting in governance leads to uninformed decision making manifested in poor service delivery, high project cost, insuficient amenities and overall below average performance of leadership. The effect of this on stakeholders is enormous. Tax-payers face untold hardship and lack of basic amenities. Governments waste resources on wrongly chosen projects and communities loss confidence and interest in the government. Consequently, the governance becomes unpopular and leadership fails. The application of computer-based forecasting provides a solution to these economic ills.

Keywords: Forecasting, Computer-Based Forecasting, Application software, Forecasting in Governance, Forecasting in Leadership, Governance, Leadership.

1.0 INTRODUCTION

Forecasting cum planning form the basis for rationale decision making. Development of accurate and reliable forecast requires hence necessitates the knowledge of the *methods* and *ability* to analyze and evaluate the changing conditions in micro- and macro- environment in which governance cum leadership operates. Governance is a point where management of resources, administration of policies and leadership of people intersect (Ihezuo, 2016:p.214). While governance is sub-leadership, management is a sub-economy [Ihezuo, 2016]. Forecasting is great tool in management of resources and it keeps coming up in economic indices. Evolution and skillful use of correct forecast is the basis for making optimum unexaggerated decisions in managing resources. Put inversely, the *goal* of forecasting is to provide possible most objective and substantial prerequisites "informed/educated guess" for decision-making and analysis of many facts that may involve governance as the case may be in this topic: *Significance and Application of computer Based Forecasting to Governance cum Leadership*. The need for forecast is *inherent* in any system for organization. For instance, governance is set to organize and govern the people to achieve a desired goal (Ihezuo, 2016:p.214). Many things are involved! But just for instance, every year, every governance system be it corporate or public make annual forecast of revenue and expenditure for the expected incomes and expenditures [recurrent and capital] for their organization technically called annual *budget*. It, as an estimate as the name implies is best a forecast, an estimate and prediction … a guess of what should be gotten and what should be spent. They are arrived at quantitatively [measurable, hence programmable – computer-based] and qualitatively [non measurable, intuitive, gut and judgmental]. "Forecast is based on predicting thus *inferring about the unknown event based on the already known events*" (Cieślak, 2011:p.18). The *aim* of forecast in any organization or society is to provide data cum information about current estimates and future changes and impacts these changes can bear on actuality.

Motivation:

The concern for finding reason hence solution for failed/abandoned projects, deficit in budget appropriation, wrong population census estimates and its adjustments, amenities and developments distributions were strong motivation for this work. These decisions failed because the information required to generate the right decision are wrongly reached. Note, information is a processed data. The act of processing these data can take different nature. In forecasting world, most decisions are arrived at subjective and by mere guesswork: qualitative techniques. This is seen by high mortality rate of those decisions. An objective technique needs examined to study their significance and application in forecasting.

Aim & Objectives

The aim of the study is to exegetically explore the significance and application of Computer-Based Forecasting in governance.

The specific objectives are to know the place of Computer-Based in terms of values, usefulness and applications as contrasted from other forecasting techniques.

The study focuses more on the public and corporate administration which are the objects of governance. But

underlying in any group is "governance" [Ihezuo, 2016:pp.280]. It can be good or bad. No government can lay claim to good governance if decisions affecting the people are not reached objectively but judgmentally [subjective] making the people to wallow in poverty and underdevelopment. People want to **belief** [have confidence in] the process for reaching their life decision to be unbiased. This is so because according to Ihezuo (2016:281), governance is defined as ***the process of decision-making and the process by which decisions are implemented or not implemented***. It is an intersection crossing the two parallel lines of management and leadership.

The essential component of this forecast system is *collection*, *selection* and *analysis* of *internal* and *external* data mostly *historical*, which should meet a number of formal criteria such as *availability*, *completeness* and *comparability*.

2.0 REVIEW OF LITERATURE

2.1 Forecasting Defined

According to onlineDictionary.com forecasting literally means to predict or estimate a future event or trend. Forecasting is a judgment of the likelihood of a particular event at the time defined with the accuracy of a moment (point) or a period (range) of time in the future (Cieślak, 2001). The online dictionary (Wikipedia.org) defines forecasting is the process of making predictions of the future based on the past and present data and most commonly by analysis of trends. It says prediction is similar but more general term. The business dictionary (www.BusinessDictionary.com) sees forecasting as a planning tool that helps management in its attempts to cope with the uncertainty of the future, relying mainly on data from the past and present and analysis of trends. More still, investopedia (www.Investopedia.com) stated the forecasting refers to uses of historic data to determine the direction of future trends.

Forecasting according to BusinessDictionary.com starts with certain assumptions based on the management's experience, knowledge and judgment. The appropriate forecasting method depends largely on what data are available. This judgment characteristically should;

- ≈ Be formulated using achievement of *modern science*
- ≈ Relate to a *predefined future* not endless future
- ≈ Be *empirically verifiable* even when judgmental.

2.2 Forecasting Techniques versus Forecasting Systems

A forecasting technique or method is a mathematical equations that forecasts some future value or event. While many statistical forecasting user-programs or software packages are implementations of forecasting methods. A forecasting system is said to be a computer-based system if that system collects and processes demand data for thousands of items (iteratively), develops forecasts using forecasting methods, has an interactive administrative and management user interface, maintains a database of demands, and has report file-writing capabilities [Kurzak, 2012]. This is the crux of this research: Significance and Application of Computer-Based forecasting…

A forecasting system is much more complex than a forecasting technique. For this reason, this study dwells on forecasting systems. *Computer-based* forecasting system is just one example of theses forecasting systems. Of course, the forecasting method is a part of the forecasting system (CFO Research Service, 2009). Computer-Based forecasting system incorporates forecasting techniques such as regression analysis, curve fitting, evaluation of closeness to a fit, moving averages (simple, exponential and weighted) and seasonal adjustments.

2.2.1 Theory Of Forecasting

Essential part of managing enterprises is to forecast future events or occurrences which is, technically called forecasting (Lucian, 2012:pp.176). Please note that governance is a form of enterprise, a form of project and a form of organization (Ihezuo, 2016). These three; enterprise, governance and organization involve coming together of two or more people organized in a specific form for a goal. It is Robbins & Judge (2014:pp.39) that define organization *as a consciously coordinated social unit, composed of two or more people, that functions on a relatively continuous basis to achieve a common goal or a set of goals*. So, no decision can worth a pinch of salt without accurate and precise forecasts as they primarily concern the enterprise's, organization's and government's future. Hence, the causes for development of forecasts are:

- ⇘ uncertainty even in environment
- ⇘ delay in time between the moment when the decision is made and its effects.
- ⇘ conditions of certainty i.e. knowledge of the enterprise's environment,
- ⇘ conditions of risk, when the likelihood of possible variants in the environment state is known,
- ⇘ conditions of uncertainty i.e. lack of knowledge of the likelihood of the possible variants of the environment state
- ⇘ conditions of incomplete information, which are connected with the lack of knowledge of all the possible variants of the environment state.

The environment [often called macro-environment generally] in this case is regarded to be *'a set of external factors, directly or indirectly affecting the operations in this case governance*'. The Figure [Fig. 1) below shows

the macro environment of governance in which forecast can be deployed and the associated factors affecting the environment hence the variables the computer-based programme will take into consideration. Forecasting the future means pointing to the future goals, means and

methods of operation in order to achieve the forecast objectives.

When forecasting, one should not neglect current theories. Formal models of individual processes that occur in a deliverables should correspond to the theoretical assumptions which are connected with governance processes. Proper forecasts can be obtained only if a model that corresponds to theoretical assumptions is developed as in Fig. 2.

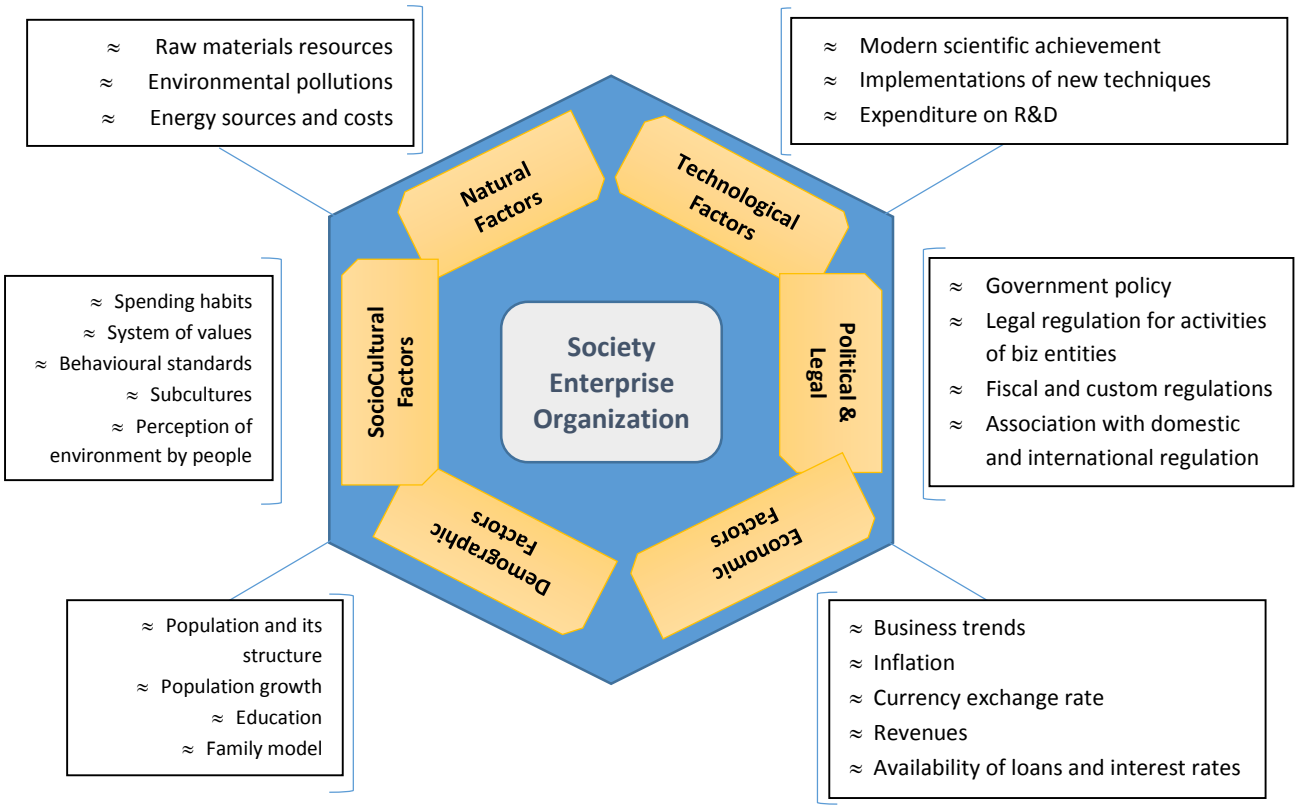

Figure 1: Components of Macro Environment [society, Organization, Enterprise]

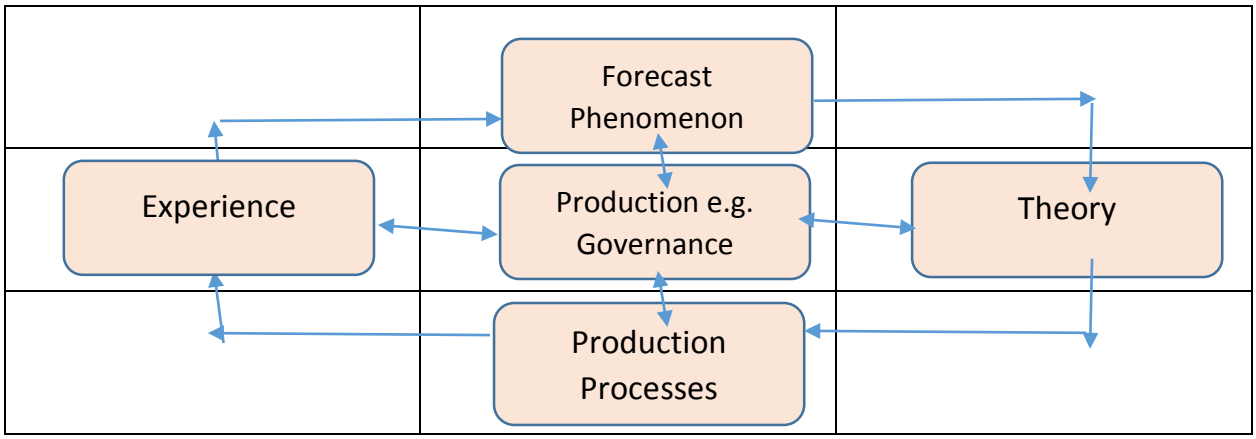

Figure 2: Using Theory in Forecasting

2.2 Data Angles of Forecasting

These are the two data angles of studying forecasting; these include qualitative and quantitative angles.

a. **Qualitative:** Qualitative data allows for *describing patterns* and *importance* of *factors* affecting particular events. It is subjective judgment of expert. Examples of tools here are Delphi, decision tree and Monte Carlo. It depends on conscious creation of past and on the ability of a forecaster to order and associate particular pieces of information. Qualitative methods use qualitative data. These methods are not objective, analytic and sensible especially at short-term forecasts when compared with the other method yet-to-explain.

b. Quantitative: Quantitative data are incidentally basis for analysis of phenomenon and processes of economy. These forecast uses mathematical, statistical, econometrics and optimization models. The condition for using it is availability of (historical) data. This technique is mostly referred as computer-based forecasting.

Whether quantitative or qualitative, forecasting models can be categorized into *time-series*, *causal* and *judgmental* forecasts. These categories are described in the summary box shown in figure 2.

Summary Box: Quantitative and Qualitative Techniques of Forecasting

Quantitative Forecasting Techniques:

- ≈ *Regression Analysis*: statically relates sales to one or more explanatory (independent) variables. Explanatory variables may be marketing decisions (price changes, for instance), competitive information, economic data, or any other variable related to sales.
- ≈ *Exponential Smoothing* makes an exponentially smoothed weighted average of past sales, trends, and seasonality to derive a forecast.
- ≈ *Moving Average* takes an average of a specified number of past observations to make a forecast. As new observations become available, they are used in the forecast and the oldest observations are dropped.
- ≈ *Box-Jenkins* uses the auto correlative structure of sales data to develop an autoregressive moving average forecast from past sales and forecast errors.
- ≈ *Trend Line Analysis* fits a line to the sales data by minimizing the squared error between the line and actual past sales values. This line is then projected into the future as the forecast.
- ≈ *Decomposition* breaks the sales data into seasonal, cyclical, trend and noise components and projects each into the future.
- ≈ *Straight-Line Projection* is a visual extrapolation of the past data, which is projected into the future as the forecast.
- ≈ *Life-Cycle Analysis* bases the forecast upon whether the product is judged to be in the introduction, growth, maturity, or decline stage of the life cycle.
- ≈ *Simulation* uses the computer to model the forces, which affect sales: customers, marketing plans, competitors, flow of goods, etc. The simulation model is a mathematical replication of the actual corporation.
- ≈ *Expert Systems* use the knowledge of one or more forecasting experts to develop decision rules to arrive at a forecast.
- ≈ *Neural Networks* look for patterns in previous history of sales and explanatory data to uncover relationships. These relationships are used to produce the forecast.

Qualitative Forecasting Techniques

- ≈ *Jury of Executive Opinion* consists of combining top executives' views concerning future sales.
- ≈ *Sales Force Composite* combines the individual forecasts of salespeople.
- ≈ *Customer Expectations (Customer Surveys)* use customers' expectations as the basis for the forecast. The data are typically gathered by a customer survey by the sales force.
- ≈ *Delphi Model* is similar to jury of executive opinion in taking advantage of the wisdom of experts. However, it has the additional advantage of anonymity among participants.
- ≈ *Naïve Model* assumes that the next period will be identical to the present. The forecast is based on the most recent observation of data.

2.3 Computer-Based Forecasting

Two major advantages of modern computers are *incredibly high speed* and *great accuracy* with which they can do calculations. Hence, any forecasting method can be programmed to run on a computer for this high speed and great accuracy gains. Even the most calculation-intensive methods can be run on micro-computer within a few minutes.

Finally, the whole model is put together and run as a *system of equations*. In typical small models there may be one or two dozen equations. Today, large systems forecast from a few hundred to tens of thousands variables.

After specifying the exogenous and policy variables (such as population, government spending and tax rates, monetary policy, etc.), the system of simultaneous equations can project important economic variables into the future.

2.3.1 Important Characteristics Which Determine Forecasting Methods

The six (6) important characteristics or dimensions of planning and decision-making which determine the choice of forecasting methods are the following:

1. **Time Horizon:** The period of time for which the decision is made will have an impact. It may be the immediate term (i.e., less than one month), short-term (up to 3 months), medium-term (up to-2 years) long-term (more than 2 years).

2. **Level of Details:** While selecting a forecasting method for a particular situation, one must know the level of details which will be needed for the forecast to be useful for decision-making purposes. The need for detailed information varies from situation to situation and time to time.

3. **The Number of Variables:** The number of variables to forecast affects the need for detail which, its turn, determines the choice of appropriate methods even in the same situation. When forecast is to be made for a single variable, the procedures used can be more detailed and complex than when forecasts are made for a number of variables.

4. **Constancy**: Forecasting a situation which does not change is different from forecasting a situation which is fairly unstable (i.e., a situation which often keeps on changing).

5. **Control Vs Planning:** The controlling function is performed by using a new technique called Management By Exception [MBO]. Any forecasting method must be sufficiently flexible so that the changes in the basic patterns of behaviour of variables or relationships among them can be detected at an early stage.

6. **Existing Planning Procedures:** For introducing new forecasting methods, often the existing planning and decision-making procedures have to be changed. Moreover, in case of any deviation from a set path it gives early warning and the managers face human resistance to such changes.

2.3.2 Examples of Computer-Based Systems for Handling Multiple Quantitative Forecasting: SIBYL & Others

A single forecasting method may not be suitable for all purposes and appropriate for all situations hence it is better to have separate computer programme [computer-based forecasting] for different methods similar to Holt's method. Additionally to this, there must be an overall control programme with a "*menu*" of alternative methods, to check the results of various methods and take corrective actions.

Various computer-based forecasting systems have been developed of which SIBYL1 is most useful and is used in most universities settings and business organizations.

The SIBYL-forecasting system is a philosophy for methodical forecasting and a computerized package of programme. These deal with simple applications at first and go into difficult problems thereafter. The SIBYL system provides software programme for dealing with the following four essential forecasting functions:

- ≈ Data preparation and handling of data.
- ≈ Screening of existing forecasting methods.
- ≈ Application of the methods chosen.
- ≈ Comparison, selection and combination of forecasts.

Item #1 deals with preparation of data files; data entry, data updating, transformation of data programmes and graphing.

Item #2 deals with selection of an appropriate forecasting technique for a particular purpose. This is done in the SIBYL programme. The user is given a list of methods which are suitable for a given situation and a summary of the characteristics of the given situation.

Item #3 deals with the application of the method chosen to the specific forecasting situation; the SIBYL package has 24 computerized subroutines of the most commonly used univariate and multivariate time-series and multiple regression techniques.

Item #4 deals with preparing and combining results obtained from alternative forecasting methods. Individual techniques are applied to a given situation and the results are automatically stored in the memory and recalled at the end of the programme. Thus computer-based system helps us to locate the best method for obtaining the most satisfactory results.

[11] New versions and batch versions are available which can be run on most large (main frame) computers, major time-sharing networks, mini computers and microcomputers like IBM and APPLE machines.

Others such as IBM SPSS, MatLab, Microsoft Excel, etc are available.

2.3.3 Weaknesses of Non Computer-Based Forecasting

The following are identified forecasting weaknesses for non-computer-based forecasting.

1. At project and programme level:

Poor Forecasts;

- ≈ often lack ranges and sensitivity analysis; without this information, decision-makers cannot manage risks effectively.
- ≈ as a result have led to poor value-for-money decisions.

2. At the aggregate level:

Poor forecast is;

- ≈ a result meant for opportunities to spend on worthwhile projects were missed.
- ≈ several root causes for government departments' poor production and use of forecasts
- ≈ implying governments make rapid allocation decisions to meet end of-year pressures
- ≈ able to erode confidence in forecasts generally. They can if consistently done:
    - authorized unplanned spending to utilize underspends;
    - offset overspends in one programme with underspends elsewhere;
    - carried forward underspends; and
    - be unable to reallocate underspends because these were declared too late.

2.3.4. Impacts/Implications of Poor Forecasts

Poor forecasting can cause avoidable differences between expectations and outcomes such as:

- ≈ Private sectors: poor forecasting can lead to lost market share, lower profits or even bankruptcy.
- ≈ Governance: poor forecasting can mean ill-informed decisions and taxpayers bearing the costs and poor delivery of services.
- ≈ Governance: poor forecast means that projects cost more, are completed later or produce fewer benefits than predicted, over/underspends can mean that opportunities to spend on worthwhile projects are missed.
- ≈ Governance: poor forecasting on one project can affect other projects in governments' spending portfolios, as budgets are varied to accommodate unexpected changes.

2.3.5. Major Important Factors To Considered In Computer-Based Forecasting

To deal with weaknesses of forecasting, it is important to consider these six factors;

1. **Time Horizon**: Two aspects of the time horizon are related to most forecasting methods and they are the span of time in future for which different methods are appropriate and that the number periods for which a forecast is required.
2. **Data Pattern**: For matching forecasting methods with the existing pattern of data (i.e., seasonal/cyclical, time-series/cross section etc.) an appropriate method is possible to be selected.
3. **Accuracy**: Forecasts must be as accurate as possible within the limit of human error.
4. **Cost**: In any forecasting procedure the following four costs are generally involved:
    - ≈ Development;
    - ≈ Data preparation;
    - ≈ Actual operation; and
    - ≈ Cost of foregone opportunity.
5. **Reliability**: Never forecast anything based on data which is not reliable for the purpose of decision making.
6. **Availability** of computer software: It is not that easy to apply any given quantitative forecasting method without an appropriate computer programme. Programmes must be "free" from major "bugs", well documented and easy to use, for getting satisfactory results.

2.4 Brief History of Forecasting

Prior to 1950s, there existed hardly any method for business forecasting talk less of computer-based forecasting or introduction of computers into forecasting. In the mid-1950s, light came: exponential smoothing technique was first used by the defense personnel for forecasting purposes. Subsequently, this technique was applied to business organizations beyond defense organization.

In the 1960s, the computer power became cheaper and techniques like multiple regression and econometric models were widely used to quantify and test economic theory with statistical data. As economics entered the age of computers in the 1970's the process was hastened by the availability of cheap computers.

In 1976, the Box-Jenkins method was developed. It is a systematic procedure for analyzing time series data. In truth, the Box-Jenkins approach to time-series forecasting was as accurate as the econometric models and methods.

In the 1960s and 1970s, technological forecasting methods were developed of which the Delphi method and cross-impact matrices were very popular.

However, in 1970s, it was first realized that forecasts were useless unless they were applied for planning and decision-making purposes.

3.0 METHODOLOGY & DESIGN

Main sources of data for *Significance and Application of computer Based Forecasting to Governance cum*

Leadership are both primary and secondary data. The primary source is interview and observation while the secondary source comprises of literature via textbooks and internet access and published reports. [See References.]. Few people knowledgeable and involved with government budgets, projects, planning and developmental and amenities distribution and executions were interviewed. The researcher observed these activities intensively in spite of the previous knowhow of the observation long ago. Similar interview and observations were extended to corporate organization where corporate governance is practiced. Literatures cum publications were used to support the primary data and to bring out the underlying principles therein. Next Section 4.0 will show that.

4.0 DISCUSSION

4.1 Significance of Computer-Based Forecasting to Governance cum Leadership

Effective forecasting for governance requires governance and administrations to recognize that forecasts are more than a technical activity, and emphasize their importance to financial and operational management of economy and governance. It is essential that government ministries, agencies and departments generate cooperation and understanding between the analysts who are involved in the production of forecasts, and their policy, operational and finance colleagues who use them to manage the business of governance [CFO Research Services, 2009]. Often, we identify problems with project-level forecasting, but these latest developments mean this is a good time to consider government forecasting holistically. But the only thing that can make this paradigm right is to make it system-based or computer-based. The significance are;

1. The computer system provides daily global exposure reports [a forecast], facilitating centralized exposure management, aggressive leading and lagging strategies and substantial savings on holding costs each year – a very impossible chore without a computer.
2. In today's highly competitive business world, firms strive to increase productivity and slash costs, in fact, a growing number of companies are instituting austerity programmes to cut layers of corporate management, especially on the international side - computers play a critical role in this effort.
3. By automating finance [forecast with computers inclusive], firms can reduce labour costs, and dramatically improve the speed and accuracy of many routine tasks.
4. Forecasting is an essential component of good financial management and informed decision-making. By the way, effective financial management [only through C-Based] is vital for sound decision-making, accountability, planning and managing risks.
5. Computer-based forecast in addition to putting in place the right processes and culture to support Quality Assurance can *forestall* high-profile ERRORS & or create an atmosphere/abating of FRAUD which can led to unforeseen costs and suspicion to taxpayers and crime hence prompting greater focus on quality and accuracy.
6. Poor forecasting is a *DEEP-ROOTED* problem, leading to poor value for money and taxpayers bearing the costs – which couldn't be with computer-based forecasting.
7. Poor forecasts of aggregated expenditure can lead to late identification of under/over-spending and rapid, poor value-for-money responses; other systems that cannot be as reliable as computer-based can promote this challenge.
8. Demonstrating excellent financial management – including accurate aggregate spending forecasts cannot be entirely judgmental but sound analytical and technical application of only computer-based forecasting can provide via programming.
9. In computer-based forecasting, changes to the budgetary system to encourage earlier and more transparent forecasting of future can easily be incorporated via the mathematical models and programming logics.
10. Governments use forecasts to consider new investment as well as whether existing initiatives need to be changed, terminated or resourced from elsewhere. Nowadays computer-based forecasts include projected:

    i. costs, such as the capital expense of building and maintaining a large infrastructure project;
    ii. demand for services;
    iii. staff resources to deliver a service; and
    iv. revenue receipts.

11. Robust computer-based forecasts of future demand and costs are an essential element of the financial management needed to plan and prioritize services effectively for the governed.
12. The need for accurate forecasting made possible by computer technology has increased with the difficult economic climate and cuts to government spending.
13. Forecast especially reliable and accurate one that can be facilitated by computers can help staff at all levels of an organization understand what is expected to occur and the range of uncertainty to inform planning and risk management.
14. Forecasts can reflect simple trend extrapolations, but ideally involve computer-based modelling and more complex quantitative analysis.

4.2 Applications of Computer-Based Forecasting to Governance and Leadership

Severally computer-based forecasting is deployed virtually in all areas of economic decision-making, statistics, accounting, management, marketing, etc. This researcher discussed few of the application areas here.

1. **Population:** population forecasting are achieved through computer-based forecast. Head count popularly called CENSUS done in developing nations are not yearly but periodically for instance once in 10years in Nigeria is continued as projection based on population growth rate factor. Same is where census is not done but registering of birth and death. Computer-based forecast is used to know the growth factor and number. These exercises would have been helpless if there is no computer-based forecast.
2. **Budget Appropriation:** annual national expected income and spending of government to the society is a forecast based on expected growth and development rate expected to achieve in the coming fiscal year. This estimate projects into the future expected income to make and expected expenditure. 100% full deployment of forecast can help governance.
3. **Developmental Projects:** distribution of developmental projects can use computer-based forecast to remove biases and arbitrariness in its modus operandi when distributing projects. The Computer-Based forecast model and or database should be able to know places that have received government projects and areas that remains to receive so that not some localities receive and others denied.
4. **Decision Support System [DSS]:** there is support for using Decision Support Systems (DSS) with the following issues addressed: techniques within the DSS, corporation needs and limitations, the forecast cost effectiveness, and the appropriate software system.
5. **Distribution of Amenities:** similar to projects computer-based forecasting is a strong tool that government that want to deploy e-governance use in distributing, knowing where to distribute and know where to focus government services and amenities. A typical DSS or a database system can be developed for this purpose.
6. **Sales**: Sales forecasting is an integral part of marketing DSS. The DSS contains tools to help the forecaster prepare better forecasts; tools are data, records of previous forecasting, and techniques. Sales forecast application can also be on standalone.
7. **Marketing**: Computer-based forecasts assist marketing managers improve decision-making.
8. **Organizational Design:** here, forecasting should not be regarded as a self-contained activity, but should be integrated within the planning context of which it is a part. In large matrix organization, accurate forecast can be a major success to OD.
9. **Planning**: this researcher believes that forecasting and planning functions should be combined largely. Involvement of the forecasters in planning enables them to select criteria for evaluating forecasting methods that are meaningful within the planning context.
10. **Operators' Expenses [Imprest] & DSS:** authors Rubinstein & Liddle (1997) stressed that restaurant operators must go beyond the typical spreadsheet software that only allows for tallying of operator expenses and does not include the technology of a DSS.
11. **Production Requirements:** Many software packages are available to restaurant operators that incorporate inventory management, purchasing and sales data, this assist restaurant operators in forecasting sales and production requirements.
12. **Supply Chain Management:** one means of an automated system in supply chain management in the restaurant industry is electronic data interchange (EDI). EDI is the computer-to-computer exchange of business transactions between companies. EDI is seen as a means to facilitate sales forecasting efforts by providing information that would pertain to a channel member's demand for the products and/or services offered by the supply channel member. In turn, the supply channel member, upon receiving this information, would respond with an update to production and/or distribution schedules in order to meet this demand.
13. **Customers Services [via POS]:** the POS [another system to manage sales forecasting in restaurants] system operates on the property level, with the capacity to be interfaced with regional and corporate systems to provide efficiency in the collection and transfer of sales data, inventory management, recipe maintenance, payroll, and many other functions. Hand-held server terminals, which are actually POS systems, allow servers to accurately enter orders that are linked to restaurant databases containing inventory and sales information. Nowadays, this technology in hardware and software is increasing customer service while decreasing inaccuracies in restaurant forecasting.

5.0 RECOMMENDATIONS AND CONCLUSION

5.1 Recommendations

The following recommendations are therefore made to authorities in respect of understanding the importance and applications of computer-based forecasting, helping them to deploy it;

1. Decision-makers need greater understanding of forecasts to provide effective challenge and manage risks.

2. When decision-makers need to introduce new interventions quickly they sometimes fail to recognize and manage the risks their non-computer-based forecasting creates for the quality of forecasts. This ought not to be so.

3. Effective forecasting for governance requires governments and administrations to recognize that forecasts are more than a technical activity, and emphasize their importance to financial and operational management of economy and governance.

4. 'Optimism bias' is a significant problem, with analysts concerned about the pressure to provide supportive which are subjective rather than realistic objective forecasts.

5.2. Good Practice In Forecasting: The Way Forward

The processes of producing and using forecasts must be well integrated, with shared understanding between all parties and capability to produce and use forecasts at project, programme and aggregate levels in order to drive effective decision-making and value for money for the taxpayer. Publics and organizations need the right incentives to maximize the benefits of forecasting. This requires a supportive environment within government ministry and across government, which promotes good practice and ensures accountability. Setting out our good practice framework for maximizing the benefits of forecasts as a leadership tool:

- ≈ When ***PRODUCING*** forecasts, high-quality data, skilled staff, well-reasoned assumptions and clear presentation of uncertainty are required.

- ≈ When ***USING*** forecasts, decision-makers need to understand the level of risk and uncertainty and the reasons behind this, to make informed decisions on how to allocate resources to deliver services on time and budget.

5.2 Conclusions

Today we all recognize the transformation that computers have wrought in the society, workplace and in our lives. In just few decades since the microcomputers brought new power to our desktops and workbenches, the changes have been stunning. A close look at public and corporate governance simply reveals that with information now moving from society to government house and from the factory floor to the company board at blinding speed, whole long layers of corporate management have been rendered obsolete.

People have now learned that the speed of today's more competitive environment does not leave time for dithering over decision, anyway. The resulting learner style has thinned management ranks while encouraging initiative and giving people more responsibility.

Like any powerful technology or invention, the computer leaves little room for sentiment. It has spawned an Information Revolution that promises even more profound changes that we have witnessed already. There can be no doubt that these changes, like those of the Industrial Revolution, will, on balance, provide great benefits.

The digital tide has already reshaped the business and governance world. Now it's spilling out of the office to touch every aspect of our lives. Today's software lets computers simulate workings of machines that don't exist yet. The traditional self-contained computer is merging into the collective identity of the network.

However, predicting the computer's effects seems perilous. As the most symbolic of all tools, it can be just about anything we programme it to be—a telephone switch, calculator, missile guidance system, or fantasy environment. That malleability is what stirs the imagination so strongly.

REFERENCES

CFO Research Services, (2009). *Driving Profitability in Turbulent Times with Agile Planning and Forecasting*. Prepared in collaboration with SAP and Deloitte, May 2009.

Cieślak, M., (2012). Prognozowanie gospodarcze – Metody I Zastosowanie, PWN, Warszawa, 2012, pp. 204.

Comptroller and Auditor General (2013). *Financial Management In Government*, Session 2013-14, HC 131, National Audit Office, June 2013; National Audit Office, *Evaluation in Government*, Dec. 2013.

Comptroller and Auditor General, (2014). *Forecasting In Government To Achieve Value For Money*, Report Ordered by the House of Commons, USA, Ed.: 30 January 2014

Dittmann, P., (2010). *Metody Prognozowania Sprzedaży W Przedsiębiorstwie*, Wyd. AE, Wrocław 2000, p. 13-14 panel of OECD countries, Energy Policy vol. 38 (2010), pp. 656–660

Ihezuo, M.O. (2016). Leadership Is Everything, Soteria Publisher, Niger Road, Port Harcourt

Lucjan K., (2012). Importance Of Forecasting In Enterprise Management, *Advanced Logistic Systems, Vol. 6. No. 1. (2012), pp. 173-182.*

Lucjan, Kurzak (2012), Importance Of Forecasting In Enterprise Management, *Advanced Logistic Systems Vol. 6. NO. 1. (2012) PP. 173-182*, Poland.

Nau, R., (2014), accessed online on 20/02/2017 at *www.people.duke.edu/~rnau/forecasting.htm*

Nowicka-Skowron, M.; Dima, I. C.; Man, M.; Grabara, I., (2011). Econometric Patterns And Methods Used For Analysis Of Technological Innovations In Workshops And Production Departments Equipped With Flexible Manufacturing Systems, *Polish J. Management Study, Vol.3, W.Z.PCz., Częstochowa 2011, pp.7-31*.

International Journal of Computer Applications Technology and Research
Volume 6–Issue 4, 209-212, 2017, ISSN:-2319–8656

Unstructured Datasets Analysis: Thesaurus Model

Parvathy Gopakumar
Department of Computer Science and Engineering
Mangalam Engineering College
Kottayam,Kerala,India

Neethu Maria John
Department of Computer Science and Engineering
Mangalam Engineering College
Kottayam,Kerala,India

Vinodh P Vijayan
Department of Computer Science and Engineering
Mangalam Engineering College
Kottayam,Kerala,India

Abstract: Mankind has stored more than 295 billion gigabytes (or 295 Exabyte) of data since 1986, as per a report by the University of Southern California. Storing and monitoring this data in widely distributed environments for 24/7 is a huge task for global service organizations. These datasets require high processing power which can't be offered by traditional databases as they are stored in an unstructured format. Although one can use Map Reduce paradigm to solve this problem using java based Hadoop, it cannot provide us with maximum functionality. Drawbacks can be overcome using Hadoop-streaming techniques that allow users to define non-java executable for processing this datasets. This paper proposes a THESAURUS model which allows a faster and easier version of business analysis.

Keywords: Hadoop;MapReduce;HDFS;NoSQL;Hadoop-Streaming

1. INTRODUCTION

Data has never been more important to the business world as it has become a vital asset as valuable as oil and just as difficult to mine, model and manage. The volume and veracity of the datasets that are being stored and analyzed by the business are unforeseeable and the traditional technologies for data management such as relational databases cannot meet the current industry needs. Bigdata technologies play a vital role to address this issue. Early ideas of big data came in 1999 and at present it becomes an unavoidable phenomenon tool through which we manage business and governance. For a layman the idea of Bigdata may relate to images of chaotic giant warehouses over crowded office space with numerous staffs working through huge number of pages and come with boring formal documents under supervision of some old bureaucrat. On the contrary working of Bigdata is simple and well structured, yet exciting enough to pose new challenges and opportunities even to experts of industry. It provides parallel processing of data in hundreds of machines that are distributed geographically. Necessity of Bigdata arises under the obligation of the following:

1. When existing technology is inadequate to perform data analysis.
2. In the case of handling more than 10TB of dataset.
3. Relevant data for an analysis present across multiple data stores which are filed in multiple formats.
4. When steaming data have to be captured, stored and processed for the purpose of analysis.
5. When SQL is inefficient for high level querying.

In today's data centered world Hadoop is considered as the main agent of big data technology due to its open source nature. However as it is a java based ecosystem, it created hurdle for programmer from non-java background. To address this issue it has facilitated a tool, 'Hadoop-Streaming' by enabling flexibility in programming with effective parallel computability.

2. PROBLEM STATEMENT

Why Big data analysis? Well, it helps the organization to harness their transactional data and use it to identify new opportunities in a cost effective and efficient manner. Primary aim of data analysis is to glean actionable logic that helps the business to tackle the competitive environment. This will alert the business for their inevitable future by introducing new products and services in favor of the customers. Unfortunately for the matter of convenience 80% of the business oriented data are stored in an unstructured format. Structured data usually resides in a relational database with predefined structures so converting the data to different models and analyzing them seems mundane. Here the role of Hadoop-Streaming arises which works on a Map and Reduce paradigm by analyzing the unstructured data and presents viable business logic.

The aim of the paper is to:

- Study existing framework employed by industry players.
- Present a new roadmap for efficient and effective approach to Bigdata problems: THESAURUS MODEL

3. BACKGROUND

3.1 Structured Vs Unstructured datasets

The question that encounters a rookie is that why one uses unstructured dataset when there is always a possibility of using structured data. At the outset of computing, the term storage corresponded only plain texts. Now user needs to store richer content than plain text. Rich data type includes pictures, movies, music, x-rays ,etc.It provides superior user experience at the expense of storage space. Such data sets are called unstructured because they contain data that do not fit neatly in

a relational database. Industry came up with a third category called semi structured data which resides in a relational database, similar to structured data. However it does not have some organizational property necessary to make them easy to be analyzed.(Eg.XML doc)

3.2 NOSQL Data store

A NOSQL database [4] provides mechanism for storage and retrieval of data which is modeled in contrast to the tabular relations used in relational databases. It become common in the early twenty first century when the industrial requirements triggered a need of database structures that support query languages other than SQL.(called "Not only SQL", non SQL).This is mostly used in big data and real-time applications as it provides simpler design, horizontal scalability and high availability. The most popular NOSQL databases are MongoDB, Apache Cassandra [3], Datastax, Redis.

3.3 Hadoop & Hadoop Streaming

Apache Hadoop [1] is open source software for reliable, scalable and distributed computing. Hadoop framework allows distributed processing of large datasets across low level commodity hardware using simple programming models. This framework is inspired by Google's MapReduce structure in which application is broken down into numerous small parts and each part can be run in any node in the cluster. Hadoop contains two major components - a specific file system called Hadoop Distributed File System (HDFS) and a Map Reduce framework. Hadoop works on divide and conquer principle by implementing Mapper and Reducer in the framework. Mapper function splits the data into records and converts it into (key,value) pairs. Before feeding the output of the Mappers to Reducer an intermediate Sort and Shuffle phase is implemented in the MapReduce framework to reduce the work load at Reducer machine. The sorted (key,value)pair is given into Reducer phase. The Reducer function does the analysis of the given input and the result will be loaded to HDFS(eg.The maximum temperature recorded in a year, positive and negative ratings in a business etc.).The analyst has to develop Mapper and Reducer functions as per the demand of the business logic.

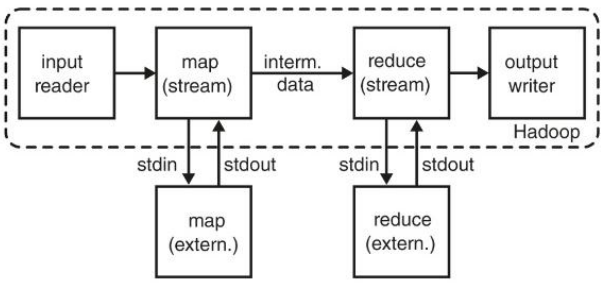

Figure 1 Hadoop-Streaming

Hadoop Streaming (see Figure 1) is an API provided by Hadoop which allows user to write MapReduce functions in languages other than java[2]. Hadoop Streaming uses Unix standard streams as the interface between Hadoop and our MapReduce programs, so the user has the freedom to use any languages (Eg. Python, Ruby, Perl etc.) that can read standard input and write to standard output.

4. ANALYSING UNSTRUCTURED DATASETS USNG HADOOP-STREAMING

Due to the difficulties in analyzing the unstructured data organizations have turned to a number of different software solutions to search and extract prerequisite information. Regardless of the platform used, the analysis must undertake three major steps– data collection, data reduction, data analysis [7][8][9][10](see Figure 2):

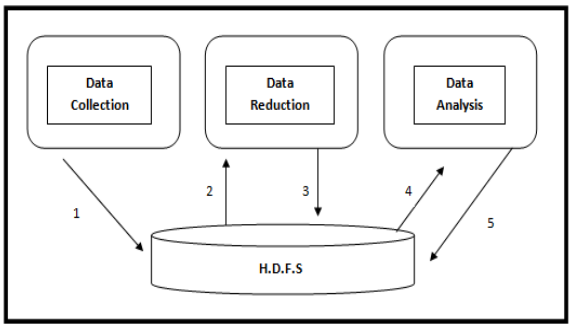

Figure 2 Analyzing Unstructured Dataset

A. Data Collection: In this stage the datasets to be analyzed can be collected through two methods. Firstly, data can be downloaded from different nodes containing the specified records to HDFS. Alternatively it can be done by connecting to the local servers containing the records. The former can be achieved by tools such as Sqoop, Flume and the latter using Apache Spark[6]. In a real time environment the streaming datasets can be accessed using standard public key encryption technique to ensure authenticity.

B. Data Reduction: Once the unstructured dataset got available, analysis process can be launched. It involves cleaning the data, extracting important features from data, removing duplicate items from the datasets, converting data formats, and many more. Huge datasets are minimized into structural and more usable format using series of Mapper and Reducer functions. This is done by projecting the columns of interest and thus converting it in a format which will be adaptable for final processing. Cleaning text is extremely easy using R language, whereas Pig and Hive supports high level abstraction of data preprocessing.

C. Data Analysis: Before the inception of Bigdata technologies collecting, preprocessing and analyzing terabytes of data was considered impossible. But due to the evolution of Hadoop and its supporting framework the data handling and data mining process seems not so tedious. Programmer with the help of Hadoop Streaming API can write the code in any language and work according to the domain of user. In this stage the pre processed data is studied to identify the hidden pattern. Hadoop provides a Mahout tool that implements scalable machine learning algorithms which can be used for

collaborative filtering, clustering and classification .The analyzed data then can be visualized according to the requirement of the business using Tableau, Silk, CartoDB, Datawrapper.

Thus the whole process of analysis can be explained in a five step workflow:

1. Collecting the data from alien environment and keep it inside the Hadoop Distributed File System.
2. Apply set of MapReduce tasks to the step one collected data and project the columns of interest based on the user query.
3. Keep the preprocessed data in HDFS for further analysis.
4. Use the preprocessed data for analyzing the pattern of interest.
5. Store the result in HDFS so that with the help of visualization tools user can selectively adopt the method of presentation.

5. MODIFICATION OF EXISTING SYSTEM: THESAURUS MODEL

The underline motivation behind this model is the lack of knowledge base in the existing analysis framework which in turn causes the system to follow some unnecessary repetition. Consider an analysis problem to find the maximum recorded temperature in last 5 years. So the analysis is done by

1. Collecting the data from National Climatic Data Center [5] and store in HDFS.
2. Project the field which contains the temperature data i.e. the column of interest.
3. Store the preprocessed result in HDFS.
4. Find the maximum temperature reported by analyzing the (key, value) pair.
5. Store the final result in HDFS.

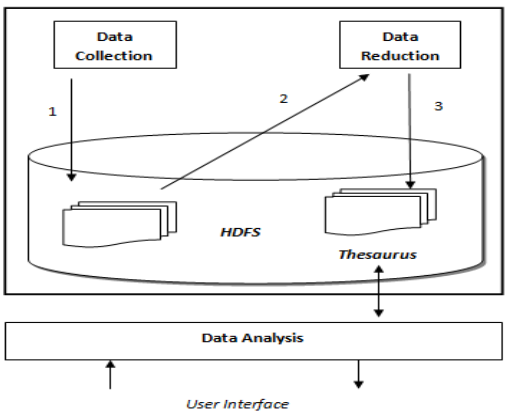

Figure 3 Thesaurus Model

So the maximum temperature of the year is accessed from the file system and can be used for monitoring and reporting purposes. Later if the same analyst needs to find the maximum humidity reported, he has to go through the whole datasets and has to bear the trouble of preprocessing and reducing the data again. This can be avoided by using Thesaurus model. According to this module, minable information are logically arranged and kept in the HDFS so that the future request for the information retrieval can be done in no time. Once the data set is converted into a structural format the schema of the dataset should be specified by the preprocessing programmer so the analyst need not come across the trouble of understanding the newly created data set.. This preprocessed datasets can replace the old datasets so that the unnecessary storage issue is taken care of by the model. The working of the system is specified in two phases, one for collection and preprocessing, and second for analysis. In the first phase the necessary data which can be analyzed are collected and preprocessed. This data is then stored in the thesaurus module in HDFS and made it available for the user to analyze based on the industry needs. Thesaurus not only contains the structured data but also the schema of the data storage. In phase two, the required query can be addressed by referring the schema .Thus analyst need not consider the problems of unstructured data collected by the system. The Figure 3 represents the work flow of Thesaurus model.

1. Collect the data from distributed environment and store in HDFS.
2. Use the stored data for preprocessing.
3. Store the preprocessed data in Thesaurus with a predefined schema. To avoid the storage bottleneck the data that are collected on the first place can be removed as it is no longer necessary.

6. CONCLUSION & FUTURE SCOPE

Mining the inner pattern of business invokes the related trends and interests of the customers. This can be achieved by analysing the streaming datasets generated by the customers in each point of time.Hadoop provides flexible architecture which enables industrialist and even starters to learn and analyse this social changes.Hadoop-Streaming is widely used for sentimental analysis using non-java executables.Also proposed a THESARUS model which works in a time and cost effective manner for analysing these humongous data. Future scope is to enable the efficiency of the system by developing a THESARUS model which is suitable to analyse terabytes of data and returns with the relative experimental results.

7. ACKNOWLEDGMENTS

I would like to thank Almighty, my professors, friends and family members who helped and supported in getting this paper work done with in time.

8. REFERENCES

[1] Apache Hadoop. [Online]. Available: http://hadoop.apache.org

[2] Apache Hadoop-Streaming. [Online]. : http://hadoop-streaming.apache.org

[3] Cassandra wiki, operations. [Online]. Available: http://wiki.apache.org/cassandra/Operations

[4] NOSQL data storage [online]: http://nosql-database.org

[5] National energy research scientific computing center. [Online]. Available: http://www.nersc.gov

[6] Apache Hadoop [Online]: http://spark.apache.org

[7] E. Dede, B. Sendir, P. Kuzlu, J. Weachock, M. Govindaraju, and L. Ramakrishnan, "A processing pipeline for cassandra datasets based on Hadoop streaming," in Proc. IEEE Big Data Conf., Res. Track, Anchorage, AL, USA, 2014, pp. 168–175.

[8] E. Dede, B. Sendir, P. Kuzlu, J. Weachock, M. Govindaraju, L. Ramakrishnan, "Processing Cassandra Datasets with Hadoop-Streaming Based Approaches", *IEEE Transactions on Services Computing*, vol. 9

[9] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu, and G. Fox, "Twister: A runtime for iterative mapreduce," in Proc. 19th ACMInt. Symp. High Perform. Distrib. Comput., 2010, pp. 810–818.

[10] Z. Fadika, E. Dede, M. Govindaraju, and L. Ramakrishnan, "MARIANE: MApReduce implementation adapted for HPC environments," in Proc. 12th IEEE/ACM Int. Conf. Grid Comput., 2011,vol. 0, pp. 1–8.