KHAZAR UNIVERSITY

Faculty: Engineering and Applied Sciences

Department: Computer Science

Specialty: Computer Engineering

MA THESIS

Theme: Enhanced Selective Fault-Tolerant Task Clustering Method using Database

Master Student: Zahra Khudaverdiyeva

Supervisor: Ph.D. Mahammad Sharifov

BAKU - 2016

KHAZAR UNIVERSITY

ENGINEERING AND APPLIED SCIENCES

COMPUTER SCIENCE DEPARTMENT

ABSTRACT

OF DISSERTATION FOR MASTER'S DEGREE

THEME

Enhanced Selective Fault-Tolerant Task Clustering Method using Database

Master student: Zahra Khudaverdiyeva

Supervisor: Ph.D. Mahammad Sharifov

Abstract

In computer science for reducing execution overhead and increasing the computational granularity of workflows for executing of tasks one method is proved as the best effective approach. This approach is called task clustering.

Nevertheless, as you know a job collected of multiple tasks in which in this way may have a paramount risk of troubling from failures than a just one task job. In this dissertation, firstly I demonstrated theoretically main aspects in which conducted with my thesis.

I further propose 1 fault-tolerant clustering strategies (Enhanced Selective Fault-Tolerant Task Clustering Method using Database) in which this method was enhanced according to Selective task clustering method and added checkpointed to restrict amount of retired jobs. As you know clustering help us to improve the runtime performance of workflow executions in faulty execution environments. From the proved works, experimental results illustrates that failures may have important effect on running when task clustering methods is used. This solutions product makespan progress in such approaches. Moreover, with my method strategy not just only to optimize the workflow’s makespan by adjusting the clustering granularity when failures are available. Excepts this, The main goal of my method to provide dynamic, stable system in which using database we can prevent loss of information about status of tasks, execution of tasks will be exact, failed and successful task will be known and log of tasks can be investigable.

Executive Summary

Enhanced Selective Task Clustering Method using Database

Subject matter: This dissertation provides broad detail about new "Enhanced Selective Fault-Tolerance Task Clustering Method using Database" in which was written in Java. In this approach, we can achieve not just only improvement of system performance also we will get dynamic, stable system for researching of failures, decreasing cost and waste of resources.

Introduction: As you know, using of large-scale distributed systems, such as Grids and Clouds are more widespread. The main problems of such systems are performance and system overheads. That's why from that time researchers try to find methods for increasing performance of workflow and decreasing system overheads. General methods is task clustering to achieve all above noted features. However, in all this approaches researchers try to find impact of failures on distributed resources for improving workflow performance, to decrease system overheads. In addition, dynamic and fault-tolerance clustering methods for scientific workflows also effect improvement of performance on distributed resources. Various techniques have been developed to manage with the negative effect of job failures to scientific workflows execution. The most common way is to repeat the failed job again. But this approach also can be expensive and waiting of resources because of recomputed tasks.

Methodology: In this dissertation paper, my aim is not just only to increase performance of systems using fault tolerant clustering method. Main goal is using new methods to achieve more dynamic and stable system in which with this way we can decrease cost and wasting of resources. Moreover, helping of this methods we can determine list of successful and failed tasks. After determining failed tasks we may investigate reason of failed tasks. In this paper, I used selective clustering method and add other functionalities to it such as checkpointed for restricting amount of retired task and achieved goals in which noted above.

Findings: Dissertation specific research and this dissertation demonstrate significant advantages of Enhanced Selective Fault-Tolerant Task Clustering method using Database. There are more than one and some of them are: increase performance of system in compression with other common clustering method, more tolerance in faulty distributed resources, decrease cost and wasting of resources, dynamic and stable systems, controlling failed tasks and so on.

Conclusion: Although Enhanced task clustering brings important preferences with it, but it is still depend on customer requirement. Because in some systems performance more important for them, in another systems don’t want to spend more revenue, in others want to get information about failed task for reestablish it again. That’s why usefulness of my method depend on customer requirements.

Table of Content

Acknowledgement

I would like to take this chance to say my thanks to the below people or group:

- My Supervisor – For always believing me, for don’t allowing me to underestimate myself, being confidence about me, supporting the ideas, giving right direction and supporting me in every possible way.
- My Mother – For believing and being my side and helping me in every time spiritually
- My Boyfriend – For always loving me, helping me always for fighting against any difficulties.

1. Introduction

Scientific workflows may be collection of many fine calculation granularity tasks, in which runtime of the task may be less than the system overhead. The period of time within which heterogeneous work other than the user's computation is accomplished. Task clustering procedures mix several short tasks into a one job like that the job runtime is improved and system general overhead is diminished. Task clustering is the most general method used to appeal execution overheads and rise the computational granularity of workflow tasks run on distributed resources. But, strategies of existing clustering methods focused increasing of system performance, estimate the impact of failures on the system, despite investigating reason of failures on large-scale distributed systems, such as Grids and Clouds.

A clustered job means that multiple tasks are collected into one single job. If a task during a clustered job fails (for instance, is destroyed by unsuspected events during its executions), the job is signed as failed, although tasks inside the same job have successfully finished their own execution. Different techniques have been improved to eliminate with the negative impact of job failures on the running of scientific workflows. The most general technique is to repeat execution of the failed job. But, retrying a failed clustered job may be more costly since finished tasks in the job usually is necessary recalculation of it, thus resource cycles are wasted. Furthermore, there is no guaranty that retried tasks will succeed. As another approach, jobs can be replicated to escape failures especial to a worker node. But, job replication can also waste resources, in especially for long running time of jobs. To diminish wasting of resource, job executions can be occasionally checkpointed to restrict the amount of retried work. However, the overhead of performing checkpointing can limit its benefits.

In this dissertation paper, I propose 1 enhanced selective fault-tolerant task clustering method using with database beside to upgrade existing task clustering methods in a faulty distributed resources environment, to create dynamic and stable system, to decrease cost and utilization of resources. This methods are created on the basis of selective fault tolerance task clustering and adding checkpointed helping of database. The selective method retries failed tasks within a job by extracting them into a new job. But using checkpointed we can give one condition to restrict the amount of repeating of failed task. Using this new method we can also keep the good performance of system, in addition to this superiority we can reduce revenue, utilization of resources and learn information about failed task and so on.

The paper consists of 4 parts. First part is an introduction about the new – Enhanced Selective Fault Tolerant Task Clustering Method using Database. It is about the advantages of this approach and what made it real for now. Second part is about background and definitions in which are related with my dissertation theme. There are base information and the briefly explanation of some terminology and methods. Third part gives an overview of the related work. In here about is which works was done about this topics such as clustering, fault tolerance systems, distributed resources and others. Fourth part is about implementation side, the algorithm of new method, results, benefits and other factors. After all this parts, paper closes with conclusion. End of dissertation paper consist of bibliography list.

2. Background and Definitions

2.1 Workflow

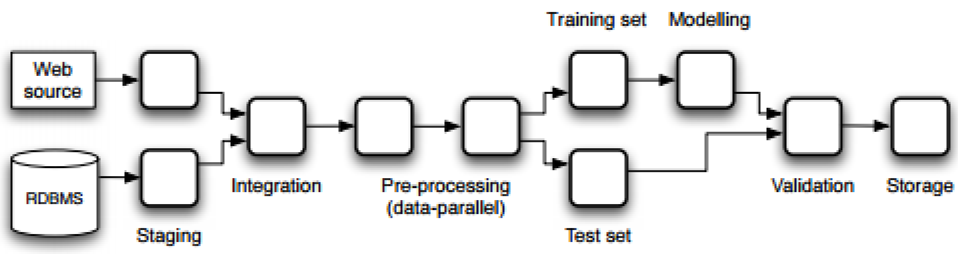


Figure 1: Workflow example

Unofficially, a workflow in other word he sequence of actions (See Figure 1: Workflow example on page 3) an abstract description of the steps required to perform specific real processes and information flows between them. Each step has a number of events that must be made is identified. In the working process, the business (eg, data or business) from start to finish through a variety of steps, and each step in the procedure established by the activities of people or system functions (such as computer programs) are performed by. Workflows are usually visual front-end to be written or encoded using the workflow engine and call their delegated executive functioning remote administration. Traditionally, document management systems for two large families, one control the orchestration of business processes and functional style of the other data separated for the calculation. However, many applications do not need to fit into any of these categories. This layer acts as a scientific research process and, as a rule, as the data management and workflow features of scientific workflow systems, was the cause of evolution. Their calculation functions and cooperation between the abstract and researchers about data sources, allow a task that requires both ways. We are interested in the issue of (scientific or non-scientific) of any one document management system, but if he could count on to cover the entire range of requirements from different areas. [1]

Workflows, has become an efficient paradigm for solving complex scientific and business applications. High performance of computing system, networking and cloud enabled scientists and engineers is enough space for better access and more complex applications and processes for the implementation of large data warehouses, as well as in distributed computing silicon platform for scientific experiments. Most of the data analysis of this application is designed as a scientific calculation methods and workflows involve complex modeling techniques.

Many scientific applications require a high level of design and implementation of tools and mechanisms. Simple and complex workflows are often used for this purpose. Thus, in recent years, many efforts have been focused on the development of a distributed process control system of scientific applications. Workflows cannot hide the low-level details that define the basis for a high-level application design application logic allows you declaratively. They are also used in scientific discovery process complex compositions, existing programming procedures, data sets, and the services can be integrated. [2]

Workflow definition, development and implementation of the so-called workflow management systems (WMS) are supported. The workflow (or adoption) coordinates the activities of certain activities during WMS is a key function of the working process.

Comprehensive use of scientific methods and computer technology and the process of scientific discovery, science, research, scientific methods, calculation methods, and so-called e-science paradigm established has changed significantly with the use of a new data analysis strategy has gone into a new era under discussion in [3]. For instance, Pan-STARRS astronomical research uses workflows to test the Microsoft Trident Scientific Workflow Workbench to install about 30 TB per year and employee perception of the telescope. Similarly, the USC Center for epigenome mapping epigenetic status of the time, human cells (for every week up to 8 billion nucleotides) for creating high-capacity DNA sequence data system of genetic analysis Illumina (GA) with abuse of Pegasus workflow systems genome scale. In this scenario, scientific workflows older complex scientific applications demonstrate its effectiveness as an effective programming paradigms in a super generic term or grids, ad-hoc network systems, as well as on a distributed computing infrastructure as a cloud. [2]

2.1.1 Workflow Programming

Workflow management systems are one of the main problems that they have to provide a scientific application developers build software applications. Some systems also provide textual programming interfaces, while others are based visual programming interface. This suggests two different interfaces of various programming approaches. This section discusses the current programming approaches used in scientific document management systems and compare them. As a rule, the scientific workflow of various data and Java, C ++, and including predefined procedures programmed in a programming language such as Perl has been programmed as a graphics processing unit. According to this approach, the methodology for the creation of pre-defined programs, scientific workflow to perform a single task, but at the same time forms require more resources to run a whole and can take a long time to complete the representation of complex application. For an introductive reference, see [4], it gives a classification of the most important features of scientific workflows. In fact, astronomy, bioinformatics or a few days or weeks required for the completion of [5], for example, scientific process is not uncommon

Workflow Tasks various application scenarios that help meet the needs of a wide range of designers, a number of different designs can be created with the following steps. Dependence of the flow control between the descriptions of the tasks focused on the extensive collection of workflow patterns has been described in [6]. A workflow management system commonly used in most programming constructs a directed acyclic graph (DAG) (see Figure 2 on page 6) or its extension that joins loops, that is the directed cyclic graph (DCG).

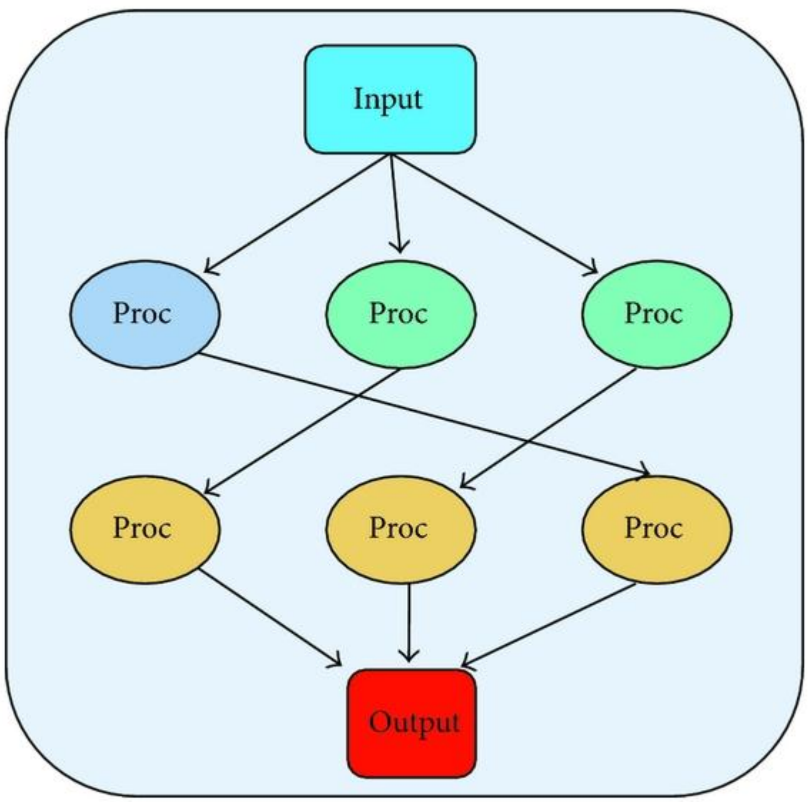


Figure 2: Example of DAG view.

2.1.2 Workflow Management Systems

Workflow Management Systems are software environments supply equipment to, execute, compose, map, and define workflows. There are many Workflow Management Systems on the market, most of them defined to a special application domain, like in our situation of scientific workflows [7] [8]. In this part I center on some important WMSs improved by the research society, with the aim of detecting the most significant characteristic and resolutions in which have been suggested for workflow management in the scientific field [9] [10].

Even though a standard workflow language like Business Process Execution Language (BPEL) [11] [12]has been identified, scientific workflow systems frequently have sophisticated their own workflow pattern for allowing users to demonstrate workflows. Apart from BPEL, alternative formalisms such as UML, Petri nets [13], and XML-based languages [14] [15] have been used to illustrate workflows. All this characteristics makes inconvenient the participation of workflow standards and borders interoperability between workflow-based applications flourished by using several workflow management systems. On the other hand, there are some diachronic causes for that, workflow systems and their workflow models like other scientific managements systems were improved prior to BPEL endured [16].

Generally workflow managements system can be divided into below two major categories [17].

a) Script-based systems, whereas workflow specifications indicate workflows by virtue of a textual, literal programming language in which can be depicted by a grammar in a similar way to conventional programming languages such as Java, Ruby, or Perl. All this language often exist complicated a spacious syntax and semantics. These sorts of descriptions clarify tasks and their standards, parameters by a textual characterization. In other words data dependencies can be determined between them by means of annotations. These languages include special design workflow as sequences or loops, while making or parallel design in order to create a workflow. GridAnt [18] and Karajan [19] can be example of script based workflow management systems. A widely used approach is based on scenarios to describe the work processes, primarily in the community business workflow, BPEL and is the latest version of Web Services, which is based on Web Service Flow Language of IBM, WSFL.

b) System where the working process models define the workflow with just a few basic graphic elements, which correspond to the components of the graph, such as nodes and edges. This type of workflow management system is named graphical-based systems. Graphical-based systems are easier to use and more intuitive for an inexperienced user primarily because of system graphical representation in comparison with script-based descriptions. In graphical-based systems nodes characteristically demonstrate workflow tasks in which communications (or data dependencies) amid several tasks are illustrated as links acting from one to another. Workflow system support based on graph models often include graphical user interfaces that allow users to model workflows by means of dropping and dragging the graph elements. Merely graph-based workflow explanations commonly use Directed acyclic graph (DAGs). As mentioned earlier, directed acyclic graph-based languages proposal a restricted expressiveness, like that they cannot demonstrate complicated workflows (for example, DAGs cannot describe directly loops).

2.1.3 Taverna

Taverna [20] [21]is a workflow management system as an open-source Java-based workflow management system improved at the University of Manchester. The basic aim of Taverna is sustaining the real sciences community (biology, chemistry, and medicine) to model and perform scientific workflows and maintenance in silico experimentation, whereas research is executed across computer imitation simulations with models nearly expressing the real research world. In spite of many Taverna applications fabricate in the bioinformatics field, it can be applied to a broad range of domains since it can invite any web service by solely providing the URL of its WSDL document. This property is very prominent in empowering users of Taverna to reutilize code (represented like a service) in which is existing, available on the internet. For this reason, system is open to third-party legacy code, providing the opportunity to interact with web services. (See Figure 3 on page 9)

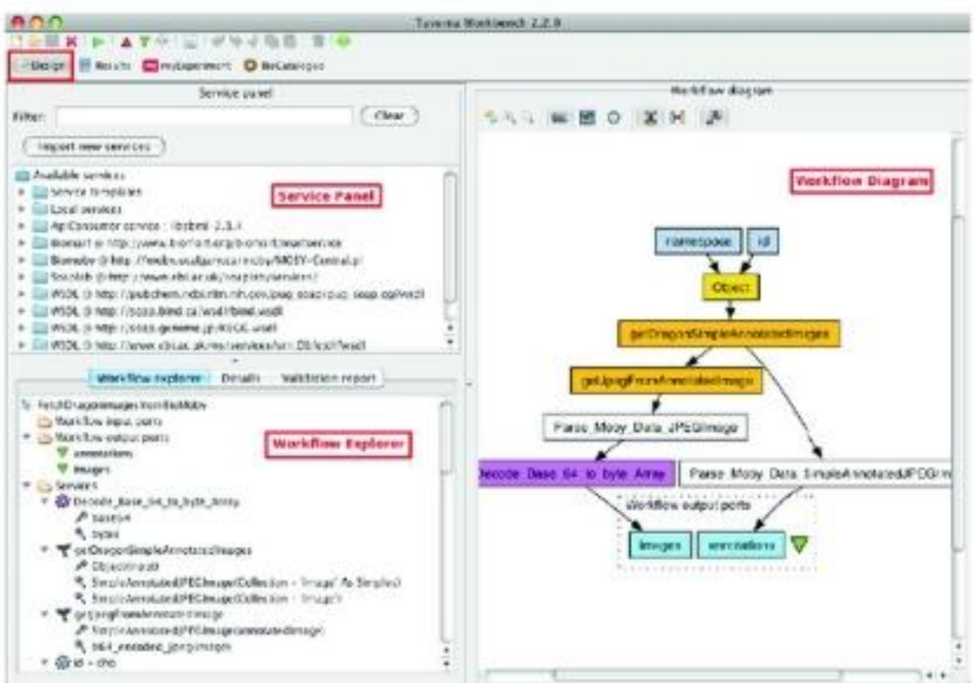


Figure 3: A view of Taverna Workbench.

2.1.4 Triana

Triana [22] [23] [24] is also a Java-based workflow system, developed by university staff of Cardiff University, which unions a demonstrative, visual interface with tools of data analysis. It can create connection among heterogeneous tools (e.g., JXTA services, Java units, and web services) in one workflow. Triana uses its own language of the user workflow, although he may use other representations of the external document language, such as the BPEL, which are available, open source through writers and readers of plug-language. Triana supplied with broad range of built-in tools for signal analysis, image processing, desktop publishing, and so on, and has the ability for users to easily integrate their own tools.

Framework of the Triana is based on a modular architecture in which the GUI a Triana engine was connected by the Graphical user interface (GUI), whereas it is called Triana Controlling Service (TCS), either remotely or locally. A client can sign in into a TCS, compose and execute an application remotely, and after visualize the outcome locally. Application may also be execute in batch mode; in this situation, a client can periodically registry back to check, monitor the status of the application. (See Figure 4 page on 11)

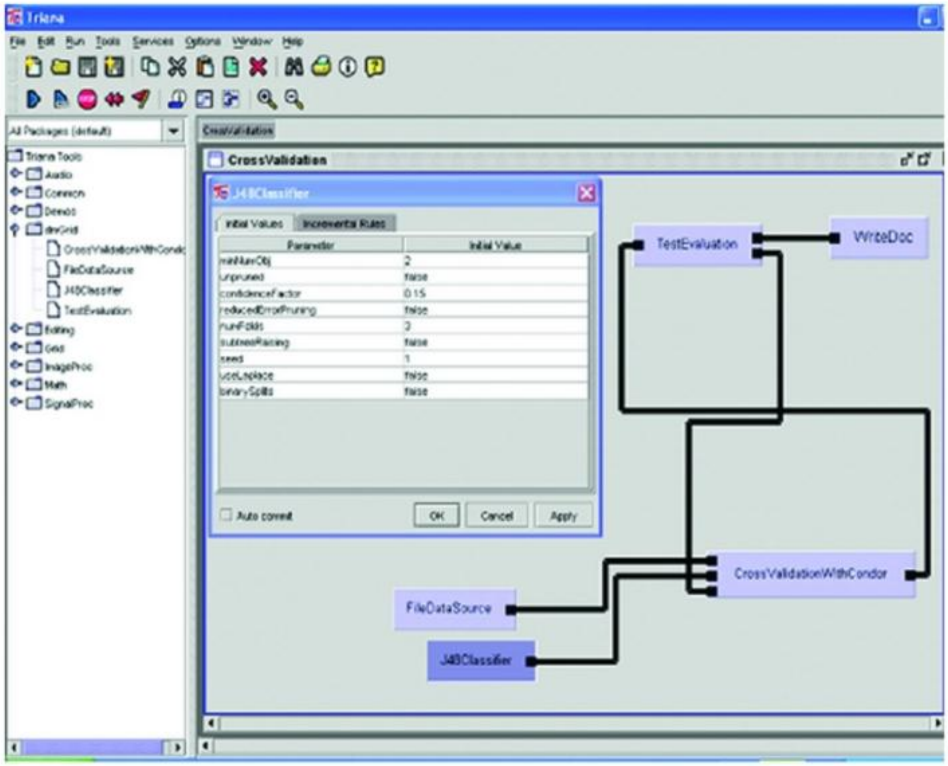


Figure 4: A view of the Triana GUI for data analyses.

2.1.5 Pegasus

The Pegasus workflow management system [61], created and improved in University of Southern California by means of student and teach staff, interpolates a series of technologies to run applications like workflow-based in a number of several environments, with the inclusion of desktops, Clouds, Grids, and clusters. Pegasus has been used in several scientific fields including wave physics, astronomy, ocean science, gravitational, bioinformatics and earthquake science.

The Pegasus workflow management system may control the running of an application packaged like a workflow by pattering it onto tasks of executing workflow and available resources according to dependency of applications (see Figure 5 on page12). Whole the computational resources and input data important for workflow running are mechanically, automatically situated by the system.

Pegasus also covers an advanced failures recovery system that tries to rescue from errors by trying again tasks or the entire, whole workflow, by redesigning sections of the workflow, by provisioning workflow-level checkpointing, and by using alternative, another data sources, when feasible. Eventually to ensure that the workflow is to be played, information about the origin of the recording system, including the location of the data used and produced in which software was used with this parameters.

Figure 5: Overview of The Pegasus WMS

2.1.6 Askalon

Askalon is also one of the scientific workflow management system. It is an application progress and runtime environment, educated at the University of Innsbruck, which admits the running of distributed workflow applications in service-oriented Grids [25]. Its uses Glous Toolkit like Grid middleware because it is SOA-based runtime environment.

Workflow applications in Askalon are characterized at a top level of abstraction using a convention XML-based language in which it called Abstract Grid Workflow Language (AGWL) [26]. AGWL lets users, clients to focus on designing of scientific applications without dealing with the complication of the Grid middleware (software that acts as a bridge between an operating system or database and applications) or any special application technology like Grid and Web services, software components, or Java classes. Activities in AGWL can be joined using a intensive set of control, inspection constructs, including series, conditional majors, parallel, and loop sections. (See Figure 6 on page 13)

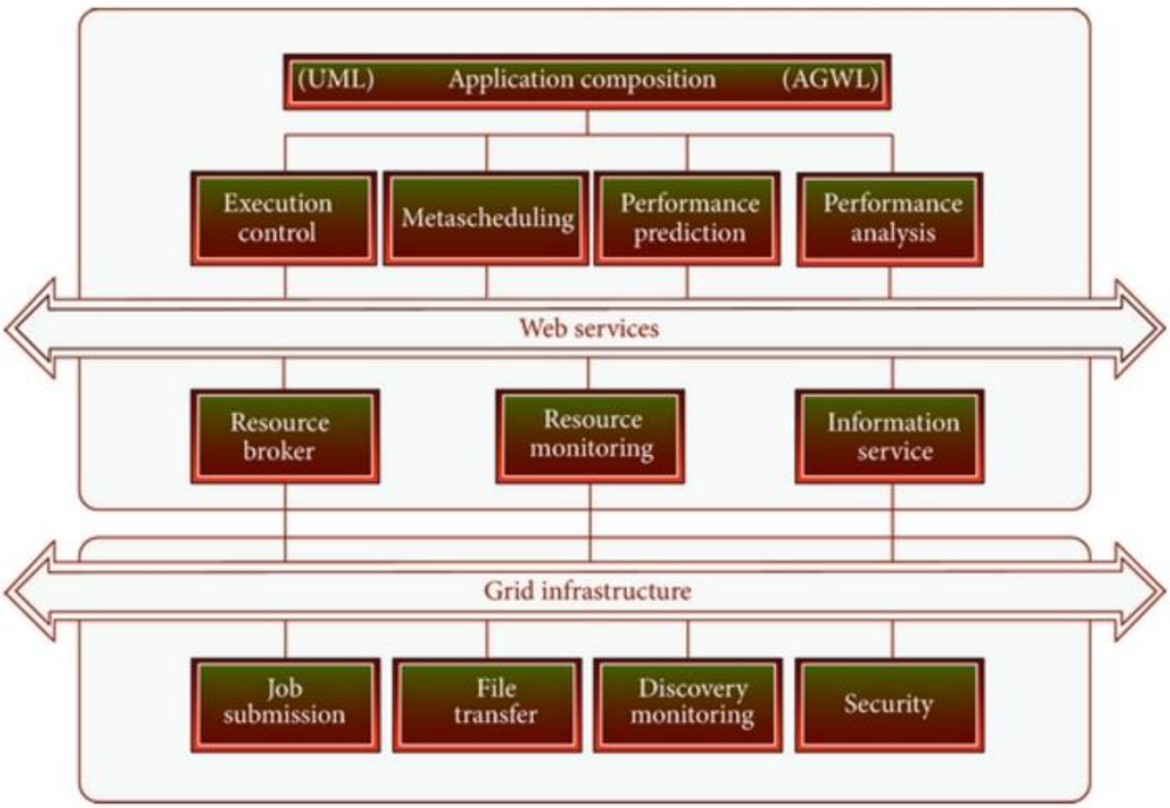


Figure 6: Structure of the Askalon system.

2.2 Distributed data resources

The latest development technologies collecting data with high capacity in a number of areas (e.g., space sciences, atmospheric sciences, biological sciences, commerce) together with advances in digital storage devices, computing and communications technologies have led to the spread of a plurality of physically distributed data stores created and supported autonomous entities (e.g. organizations, scientists). The resulting incrementally information rich fields proposal unprecedented possibilities in computer assisted data-driven experience obtaining in a many of applications inclusive in particular, discovery of data-driven scientific (e.g., characteristic sequence of the protein-structure-function relationships in the field of computational molecular biology), making of data-driven decision in commerce and business, controlling and monitoring of sophisticated systems (e.g., in area of electric power networks load forecasting), and security issues of informatics (detection of and countermeasures contrary attacks, assaults on communication infrastructures and critical information). Machine learning, proposals best way like the most cost-effective approaches for, extracting ,exploring and analyzing knowledge (characteristics, correlations, and another sophisticated connections and hypotheses that demonstrate potentially attractive regularities) from data. However, the feasibility of current, nowadays approaches to machine learning in emerging information, full of data applications is specially limited by a number of elements, factors:

a. Size of data storages are large, physically, and dynamic distributed. As a result, it is neither desirable nor possible for analyzing to collect all information and data in a centralized location. Therefore, there is a new approach for efficient, possible algorithms for analyzing and discovering manifold distributed data sources out of transmitting large numbers of data.

b. Autonomously improved and operated sources of data often be contrary in their construction and organization (e.g. flat files, relational databases etc.) and the operations that may be applied on the data sources (e.g., types of queries - statistical queries, keyword matches and relational queries). Hereby, there is a necessity for theoretically well-grounded tactics for efficiently getting the information needed for monitoring, controlling and analyzing within the operational constraints embarked by the data sources. The objective of this introduction is to clearly define learning classifiers problem from distributed data and sum up nowadays advances that have led to a help to this problem. [27]

2.2.1 Learning from Data Task by Decomposition

A common strategy for modelling, designing algorithms for analyzing from distributed data that are demonstrably exact with regard to their centralized equivalent (like defined above) follows from the investigation in which most of the learning algorisms utilize only specific statistics calculated from the data D in the process of creating the hypotheses that they throughput. (Recall that a statistic is just a function of the data. As an examples of statistics enclose mean value of an counts of instances, attribute whereas have defined values for many subset of the most permanent value of an attribute, attributes etc.). This environments a native separation, decomposition of a learning algorithm into two pieces (see Figure 7 page on 16):

a. An information extraction component that formulates and sends a statistical query to a data source and

b. The resulting statistic is used by a hypothesis generation component for modifying a partly established hypothesis (as needed information extraction component can be invoked for further). [27]

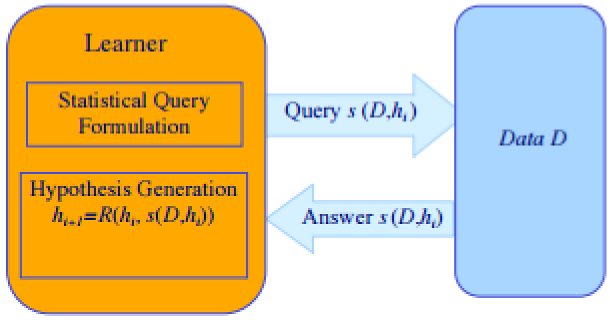


Figure 7: A native extrication of a learning algorithm

2.2.2 Distributed Data Fragmentation

Definition of Fragmentation: The system separations the relation into several, alternative stores and fragments at a different site. If a distributed database A is divided this means that the parties $P1, P2, \ldots, Pj$, $i \in I, I = 1, 2, \ldots, n$ which create her form disjoint subsets:$A = \{P1 \cup P2 \cup \ldots \cup Pj\}$, with $\{P1 \cap \ldots \cap Pj\} = \emptyset$. The fragmentation means that, the separating of a global relation R into fragments $R1, R2, \ldots, Ri$, in which existing enough data, information for reestablishing the original relation of R. Within the fragmentation process three there are 3 main rules that have to be considered, which guarantee that the database does not have any changes like semantic during process fragmentation, for example ensure succession of the database:

• **Reconstruction.** For reconstructing R from fragmentation it must be feasibly to detect relational operation. Union operation is accounted reconstruction for horizontal fragmentation whereas for vertical it is join operation.

• **Completeness.** Completeness in fragmentation means that if relation R is separated like $R1, R2, \ldots, Rn$, fragments then every data item in which may be found in relation R have to available at leastways one fragment.

• **Disjointness.** It means that data item cannot be appear in any other fragment if it is available in Ri fragment.

• **Exception.** For permitting reconstruction primary key attributes have to repeat in vertical fragmentation process. İn contrast to the vertical fragmentation each item of data is a tuple in horizontal fragmaentation whereas data item is an attribute in vertical. [28]

2.2.3 Fragmentation methods

Take into consideration a relation with R scheme. Applying of algebraic operation over R (such as operation with relations for demonstrating logical characteristics of data) and obtaining many fragments (subschema) Ri in which this process called fragmentation process R.

a) **Horizontal.** The horizontal fragmentation is if relation R is separated like $R1, R2, \ldots, Rn$, fragments then every data item in which may be found in relation R have to available at leastways one fragment. (Completeness condition). Except this, also an extra disjointness condition, supposing that every tuple of relation R be plotted into completely one tuple of one of the fragments, this way is frequently presented in distributed database systems in turn to check the existence of duplication clearly at the fragment level (according to having plural copies, duplications of the equal fragment). The followed fragments R_i have the same layout structure as well as collection R, but contrary by the data they include and are resulted by implementing a selection on relation R. Selection $o_p(R)$ - determines a connection in which implicates only those tuples of R that fulfill the condition in which remarked (predicate p):

$$o_p(\prod a_1, \ldots, a_n(R)) .$$

A horizontal fragment can be get by implementing a restriction:$R_i =$ ð$_{condi}(R)$. So using union operation I can reconstruct the original relation such as below:

$$R = R_1 \cup R_2 \cup \ldots \cup R_k.$$

For instance:

$R_1 = \theta\, type = \text{'}Computer\text{'}(PropertyForSale)$

$R_2 = \theta\, type = \text{'}Notebook\text{'}(PropertyForSale)$

Horizontal partitioning is divided into two section:

• **Primary horizontal fragmentation** means that a relation is gained through utilize of predicates determined on this relation in which limits the tuples of the relation.

• **Derived horizontal fragmentation** is understand by usage predicates that are described on other relations.

b) Vertical. In vertical fragmentation it partition the relation such as vertically by columns. The gained fragments R_i keep just only piece from the collection of structure R. It contains only definite attributes at definite site and they catch relation R 's primary key to provide that the recovery is feasible and are resulted out of relational algebra in which applied the application of a projection operation:

$$(\prod a_1, \ldots, a_n(R)) .$$

whereas $a_1, \ldots, a_n$ are attributes of the relation R. If every attribute of the relation is plotted into at least one attribute of the fragments then this shows that fragmentation is correct; in addition, it have to be possible to restore the original, unique relation by connecting the fragments together like:

$$R = R_1 \otimes R_2 \otimes \dots R_n \otimes$$

In other way, joining of relation after decomposition of relation process the fragments have to be a lossless.

For instance:

$$R1 = \prod staffNo, position, salary(Staff)$$

$$R2 = \prod staffNo, firstName, lastName, branchNo(Staff)$$ [28]

c) From time to time, merely horizontal or vertical fragmentation of a database layout is underproductive to spread adequately, equally data for some applications. To perform mixed or hybrid fragmentation for some application can be more useful than separate. This method (mixed fragmentation) from a relation consists of a vertical fragment that is horizontally fragmented, or a horizontal fragment that is vertically fragmented. Operations are selection and projection in which are used in mixed fragmentation:

$$\sigma_p \prod a_1, \dots, a_n(R) \ or \ \prod a_1, \dots, a_n \left(\sigma_p(R)\right)$$

Below advantages of fragmentation process are listed:

- *Usage* – this means that applications implement with views rather than whole relations;

- *Efficiency* - the meaning of efficiency considered that data stored close the place where is it frequently used;
- *Parallelism* - with fragments are the unit of distribution, a transaction can be separated into a few sub queries in which run on fragments;
- *Security* - local applications don’t restored data in which not required and unauthorized users can not have access to sign in application;
- *Performance* of universal applications that ask data from a few fragments established at various sites many be slower. [28]

2.3 Granularity and Performance

To obtain good parallel performance of application one way is that the best (right) granularity of application must be chose. In parallel task domains the meaning of granularity is amount of real work. There are two types of granularity: Fine and coarse.

Too coarse granularity or too fine granularity are not best approach for parallel task whereas application performance will endure from communication overhead if it is too fine. If not, granilarity iss too coarse it means that load imbalance will be problem for perfomance of application. That’s why for good performance the aim is to define the most appoprate size of granilarity (in most case large version is better) for parallel task. If system can achive escaping communication overhead and load imbalance this means that best performance of application obtained.

From here is it clear that, right size of work in greatly affects its parallel performance in a single parallel task (granularity) of a multithreaded application. One approximation is logically division of the problem into as many possible as parallel tasks during the decomposing operation an application because of multithreading. During the parallel multithreading the main work is making strong communication between shared data and sequence of execution. Since distribution tasks, defining the jobs to threads, and creating communication (sharing) data, information among tasks are not simply operations, many of them needs to combine partitions or agglomerate, to accomplish these overheads and reach the most productive, effective implementation. The agglomeration step is the process determining the best is process of agglomeration step for parallel tasks. The main goal of granularity is to balance the workload amid threads. When it is easier to stabilize the workload of a great number of smaller tasks, this action may cause excessively parallel overhead like some form such as communication, synchronization, and so on. For this reason, increasing granularity task (amount of work) is one way for reducing parallel overhead inside of each task by assembling smaller tasks such as a single task. Nowadays there is some tools like Intel® Parallel Amplifier in which it may help defining best granularity for an application. [29]

2.4 Task Clustering

In Data Mining clustering and classification are both fundamental, essential tasks. For analyzing data classification is utilized mostly such as a supervised learning approach. The main aspect of clustering is representational, descriptive, whereas main goal of classification is predictive. Since the purpose of clustering is to discover, to investigate a new set of categories, the novel groups are interest in themselves, and their evaluation is actual, reliable. Within task classification, nevertheless, a substantial section of the assessment, evaluation is extrinsic, onwards the groups have to mirror some reference set of classes. “Understanding our world requires conceptualizing the similarities and differences between the entities that compose it” [30]. Clustering data instances groups into subsets so that such cases similar instances are grouped together, whereas various instances concern to different groups. The examples are thereby orderly inside an efficient presentment that it characterizes the population in which being sampled. Officially, the structure of clustering is demonstrated like a set of subsets

$C = C_1, \ldots, C_k$ of S,

$L = L_k \; i = 1 \; C_i \; and \; C_i \cap C_j = \emptyset$ for $i \neq j$.

As a result, each instance in L belongs to completely one and just only one subset. The meaning of clustering of objects is as old as the in which form the ancient time people need for depicting the prominent properties of men and objects and defining them with a type. That's why, it hugging different scientific disciplines: from statistics and biology to mathematics and genetics, each of which uses different terms to characterize the topologies shaped using this analysis. From genetic “genotypes”, to manufacturing “group technology”, biological “taxonomies”, to medical “syndromes” and to biological “taxonomies”— the problem is same all listed scientific disciplines: building categories of entities and appointing individuals to the suitable groups within clustering.

As we know basic characterize of clustering is collecting similar objects or instance together and there is a sort of measure in which help to define whether two instances are same or different. Generally, to determine of this relation there are two basic way for measuring.

- distance measures according to distance
- distance measures according to similarity

In most case, it is preferably to use distance measure in many clustering methods for defining similarity and dissimilarity among any pair of instances. It is helpful to represent the distance between two objects zi and zj like $d(zi, zj)$. A reasonable distance measure have to be symmetrical and gets its minimum value (in most cases zero) in case of equal vectors. If following formulations are satisfied then this type of distance measure is named a metric distance measure:

1. $d(z_i, z_k) \leq d(z_i, z_j) + d(z_j, z_k) \forall z_i, z_j, z_k \in S$ This equations is called triangle inequality

2. $d(z_i, z_j) = 0 \Rightarrow z_i = z_j \forall zi, z_j \in S$. [31]

Compartmentalization of clustering algorithms is neither elementary, nor standard. Overlapping of groups in reality is described below. For reader's simplicity I showed with diagrams a classification of clustering algorithms. Appropriate terms are expressed following.

2.4.1 Clustering Algorithms

Following diagram demonstrate clustering algorithms: (see Figure 8 page on 24)

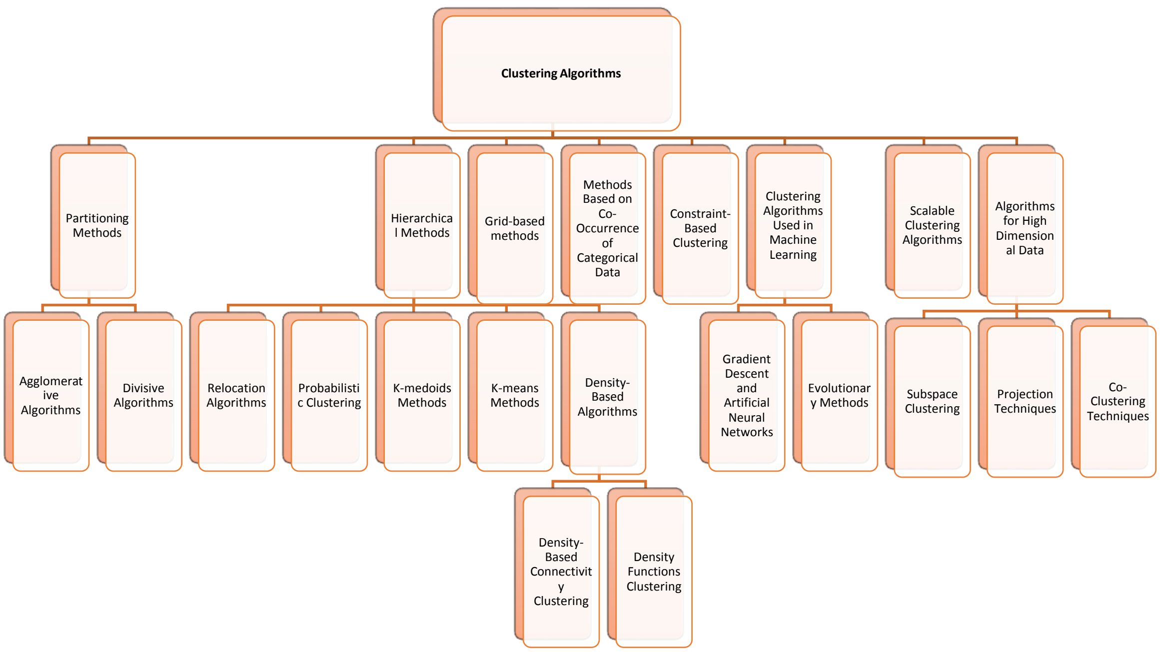


Figure 8: Diagram for clustering algorithms

2.4.2 Hierarchical clustering

Hierarchical clustering covers generating clusters in which have a predetermined, prearranged ordering from top to bottom respectively. For instance, we can say simple example from hard disk in there all files and folders are organized orderly in a hierarchy. Generally, hierarchical clustering methods divided into two types.

- Divisive
- Agglomerative. [32]

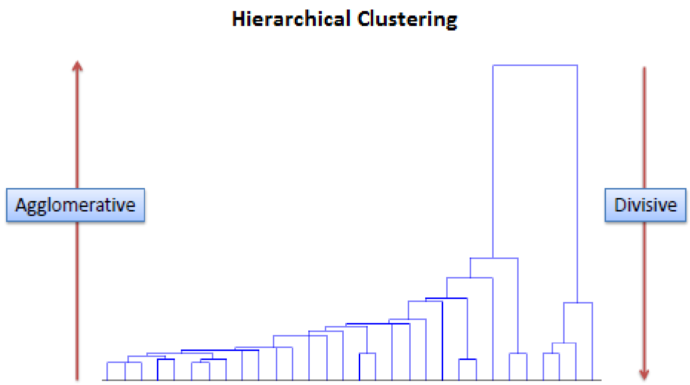


Figure 9: Types of Hierarchical clustering: Agglomerative and Divisive

I define every observation to its own cluster in this method. Then, calculate the similarity (e.g., using different measure such as distance or similarity in which described above) amid each of the clusters and participate the two most matched clusters. Eventually, repeating of steps 2 and 3 when there is only a unique cluster left. The concerned algorithm is shown following. (See Figure 10 page on 26)

Given:

A set X of objects $\{x_1,\ldots,x_n\}$

A distance function $dist(c_1,c_2)$

for $i = 1$ to n

  $c_i = \{x_i\}$

end for

$C = \{c_1,\ldots,c_n\}$

$l = n+1$

while C.size > 1 **do**

- (c_{min1},c_{min2}) = minimum $dist(c_i,c_j)$ for all c_i,c_j in C
- remove c_{min1} and c_{min2} from C
- add $\{c_{min1},c_{min2}\}$ to C
- $l = l + 1$

end while

Figure 10: Algorithm

Determining of the closeness matrix having the distance between every point using a distance function is required before any clustering method that is performed. After that, for indicating distance between every cluster the matrix have to be updated. The below three methods showed calculation of distance between each cluster.

- **Single Linkage**

Additionally accepted such as nearest neighbor clustering, this method is one of the earliest and most well-known of the hierarchical techniques. The method approached is define like: distance amid two groups is determined such as space between two nearest members. It often efficient clusters whereas each individuals are joined consecutively to a single group. The distance equation coefficients are defined as following:

$$\omega_i = \omega_j = 0.5\,;\ \varphi = 0;\ \ \theta = -\,0.5$$

- **Complete Linkage**

Complete Linkage well-known as finding of furthest neighbor or maximum method, this complete linkage method determines the distance amidst two groups such as the space between their two farthest-apart members. Being clusters well separated and compact is more efficient for this method. In this method coefficients in which described as following for calculation of distance between members of two groups.

$$\omega_i = \omega_j = 0.5 \; ; \; \varphi = 0; \;\; \theta = 0.5$$

- **Simple Average**

This algorithm determines the distance between groups according to average distance between every members of clusters whereas method is also named the weighted pair-group. It is clear that two group's member weighted so that each of them have an equal influence on the last result. Equation for calculating the coefficients of the distance is defined like:

$$\omega_i = \omega_j = 0.5 \; ; \; \varphi = 0; \;\; \theta = 0$$

- **Centroid**

In addition to above methods there is also one method in which referred to like the unweight pair-group centroid method, this technique specifies distance according to two group members centroids (distance according to centroids such as vector average or center of gravity). The technique have to only be used with Euclidean distances. Coefficients of methods calculated with following equation are:

$$\omega_i = \frac{n_i}{n_k}; \; \omega_j = \frac{n_j}{n_k}; \; \varphi = \omega_i \omega_j; \;\; \theta = 0$$

Reverse links can consist with this method. These are recognizable when the dendrogram no longer demonstrates all these are recognizable that its simple tree-like construct in which each mixing results in a novel cluster that is at a higher distance level (action from right to left). Sometimes with backward links, fusions may occur that final outcome in clusters at a lower distance level (move from left to right). The dendrogram is inconvenient to commentate in this case.

- **Median**

Also named the weighted pair-group centroid method, this methods defines the space between two groups as the weighted distance between their centroids, weight proportional to the number of individuals in every group. If we return backwards (have a look debate under Centroid) can occur with this method. The method have to only be used by Euclidean distances. Calculation of the coefficients are defined like:

$$\omega_i = \omega_j = 0.5 \; ; \; \varphi = -02.50; \;\; \theta = 0$$

[33]

2.4.3 Methods for Partitioning

The combinatorial optimization algorithms are perhaps the most famous class of clustering algorithms in which also known as iterative relocation algorithms. The basic aspect of this algorithms minimize, reduce a given clustering criterion according to iteratively shifting data points amid clusters since a (locally) optimal partition is achieved. I can say basic iterative algorithms such as K-means or K-medoids in which belong to partitioning methods. In this method joining is locally and globally cannot be optimal solution for some problems. Because of number of data points in any given data sequence is always limited and also distinct partitions number is finite, using exhaustive search methods it could be possible to prevent from the local minima problem. However, this ideas just is truth only in theory, not practically while globally optimal partition method founded. This method is known to be NP-hard problem and such types of methods are not efficient in practice. Second type of Stirling number means that number of different partitions in K groups for n investigation, observation defines as following:

$$S_n^{(k)} = \frac{1}{K1}\sum_{i=0}^{i=K}(-1)^{K-i}\binom{K}{i}i^n$$

This demonstrates that it is impossible for even nearly small problems enumeration of all possible partitions enumeration. While additionally the number of clusters is unknown on that time problem is even more troublesome. Total of the Stirling numbers of the second kind's number of various combinations is determined as following:

$$\sum_{i=1}^{i=K_{max}} S_n^{(i)}$$

where $as\ K_{max}$is the maximum number of cluster and it is clear that $Kmax \leq n$. All this formulations show that even with modern computing exhaustive search method is very time consuming. Furthermore, it appears be an infinite race amidst computer power and amount of data, in which two of them was increased constantly within the last decades. Therefore, more convenient approach for practice works are the iterative optimization than exhaustive search.

2.4.4 Density based methods

- Many partitioning methods cluster instances based on the measuring of distance between two objects.

- Density based methods may find merely spherical-shaped clusters and meeting complexity, hardship according to discovering clusters of arbitrary density.
- Density-based spatial clustering of applications with noise (DBSCAN) and Ordering points to identify the clustering structure (OPTICS) are two types of density based methods
- The idea in this types of methods is to proceed increasing the given cluster as long as the density (data points or number of instance (objects)) in the "neighborhood" overruns some threshold.

2.4.5 Grid based methods

- The idea of grid-based methods is to quantize the instance distance into a limited number of cells in which created such as a gird structure.
- In this method grid architecture is used for all the clustering operations in grid based methods.
- Main best side of this methods is processing time is very fastest.
- I can say some grid based algorithms such as Statistical Information Grid-based method (STING), Wave-Cluster, and CLIQUE.

2.5 What is Cloud Computing?

Cloud computing is best described as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction" [34] in National Institute of Standards and Technology (NIST). Cloud computing have of three various types of service provision. In every case the services are received remotely and accessed over a network (specially the internet) across a web browser of customer, rather than being installed locally on a user's computer. Initially, software as a service (SaaS) applies to the provision of software applications in the cloud computing environments. Secondly, platform as a service (PaaS) cnsults to the provision of set of services in which available users to distribute, in the cloud computing, applications generated help of programming languages and equipment whereas supported by the supplier. Finally, infrastructure as a service (IaaS) preferable to services supplying computer processing power, keep space and network capability, in which available customers to execute arbitrary software (such as applications and operating systems) in the cloud computing. These three elements in cloud computing are together counted such as stack. The providing of IT services in the cloud has been made possible two of them by the evolution of advanced data centers and widespread authority to developed bandwidth. These technical approach advances mean that services may be hosted on machines over a broad range of locations however, from the client's aspect, they exclusively generate in the 'cloud'. The cloud design available clients to get access, from any computer in which this computer connected to the internet (it doesn't matter via a desktop PC, a mobile device or others), a plurality of IT services rather than being finite to utilizing locally installed software programs and being dependent from the storage capacity of client local computer network. In recent years, IT service provision model one of the best model in which it is growing exponentially. This model is considered that one third of all income created in the software market today because of delivery of cloud computing services. That the value of the UK cloud computing market could achieve approximately £10.5 billion until 2014 in which it was upped from £6 billion from 2010.

2.5.1 Creation of Cloud

The cloud atmosphere is subdivided into hybrid, private, hybrid and public community clouds.

• Public clouds means that services are valid to the public, all users at large over the internet.

• A private cloud as you clear from the name it is fundamentally a private network used by one client for whom data privacy and security is mostly the primary attention. The disadvantage of this type of cloud is that the client will should bear the important cost of setting up and the network maintenance.

• Hybrid cloud surroundings are mostly used where a client has requirements for a fusion of dedicated cloud hosting and servers, for instance if some of the data that is being kept all that data have of a very sensitive nature. In such cases the system can select to store some data on own dedicated server and less sensible data in the cloud environments. There is another general reason for utilizing hybrid clouds is where a set-up necessity more processing power than is existing in-house and catches the additional requirement in the cloud. Cloud bursting term is more appropriate this type of cloud creation. Furthermore, hybrid cloud jargons are often established in situations whereas a client is make motion from a completely private to a completely public cloud setup.

• Community types of clouds creation usually obtain where a finite number of customers with same IT requirements split a substructure supplied by a single supplier. The revenue of the services are shared between the clients so this model is better, because of an economic view, than an only renter arrangement. Even though the cost savings are probably to be greater in a public cloud branch, community cloud customers commonly benefit from greater privacy and security, in which it can be major for policy reasons.

The main advantages and disadvantages of cloud computing are described in following.

2.5.2 Advantages of Cloud Computing

Access to resources

The biggest benefit of cloud computing is the access whereas it provides to the processing power of various remote computers or other devices. This available clients to get advantage of greater calculation speed and larger storage capacity than most organizations may supply on their antecedents and at a fraction of the money.

Mobility

Clients may login the services from almost every location in the world because the cloud services are web-based (in addition, the advent of mobile devices). This can make possible employees to login significant business tools while they are on the motion. For instance, the personnel may charge in a Tempora online timesheet while on a train, supplying the residue of the business with login to that data in real time.

Easily scalable

The monthly subscription and pay as you use-both type of this approach charging models do it simple for the amount of service being supplied to be increased or reduced. It is clear that, sometimes customer wish to multiply the number of seats involved in its subscription to Tempora whether the amount of data such as megabytes of storage space rented from Amazon Web Services (AWS), this can be done simply. The supplier easily ensure access to extra users or make more the storage space accessible in exchange for higher monthly payments by the clients. The scalability of the cloud computing environments design does it especially attractive to increasing organizations with differing levels of requirement for computer resources (e.g. whereas an agency's sites receives higher volumes of guests at certain times of year).

Security of data and capacity of storage

Data security is of special significance as intervals in procedure may cause severe reputational and financial damage. For the large number of organizations, the data centers offer security and data storage capacity is far excellent to that which can be carried in-house. This is because organizations specialize in the secure storage of data.

Cost savings

In many business-orientated cloud computing services are use one approach such as paid for and the payment model is many times a rental layout based on monthly subscription charges like "per user" or "pay as you use" system. This indicates that there is no broad upfront pay as there would be with the purchase of a license in the reasonable software license design. In addition, there may be a primary setup or format fee, it is mostly very low by collation. According to monthly subscription charges systems will also add maintenance and support fees, in which may be significantly sophisticated in the orthodox software license design application. Except this possibility, this type of models client do not need to pay any money for secure servers because of this service is supplied by third-party data centers whereas all this services are included in the subscription charge. In "pay as you use" system models are many beneficial for organizations with troughs and peaks because of demand for computer resources. This approach cheaper and useful than paying for special use of enough data resources in which to meet peak demand when it is not necessary, as is the case where all calculation is completed out by an organization in-house. Furthermore, services in cloud decrease the need for an organization to preserve in-house specialty in their own technological structure, which reduces cost for IT services. Eventually, cloud computing services do not demonstrate a capital consumption, so clients mislay less if user switch suppliers.

Maintenance and support

Most of time the supplier will present continuous support services. Nevertheless, maintaining and supporting the services less unexpected for clients because of remote hosting of the services. Without visiting a client's site backups, upgrades and updates may handle automatically by supplier. All this will commonly intend that support and maintenance may be carried out more rapidly. Moreover, clients are able to pick-a-back on their suppliers' upgrades in resources.

Environmentally friendly

It has been recommended that data resources centers are a green another to in-house computing and this is the most discussed topic. This is because of very big data centers services typically execute at approximately 80 % capability, whereas an in-house server may execute at 5% capacity, to let for peaks in resource request. In addition, a server operating at 5% capacity uses only lightly less power per hour than 1 executing at 80 %, in which this indicates 16 times less computation by comparison. At the same time, it is likely that the presence of inexpensive and more easily current cloud computing structures has increased the overall demand for calculation, passing the energy-efficiency earns that have been made in data centers. One way for choosing supplier that utilizes a data center in which makes use of wind cooling or solar technology, moreover, data center is based on area whereas renewable energy resource organize local electricity.

Free trials

Some seller proposal the chance to trial their yield for a term without charge. This way is easier by the seller's capability to destroy access at the end of the term and supplies them with the chance to "catch" the client. This business version is occasionally referred to as a 'freemium'.

2.5.3 Disadvantages of Cloud Computing

Internet reliability

It is clearly that where IT services are supplied through the internet, lack of internet availability or leisurely connections will inhibit access to those services. But sometimes this problem can be main business-critical issue. However, as internet access develops, this should be an increasing issue. In addition, there is no guarantee that service will not interrupt when system was rendered by bugs or defects.

Dependence on the supplier

As you know for user access to the IT services more important rather than supporting or maintaining of systems. If the user of IT services is in financial problem, is trusting on an unbalanced subcontractor or is concerned in lawsuit, its capability to ensure the services can be affected. These problems could leave the clients to business-critical systems without access. However, dependence on a supplier is a general idea for many organizations and the regular risk assessment can be implemented to reduce that risk. Due effort monitoring on the supplier can reveal whether it is, for instance, in fiscal trouble and references may be search from available or past clients to build whether the vendor has a history of authenticity. The client may any time look for to include current measures in the contract to ensure preservation from the risks mentioned. Finally, if in too much suspicion, the client may need to select another supplier. As piece of supplier election, the customer have to take into account what steps will be essential to switch sellers if this verifies necessary. For instance, what cancellation notice terms implement, how the purchaser's data will be took back from the supplier-audited servers (format type also including) and which level of immigration support is available from the dealer. In addition to, it is cautious to set up what level of deduction to process would be provoked by switching suppliers; in other words, determining how long it would take to get up and executing with an another supplier. IT services are provided in some cloud computing sellers such as the orthodox license layout. [34]

2.6 What is Grid Computing?

In today's widespread world of demanding information anywhere and anytime, the detonating Grid Computing atmosphere have now tested to be so prominent that they are often applied to as being the most powerful and single computer solutions. It has been achieved that with the many advantages of Grid Computing, we have as a result presented both a complex and complicated worldwide environment, which pressures a vastness of open technologies and standards in a broad different of application schemes. Actually the complication and dynamic structure of industrial problems are much more excessive to fulfill by the more single and traditional computational platform solutions in today's world.

In excess of the past several years, there has been a lot of curiosity in computational Grid Computing world. We also remark a number of reproduction of Grid Computing, including science grids, knowledge grids, compute grids, access grids, data grids, commodity grids and terra grids.

As I investigate careful inspection of these grids, we may notice that all types of grids share some form of resources. In addition, these grids may have various structures. The main aspect of a grid, whether it is a commodity profitableness grid or a computational grid, is in most time estimated based on its business qualities and the respective customer satisfaction. User satisfaction is measured according to the Quality of Service insured by the grid, such as the simplicity of access, performance, availability, business values, flexibility, and management aspects in estimating. Quality of business mostly relate to and demonstrate the problem being dissolve by the grid. For example, it may be job running, administrative aspects, simulation of workflows and other main technology-based basis.

The data features of most Grid Computing environment have to be able to efficiently manage whole aspects of data, including data transfer, data location, critical components of security and data access. The pith functional data requirements of Grid computing was described in following.

2.6.1 Characteristic of Grid Computing Applications

- ✓ The skill to integrate multiple independently, managed, and heterogeneous, distributed data sources.
- ✓ The ability to ensure productive data transfer gears and to supply data where the calculation will take place for better efficiency and scalability.
- ✓ The aptitude to provide both replication mechanisms and data caching for minimizing traffic over network.
- ✓ The knack to provide important data finding mechanisms, which let the user to discover data based on features of the data.
- ✓ The ability to accomplish integrity checks and encryption of data to provide that data is moved across the network in a safe style.
- ✓ The aptitude to ensure the recovery or back up possibilities and policies important to protect losing of data and decrease unplanned downtime over the grid.

2.6.2 Computation

The pith functional computational requirements are demonstrated in following for grid applications:

- ✓ The aptitude to permit for independent administration of computing resources
- ✓ The capability to insure mechanisms that can transparently and intelligently elect computing resources able of executing a customer's job
- ✓ The comprehension of the present and predicted dynamic resource configuration, resource availability, loads on grid resources, and provisioning
- ✓ Failover mechanisms and error detection
- ✓ Provide suitable security gears for safe resource management, access, and integrity

As characterized earlier, initially, the concentrated Grid Computing actions were in the environments of computing performance, storage resources, and data access. The determination of Grid Computing resource distributing has since changed, based upon experiments, with more concentrate now being applied to an experienced form of coordinated resource distributing distributed within the participants in an actual organization. This application idea of coordinated resource sharing comprises any resources available throughout a real organization, including computing power, applications, software, hardware, data, networking services and any other forms of computing resource achievement. [35]

2.7 Failures

2.7.1 Hardware Problems

Telecommunications tool providers have concentrated on category of hardware problems as the essential predictor of network failure rates. About 25% of all failures appear as a result of hardware problems like computer failures. To upgrade general hardware validity in telecommunications products, suppliers set up redundancy into their product. A network designer may choose and use equipment with a broad range of redundancy functions ranging from having no redundancy to the whole links and duplication of equipment. When employed in this limited context, predictors use Bellcore and US Military in which they are more useful for them.

There are approaches in more than just the hardware components that are chose in the real deployment of the networks. All this variations comprise equipment's quality, network design and planning's quality, implementation's complexity, the intercommunication and interoperability of elements. A lot of networks are exceptionally complicate systems - it is wonderful that they indicate any constancy at all.

Legation critical networks are modelled to have five-nines existence and are required to encounter that performance criteria based on an MTBF valuation. Except this, there are 4 other significant types of failure determined by Kyas that contribute to the other network failure's 75% not identified within hardware problems analysis by an MTBF. These other collaborators to network failure percentage have to be considered to exactly assess and predict network accessibility. However, MTBF approach is not a corresponding measure for 3 of these categories.

2.7.2 Operator Error

Operator Error (OE) as determined as those failures evoked straight by human activities by Kyas. OE is farther subdivided into deliberate or unintentional errors and as mistakes in which do or do not cause resulted damage. Kyas impresses that OE is responsible for more than 5% of all system errors. This figure usually differs from establishment to establishment according to the level of training and other parameters such as general culture and methods.

This sort of error is profitable in examining practicable types of network system failures. An operator error that influences the network authenticity may be created from people's mutual relation with networking equipment, connectors and cables as well as from occasions by other IT instruments result from user activities. In addition, IT devices like e-mail servers and database servers can make broadcast thunderstorms and network addresses' duplication because of activities of individuals operating the different devices over the network.

2.7.3 Mass Storage Devices

This class is determined as the failures connected with mass storage devices. Failures of these apparatuses have been learned by different producers as well as by these devices consumers. Nevertheless superior performance hard drives may obtain unusually high MTBF values of 10^6 hours in which this means 114 years. Moreover, bank employers have many such type of failures because of the large quantity of drives in use.

In addition, environmental aspects like: mishandling, physical handling or temperature variation with the frequency of current drive operations as nonstop seek operations will influence MTBF and its statistical distribution. The failure analysis may take into consideration these factors into validity planning of a network.

In spite of the failure of such devices is not by itself meditated to be a network failure, there is a fast growth in the deployment of SAN (Storage Area Networks) whereas great sequences of mass storage devices are frankly joined to a network over high capacity channels. Storage Area Networks do actually group as network devices because they are network-centric. Traditional MTBF evaluations are corresponding for these devices from a computer hardware perspective.

2.7.4 Software Problems

At present, enterprise networks join many numbers of servers that ensure functionality many users utilizing a very huge number of software applications. Broadly distributed systems are general in enterprises that are geographically destroyed. The network insures all connection amid different computer platforms and users. In systems of such complication, even with careful planning, checking and evaluation, it is hard to describe the service requirements on the network. Failures and errors may grow from unreasonable delays, insufficient capacity within peak requirements well as a disastrous failures increasing from the loss of an important element or resource.

Network software failures may be evoked by defective device drivers, intelligent distinctions in handling and protocol implementation, OS faults and other anomalies. According to Kyas investigating results software problems be responsible roughly the same number of failures is 25% like hardware problems. This problems are vital to any meaningful validity analyses.

2.7.5 Network Problems

Software and Hardware tasks that are directly connected to the Network are included in this class. These can explain over $\frac{1}{3}$ of IT failures. To preferable understanding of the distribution and character of these sort of failures it is profitable to debate them in the OSI model's context. Figure 11 indicates the separation of failures among the layers of OSI in Local Area Networks.

Reasons of failures through the lower layers of the model are often faulty Network Interface Card (NIC), defective relationship and cables, failures in interface cards between routers and switches, flashlight failure like Token Ring networks, packet size errors and checksum errors. Likewise Ethernet technologies have flourished within time passing, there is a decay in the failure percent during the lower layers of the OSI design. However there has been an improvement in the failure percent in the Application Layer such as software complication continues to detonate.

Many type of failures and errors as indicated here are often localized and not calamitous in nature. Localized failures are extremely various from that determined by the Bellcore and the US Military scheme which permit for local failures to happen and not be took into consideration a device failure. The contribution of localized failure perspective to network validity, it is substantial to pay attention the size of failures and scale in which are evoked by individual network elements. For instance, NIC failures will not reasonable result in a Single Point of Failure of the establishment network.

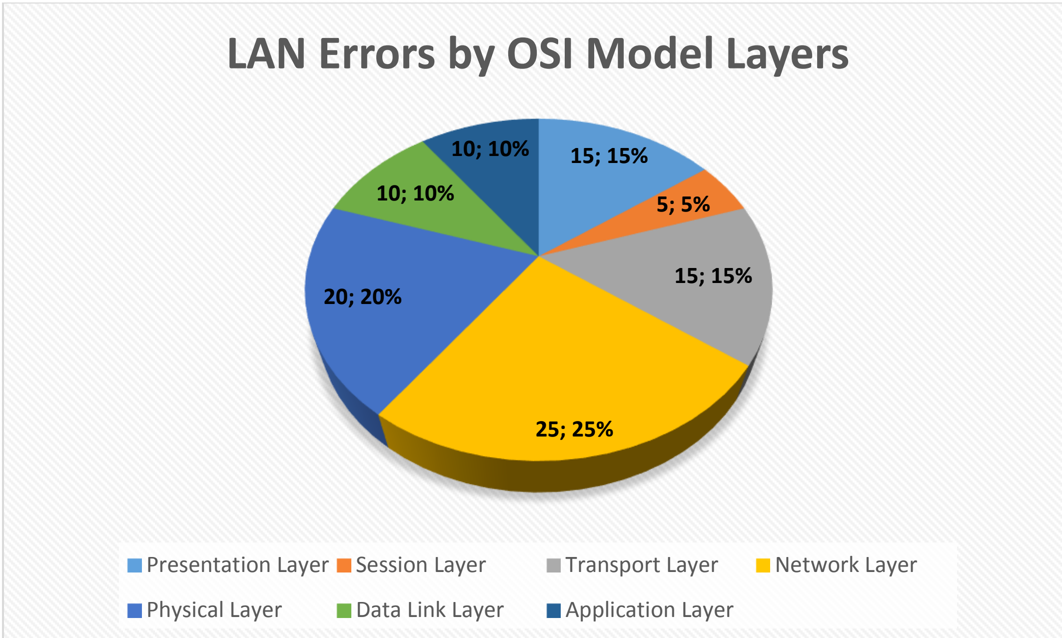


Figure 11: Frequency of LAN errors by OSI model Layers

Another failures in which not categorized by Kyas

Though Kyas's 5 categories be defendant a huge number of network failures, the below 2 extra categories merit discussion and debate:

- Failures such as Trojan Horses, Worms, Viruses and Malicious software due to Denial of Service Attacks
- Errors from catastrophe like earthquakes, outages, fire and flood, earthquakes.

2.7.6 Denial of Service Attack

Denial of Service attacks are a main cause of network failures from 2000. Nowadays all this attacks are happening several periods a year resulting in service disintegration universal. The density of this failure is improving at an alarming step. Only special closely checked networks without Internet transition are immune from this shape of assault by deploying such as air-gaps in the network. Air-gaps are a physical gap without connection and in which data is manually moved amidst nodes. This technique is not efficient for the wide majority of networks at present that based on Internet connectivity.

A sample of the influence of Denial of Service attacks we can say the Code Red virus whereas this types of viruses are a more recent version. In addition, Slammer worm disrupt many of computers by unleashing a well-established Distributed Denial of Service Attack. All this types of attacks resulted in a substantial loss of total moneys worldwide. The raised frequency of events or threat and effect of this sort of network errors on network collapsing are significant and that's why the Denial of Service Attack type have to be comprised in any reliable failure analysis schema of an Internet joined enterprise network.

Describing the range of network failures aroused by this kind of error is hard because it is like a current phenomenon with fortuitous occurrence. But, the potential effect of this errors is widespread and immeasurable whereas cannot be discounted.

2.7.7 Disaster Scenarios

The last kind of failure accounted is that of disaster scenarios which appear from a broad range of cases, most of them environmental and some artificial. Floods, hurricanes, long-term power outages, earthquakes, fires and tornadoes are included into environmental disasters. War, vandalism, arson, theft and acts of terrorism can be considered synthesized disaster. In every of these catastrophe scenarios may more reasons can be listed. In some circumstances there is local occurrence that can be useful in forecasting the future of such event. Nevertheless, in most other moments no past knowledge or helpful methods of prediction is feasible. Disaster projecting has only lately become a super IT first place as the collective mindset of the world has concentrated on dealing with the widespread threat terrorism. [36]

2.8 Fault-Tolerance

If we have look at the combination of fault and tolerance, we may determine the fault such as malfunction or deviation, deflection from observed behavior and tolerance such as the capacity for bearable or enduring to something. Connecting the words together, a system's talent to deal with malfunctions is referred as fault-tolerance.

2.8.1 Faults

As we noted above, a fault is some deviation like a malfunction from the expected behavior of the system in a computer environments. Faults can be because of a different of factors, including operator-user error, software bugs, hardware failure and network problems in which discussed above.

Generally faults may categorized into three classes:

- Transient faults

Characteristics of this types of faults described such as: They occur once and then vanish. For instance, messages in a network doesn't reach its address but does when the action is retried.

- Intermittent faults

Intermittent faults are defined by a fault occurring, after that disappear again, then reoccurring and again disappear. Because of this characteristic intermittent faults are considered the most annoying of component faults. A poor connectivity is an instance of this type of fault.

- Permanent faults

This kind of failure is persistent, insistent: it ongoing to be until the faulty element is repaired or changed. Instance of these fault are burnt-out power supplies, disk head crashes and software bugs.

Any kind of these faults in which defined in above may be either a Byzantine failure or a fail-silent failure. A fail-silent fault is one in which the faulty unit cuts functioning and manufactures no bad result. More clearly, it either makes output or produces no output that implicitly demonstrates that the component has failed. A Byzantine fault is another kind of fault that the faulty unit keep up to execute but produces wrong results. If we compare both of them we can say that Byzantine faults is clearly more troublesome. The naming of Byzantine fault is come by the Byzantine Generals Problem.

2.8.2 Approaches to faults

As improvement of the technology, we can try to model systems that decrease the availability of faults. Fault avoidance is a method in which we go over design and evaluation steps to provide that the system escapes being faulty in the initial place. This can get in formal evaluation, checking, testing, code inspection and helping of robust hardware.

Fault removal is an old approach in which errors were came across in the system and we controlled to remove all this faults. This may have been made through debugging, verification and testing also changing failed elements with better ones such attach heat sinks to solve thermal squandering problems and so on.

Fault tolerance is the understanding that we will mostly have errors or chance for faults in our system. That's why we should model the system in which this approach will be tolerant of those faults. That is, the system should balance for the faults and keep going to function.

2.8.3 Achieving fault tolerance

The common method to establishing fault tolerant systems is called redundancy. This method can be applied in different levels.

The goal of information redundancy is to ensure fault tolerance throughput coding or replicating the data. For instance, a Hamming code can insure additional bits in data to recover, restore a current range of failed bits. Error Correcting Codes (ECC) memory, parity memory and ECC code on data blocks can be considered samples of information redundancy.

An operation different times may perform for achieving time redundancy fault tolerance. Can say some example for time redundancy such as retransmissions and timeouts in valid point-to-point and group connection. This samples of redundancy is profitable in the participation of intermittent or transient faults. This kind of redundancy is not useful with permanent faults. TCP/IP's retransmission of packets may be example of time redundancy fault tolerance.

The characteristics of physical redundancy is that it connected with devices not data. We supplement additional equipment to give an opportunity the system to tolerate the loss of some failed elements. There several types of examples in which include physical redundancy such as backup name servers and RAID disks.

When applying physical redundancy, we can distinguish redundancy from replication. In replication approach, we have several units operating at the same time and a quorum system to elect the outcome. But with physical redundancy method, just only one part is working while the redundant units are permanent by to fill in in circumstance the unit stops to work.

2.8.4 Availability Levels

In modelling a fault-tolerant system, we have to understand that 100% fault tolerance will never be achieved. In addition, the near we effort to obtain to 100% but in this approach system will be more costly.

To model an experimental system the degree of replication needed have to consider. Replication needed will be got from a statistical analyzing of reasonable acceptable manner. Incentives that enter into this analyzing method are the mean of worst-case performance in a system out of the average worst-case execution and faults in a system.

The below table illustrates some availability levels, general terms and the respectively annual downtime.

Class	Availability	Annual Downtime
Continuous	100%	0
Fault Tolerant	99.999%	5 minutes
Fault Flexible	99.99%	53 minutes
High Effectiveness	99.9%	8.3 hours
Normal Effectiveness	99 - 99.5%	44-87 hours

Table 1: Class types of Fault-Tolerance with availability percentage

2.8.5 Active replication

Active replication is a method for gaining fault tolerance over physical redundancy. A general instantiation of this approach is triple modular redundancy (TMR). This model grips whether 1-fault tolerance with Byzantine faults or two fault tolerance with fail-silent faults.

Beneath this system, we ensure component's threefold replication find and check a single element failure. For instance, think a system in which the output of A connect to the output of B whereas the output of B join to C (See Figure 12 on page 50). In this techniques, failure of one single component will motivate the entire system to fail.

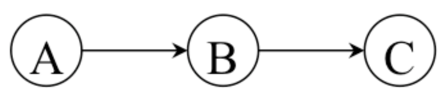


Figure 12: Simple example of No-Redundancy

In a Triple Modular Redundancy (TMR) model, we replicate every element with three methods and place electors after every stage to collect the large number of result of the stage (See Figure 13 on page 51). The elector is responsible for collecting the great number of winner of the 3 inputs. Two good votes will cancel a single Byzantine fault. At the time, the voters themselves are replicated because of malfunction problem.

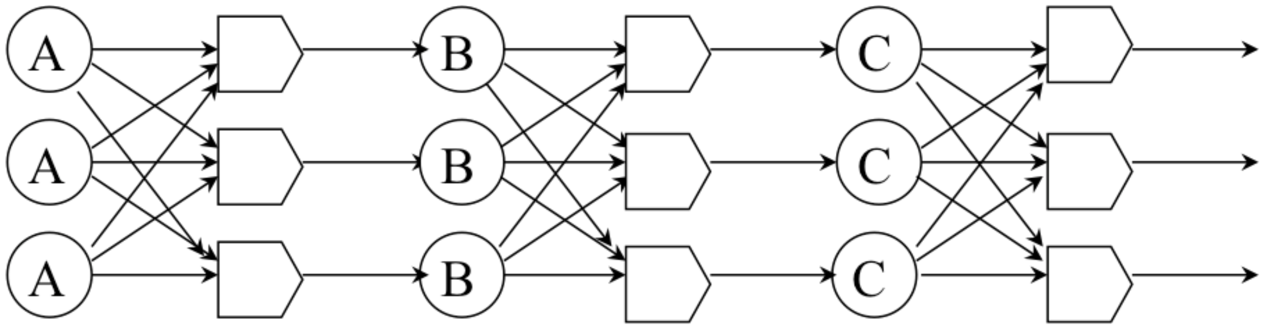


Figure 13: Architecture of TMR

2.8.6 Primary Backup (Active-Standby) approach

With an essential backup method, 1 server (the initial) make all the job and when the server fails the then backup system will take over job.

To discover whether a fundamental became insolvent, a backup can periodically humming the fundamental with "are you alive" messages. After getting it fails acknowledgement then the backup may accept that the initial failed and that's why it will take the functions of the initial server. If systems is asynchronous then this is a problem. Therefore redundant networks may help provide that a working contact channel exists. In addition previous method there is another possible way is stopping the primary forcibly by means of a hardware mechanism.

This system is comparatively easy to model when requests do not should go be multicast to a short of machines and there are no judgments to be made on who will get responsibility for failed work. An essential point to remark is that if one back up come across some problems on that time other back up take over the responsibility immediately. Backup is not good for Byzantine faults because it cannot define that the primary has really failed.

From a primary failure Recovery method can be complex or time-consuming according to needs for ongoing operation and application recovery.

2.8.7 Examples of fault tolerance

- **Error Correction Code memory**

Error Correction Code (ECC) memory contains additional logic that implements hamming codes to find and verify bit errors that are evoked by fabrication errors, cosmic ray radiation, electrical disturbances or neutron. An ordinary shape of error detection is an equality code: in this way, one additional bit shows whether the word is an odd or even number of bits. In this type of systems, a single bit error will reason the parity to inform a wrong value. Most applications of ECC memory employ a Hamming code that find two bit errors and make correct any single bit error according to per 64-bit word.

This ECC fault tolerance is a sample of information redundancy. As we know from above information coding is for ensuring fault tolerance of the data in memory and this codding request extra memory.

- **Machine failover via DNS SRV records**

In this type of systems the aim of fault tolerance is to give permission a user to join to one functioning mechanism that illustrates a given hostname. Some machines can be unachievable because of network or service connection.

For resolving a hostname problem to one IP, user can solve using DNS to look up records of SRV for that name. What is SRV record? It is a somewhat general record in DNS in which give opportunity one IP address to get information throughput available services. And every one of SRV record keeps a port, a weight, a priority and target hostname. For prioritizing of list of servers DNS use priority of hostname. In addition, for controlling equality system use weight value.

Using target hostname DNS can search for address of server. User will try another machine in list when server doesn't work.

This method is generally good for voice over IP (VoIP) systems to gather a SIP server amid several ready SIP servers for a special hostname. Such as VoIP DNS MX records also use this approach.

- **TCP retransmission**

As you know Transmission Control Protocol (TCP/IP) is the most valid actual circuit service transport layer protocol in which ensured on head of unreliable Network layer protocol. In TCP sender gets information (called acknowledgement) from the receiver when a packed is reached to address. As you know sometimes sender doesn't get any acknowledgement during some period of time. In this case, it thinks that packer wasn't reached to own place in other word it was lost. That's why according to TCP characteristic it will retransmit again. The transmission time differ because of operating systems for example in windows it is 3 seconds. In addition, five is default max number of retransmission. TCP is considered an instance of redundancy. [37]

3. Related work

Failure analysis and figuring of computer systems have been fully learned during the past two decades. All these researches include, for example, the sorting of general system failure properties and distributions [38], main reasons of analyzing failures [39], statistical and experimental analysis of errors in network systemss and failures [40], and the improvement and decomposition of methods to prevent and reduce failures in computer systems [41].

Many types of studies have been pressed on the root reasons of failure in different sorts of computer systems in which are not generally used for running Internet sites, and in operational atmospheres distinct those of Internet services. Computer system failure data was learned widely by J.Gray in which published about this topics studies of computer system failure data [42] [43]. Within 1986 he found that operator error was the largest single cause of failure in deployed Tandem systems, accounting for 42% of failures, with software the runner-up at 25% [43]. I also found brightly similar percentages at Online and Content.

Nevertheless, Gray found that software had become the main root of outages (about 55%), turning into marsh the second largest contributor and system operations (about 15%) in 1989.

Basic failure and stochastic error operations indicate time varying attitude and several models of robust connection architecture, with the inclusion of probably long-range dependence, significant cross-correlation amid less-serious errors and failures, significant periodic behavior and more- strict errors plus failures. In specific, patterns of failure and system error are frankly non-stable whereas these failures and errors consist of comparatively long time intervals in which all this futures are stationary, many spanning more than a day. In addition, the failures are not separated between the nodes uniformly and a little particle of the nodes suffer many types of the failures, with less than 4% of the nodes subsisting nearly 70% of the failures. The failures at the nodes subjecting the many failures as well have a powerful, provisional ratio with day time at the constant like hourly level and these patterns of temporal correlation modify in excess of time with such manner vary between the individual nodes. There are alternative curious directions for future work, in conditions of using collection of data as well as further work in data collection itself. Using this data, for enhancing system availability can be evaluated previously offered techniques. Besides, exploring of the probability of giving notice failures basically incidents happening at a node, or in width the network, and usefulness of there is used in proactive system maintenance. In data collection conditions, for procuring more data not only from other areas, but moreover within the production of system searched here in the hope of earning more understanding on how hardware and software progressions, and workload evolutions, effect the failure characteristics. [40]

Fault tolerance problems have also been applied in scientific workflow management systems (WMS). For example, as a basis the Pegasus WMS [44] has combined checking of task level systems, which repeats a job if system come across a task failure. Original data is also follow upped and used to study the failures reason’s [45]. A research of fault invention, protection, and recovery styles in current grid workflow management systems is exist in [46].From the this survey different methods supplies recovery collection technics like immigration, task replication, resubmission and check pointing. In this paper, I join some of these procedures with task clustering methods to increase the authenticity, stability, reliability and performance of fine-grained tasks. Best side of my research, none of the available WMS have satisfied like characteristics.

Many research papers have pressed about monitoring freely coupled task granularity. For instance, a clustering algorithm was offered in [47] in which groups bag of tasks according to the runtime, and afterwards new methods were proposed based on CPU time, task file size.

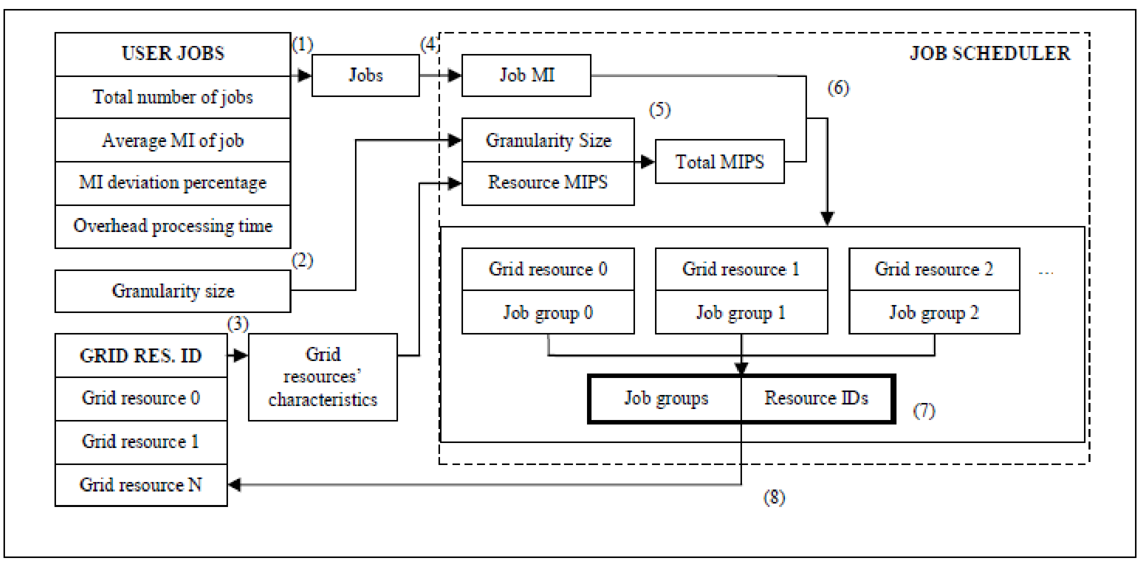


Figure 14: Dynamic job grouping-based scheduler's strategy of simulations

The job grouping strategic outcomes in raised performance in terms of low processing time and cost if it is addressed to a Grid application with a great number of jobs whereas minor processing requirements were hold by each user job. The total communication time and cost can increase because of sending/receiving each small job one by one to/from the resources. By the same token, the total processing abilities of each resource may not be completely utilized every time the resource takes a small scaled job. The aim of job grouping strategy to decrease the influence of these disadvantages on the total processing time and cost. Several groups of business strategy in accordance with the processing capabilities of the existing group of Grid resources and small business users. All this processed diminish the processing each user job's overhead time and communication overhead time. [47]

In recent years, they have offered online scheduling algorithm [48] which brings together tasks, based on the use of network resources, user's budget and deadline. Additionally, [49] and [50] also counted for increasing performance of network bandwidth in task scheduling algorithm. Longer tasks are assigned to resources with better network bandwidth. [51] Proposed an adaptive scheduling algorithm to group fine-grained tasks according to the processing capacity and the network bandwidth of the currently available resources.

Management of task granularity is also addressed in scientific workflows. Grouping tasks and overly computer technology can be very useful when planning large-scale computational- fine detail at the national cyber infrastructure workflows resources. We have presented a variety of clustering methods are embedded in display workflow Pegasus system and shows the outcomes of the planning and running the application on TeraGrid astronomy [52].

For achieving correctness of Grid-wide performance estimations of workflow activities is widely handicapped by designing partial information, and week capacity of existing methods to cover big alternations in some various factors. To give a solution, an immediate way sweating neural networks and Bayesian like methods of inference and searching to make a safer and circumstances-aware system for workflow activities execution time prediction is demonstrated in [53].

Besides this, an Instance Based Learning technique is used for prognosticating using historic workload traces on clusters according to application response times. A new kind of attributes (digital vector) is a source of status and enter the distance function to evaluate the distance between these functions. Locally scheduling technics is automatically incorporated in the state qualification of sources and historical data can be identified by genetic research [54].

4. Implementation

Irrelevant task clustering can negatively affect the workflow makespan in faulty distributed environments. In this section, I propose one fault-tolerant task clustering methods—Enhanced Selective Fault-Tolerant Task Clustering using database -that adjust the clustering size (k) of the jobs to reduce the impact of task failures on the workflow execution and giving retry count for restriction the count of retried failed task. These methods are based on the Selective Clustering (SC) [55] technique that has been implemented and used in the Java.

Selective Reclustering (SR): The selective re-clustering method, on the other words, combines only failed tasks inside of a clustered job into a new clustered job. Following algorithm (See Figure 16 page on 60) demonstrates Reclustering procedure algorithm for Selective Reclustering method. The Clustering and Merge procedures are also indicated in Figure 16 . In following, an example of the Selective Reclustering is shown also (See Figure 15 page on 60). As you see from Figure 15 in the initial attempt, the clustered job, merged of four tasks, has three failed tasks (t_1, t_2, t_3). 3 failed tasks are merged into a novel clustered job j_2 and repeated. This point of view does not mean to regulate the size of clustering size, whereas the clustering size may be smaller after each execution in which this indicate that each subsequent clustered jobs may execute fewer tasks. In the following example, k has diminished from four to tree. The advantage of SR is that it is simple to perform.

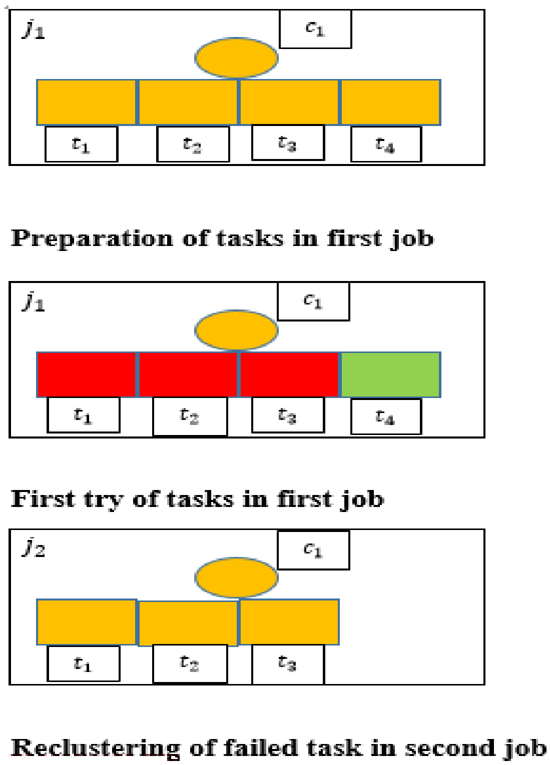


Figure 15: An example of Selective Reclustering

```
Require: W: workflow; C: max number of tasks per job defined by
    clusters.size or clusters.num
procedure CLUSTERING(W,C)
    for level < depth(W) do
        TL ← TASKSATLEVEL(W,level)   ▷ Divide W based on depth
        CL ← MERGE(TL,C)             ▷ Returns a list of clustered jobs
        W ← W − TL + CL              ▷ Merge dependencies as well
    end for
end procedure
procedure MERGE(TL,C)
    J ← {}                           ▷ An empty job
    CL ← {}                          ▷ An empty list of clustered jobs
    while TL is not empty do
        J.add (TL.pop(C)             ▷ Pops C tasks that are not merged
        CL.add( J)
    end while
    return CL
end procedure
procedure RECLUSTERING(J)            ▷ J is a failed job
    TL ← GETTASKS(J)
    J_new ← {}                       ▷ An empty job
    for all Task t in TL do
        if t is failed then
            J_new.add (t)
        end if
    end for
    W ← W + J_new                    ▷ Re-execute it
end procedure
```

Figure 16: Selective Reclustering Algorithm

Enhanced Selective Fault-Tolerant Task Clustering Method using Database:

Algorithm for this method is described in following:

id	Process URL	Response	Done date	Retry count	Insert date
1	Google.com	Response text	null	15	Now()
2	Facebook.com	Response text	null	0	Now()
3	Twitter.com	Response text	nul	2	Now()
…	…	…	…	…	…

Table 2: General example view of database

Function **getAllUnprocessedTasks** (int taskCount) {

// the aim of this function is taking all unfinished tasks from database according to tackCount in other words take task in which Done date is null and Error count<15

}

Function **setTaskAsDone** (int task_Id) {

//make the status of Done date is now

}

Function **setTaskAsError** (int task_Id) {

//increase one unit the Retry count of task

}

Function **processTask** (task_Id, URL) {

//send request to URL if it is success

// **setTaskAsDone (taskId);**

//else

// setTaskAsError (taskId);

}

Function **clustering** (tasks, task_Count) {

//make job from tasks according to taskCount and jobs works individually

}

Function **beginProcess** () {

Tasks = getAllUnprocessedTasks(taskCount);

clustering(tasks,200);

//make jobs and jobs begin to execute own works and wait for finishing of jobs

beginProcess();

}

Function **main () {**

beginProcess ();

}

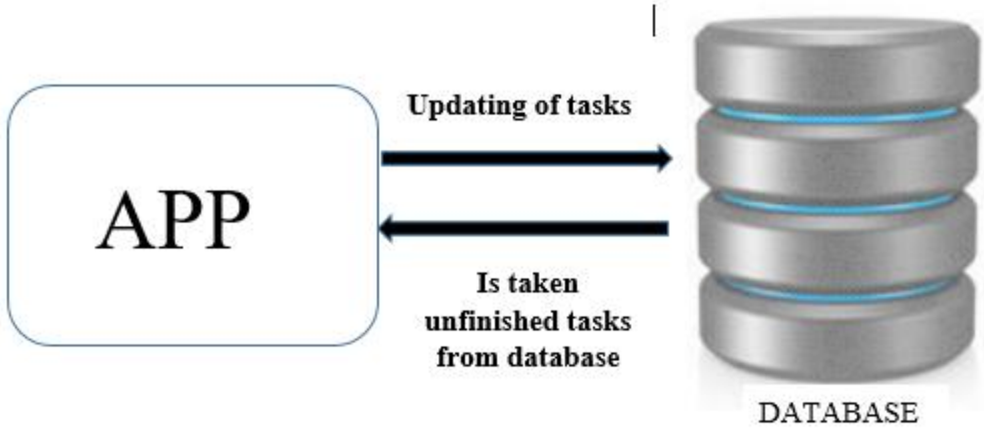


Figure 17: Scheme of Enhanced Selective Fault-Tolerant Task Clustering Method using Database

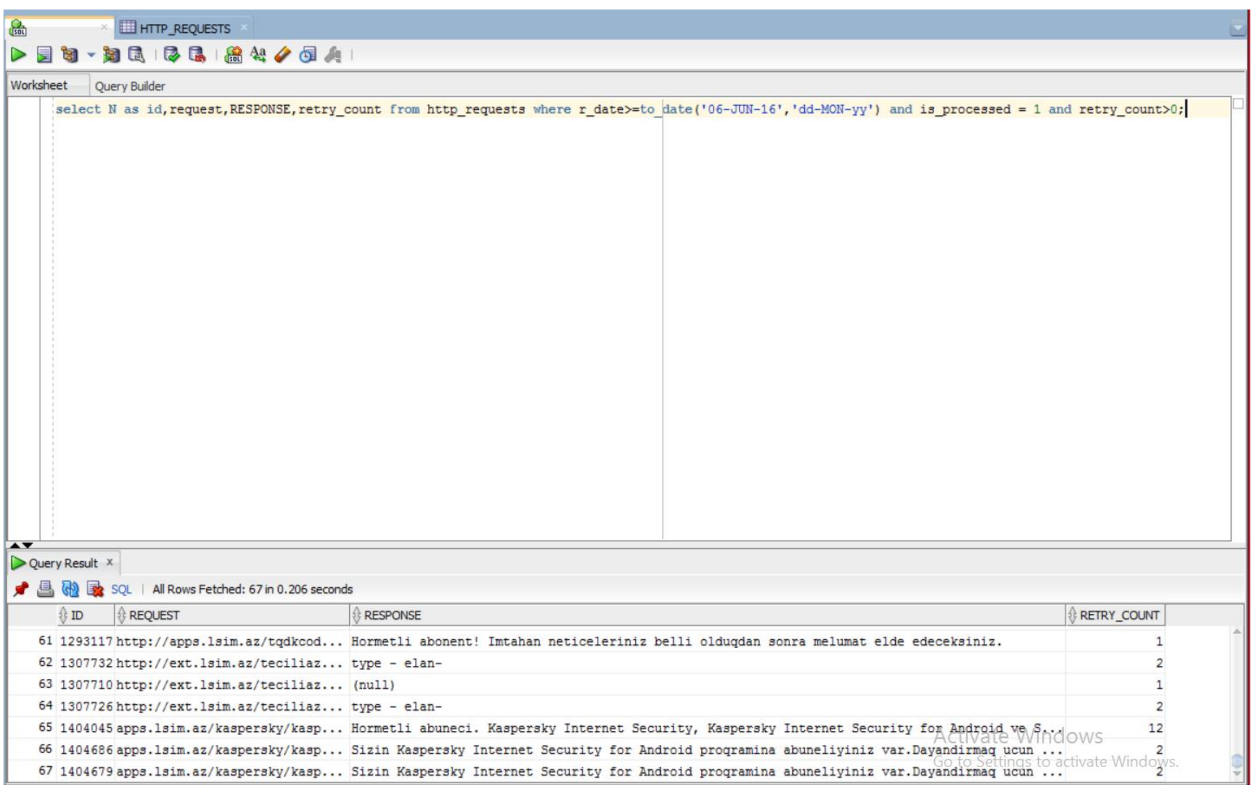
Difference between this two methods is that in first approach array was use for tasks and clustering. But in my approach database is use in which this way is more appropriate. The benefits of second method is more than in comparison with first way. As we know from the selective clustering it give us decreasing system overhead, increasing performance of systems. Beside this functionalities second method give us following advantages:

- ❖ From the database we can define which task finished or unfinished. In addition, the executing time of tasks can be determined by using simple formula such as: Done date – Insert date
- ❖ No any loss of process. For example if any process cannot accomplish because of any reason such as crashing of program or other reasons then process will not loss. Because in database this process is marked such as undone. If we do this with array (with first method) then all information about task will lose.

- ❖ You can give any modifications to program. For instance you can add new modules into your program or other modules can use this information helping of database.
- ❖ If execution of task was delayed then looking of Retry count you can define how many repeting of task it finsihed and also you can search for reasing of task execution delay.
- ❖ Existence of Retry count give us opportunity to restrict entering tasks into infinite cycles because of unexecution. For example, if we have 15000 tasks and 400 of them are undone and execution of this tasks repeatedly means that waste of CPU, waste of resourches and others.

From the above noted benefits of this method just in simple way you know the good aspects of database. All this charachteris also make more strong this method.

For example using just simple query in database we can get infromation about tasks in which interested us:



Conclusion

In this dissertation paper, I proposed new method for clustering. Firstly for good understating, I gave information about workflow, distributed resources, task clustering, failures, fault tolerance and main aspects of all this issues briefly. Good side of clustering in large-scale distrusted systems, performance of workflow because of failures in which may be come across within execution of tasks in job. The impact of failures to system in distributed resources, how can design system for achieving good performance, decreasing system overheads, making systems more tolerant in faulty environments, decreasing of cost, waste of resources and other factor in which investigated and learned in this paper. Recently using of distributed systems are more general and investigated clustering strategies cancel or don't estimate rightly the impact of failures over the system, or just focused to improve performance of systems in spite considering other problems. In this work, I focused particularly on just not decreasing system overhead or performance also considering transient failures, impact of them over system, diminish cost and waste of resources, more information about tasks and so on.

In this work, I proposed Enhanced Selective Fault-Tolerant Task Clustering Method using Database in which this way give us more opportunity doing action with failures for futures works. Learning reason of failures, adding some other functionality using database abilities and other factors in which noted in above.

The final screen also show us useful sides of method in which this method can use in many environments because of above commented benefits.

Bibliography

[1] V.Curcin and M.Chanem, "Scientific workflow systems - can one size fit all?," in *Cairo Internationl Biomedical Engineering Conference,* Cairo, 2008.

[2] D. Talia, "Workflow Systems for Science: Concepts and Tools," *International Scholarly Research Notices Software Engineering,* vol. 2013, pp. 1-15, 2013.

[3] G. Bell, T. Hey and A. Szalay, "Beyond the data deluge," *Science,* vol. 323, no. 5919, pp. 1297-1298, 2009.

[4] T. I.J, D. E, G. D.B and S. M, Workflows for e-Science: Scientific Workflows for Grids, UK: Springer, 2007.

[5] M. M., P. M. and G.aloisio, "The ProGenGrid virtual laboratory for bioinformatics," in *Computer-Based Medical Systems (CBMS), 2012 25th International Symposium on*, 2012.

[6] V. d. A. W.M.P., T. H. A.H.M., K. B. and B. A.P., "Workflow Patterns," *Distributed and Parallel Databases,* vol. vol.14, no. no.1, pp. pp.5-51, 2003.

[7] M. John A., P. Devanand, S. Amit P, K. Krys J. and S. Harvinder, "WebWork: METEOR2's Web-Based Workflow Management System," *Journal of Intelligent Information Systems,* vol. vol.10, no. no.2, pp. pp 185-215, March 1998.

[8] L. F. and R. D., "Business process management with FlowMark," in *in Proceedings of the IEEE Computer Society International Conference (COMPCON '94)*, 1994.

[9] L. B., A. I. and B. C., "Scientific workflow management and the Kepler system," *Concurrency and Computation: Practice and Experience,* vol. vol.18, no. no.10, pp. Pages 1039-1065, 25 August 2006.

[10] T. Glatard, G. Sipos, J. Montagnat, Z. Farkas and P. Kacsuk, "Workflow-Level Parametric Study Support by MOTEUR and the P-GRADE Portal," in *Workflows for e-Science*, New York, NY, USA, Springer London, 2007, pp. pp 279-299.

[11] B. Mathew, M. B. Juric and P. Sarang, Web Services Business Process Execution Language Version 2.0, Primer, 9 May 2007.

[12] T. Fletcher, P. F. C. Ltd, A. Green and R. Haugen, "Grid Service Orchestration Using the Business Process Execution Language (BPEL)," *Journal of Grid Computing,* vol. vol.3, no. no.3-4, pp. pp 283-304, September 2005.

[13] M. Alt, A. Hoheisel, H.-W. Pohl and S. Gorlatch, "A Grid Workflow Language Using High-Level Petri Nets," in *Parallel Processing and Applied Mathematics,* vol. 3911, New York, Springer Berlin Heidelberg, 2006, pp. pp 715-722.

[14] M. Atay, A. Chebotko, D. Liu, S. Lu and F. Fotouhi, "Efficient schema-based XML-to-Relational data mapping," *Information Systems,* vol. vol.32, no. no.3, pp. pp.458-476, May 2007.

[15] A. Chebotko, M. Atay, S. Lu and F. Fotouhi, "XML subtree reconstruction from relational storage of XML documents," *Data and Knowledge Engineering,* vol. vol.62, no. no.2, pp. pp.199-218, August 2007.

[16] T. Andrews, F. Curbera and H. Dholakia, "Business Process Execution Language for Web Services Version 1.1," 5 May 2003. [Online]. Available: https://msdn.microsoft.com/en-us/library/ee251594(v=bts.10).aspx. [Accessed 20 April 2016].

[17] F. Neubauer, A. Hoheisel and J. Geiler, "Workflow-based Grid applications," *Future Generation Computer Systems,* vol. vol.22, no. no.2-3, pp. pp.6-15, January 2006.

[18] G. v. Laszewski and M. Hategan, Java CoG Kit Karajan/Gridant Workflow Guide, 2005.

[19] G. v. Laszewski, M. Hategan and D. Kodeboyina, "Java CoG Kit Workflow," in *Workflows for e-Science*, Yew York, Springer London, 2007, pp. pp.340-356.

[20] T. Oinn, M. Addis and J. Ferris, "Taverna: A tool for the composition and enactment of bioinformatics workflows," *Bioinformatics*, vol. vol.20, no. no.7, pp. pp.3045-3054, November 2004.

[21] T. Oinn, M. Greenwood and M. Addis, "Taverna: Lessons in creating a workflow environment for the life sciences," *Concurrency Computation Practice and Experience*, vol. vol.18, no. no.10, pp. pp.1067-1100, 25 August 2006.

[22] I. Taylor, M. Shields, I. Wang and O. Rana, "Triana Applications within Grid Computing and Peer to Peer Environments," *Journal of Grid Computing*, vol. vol.1, no. no.2, pp. pp.199-217, 2003.

[23] I. Taylor, M. Shields, I. Wang and A. Harrison, "Visual Grid Workflow in Triana," *Journal of Grid Computing*, vol. vol.3, no. no.3, pp. pp.153-169, September 2005.

[24] I. Taylor, E. Al-Shakarchi and S. D. Beck, "Distributed audio retrieval using Triana, DART," in *International Computer Music Conference (ICMC '06)*, New Orleans, Lo, USA, November 2006.

[25] T. Fahringer, R. Prodan and R. Duan, "ASKALON: A grid application development and computing environment," in *IEEE/ACM International Workshop on Grid Computing*, 2005.

[26] T. Fahringer, Q. Jun and S. Hainzer, "Specification of Grid workflow applications with AGWL: An abstract Grid workflow language," in *2005 IEEE International Symposium on Cluster Computing and the Grid, CCGrid 2005*, 2005.

[27] V. H. Doina Caragea, "Learning Classifiers from Distributed Data Sources," in *Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends*, 2009, p. 17.

[28] N. Iacob, "Fragmentation and Data Allocation in the Distributed Enviroments," *Annals of the University of Craiova-Mathematics and Computer Science Series*, vol. 38, no. 3, p. 76–83, 2011.

[29] Intel, "Granularity and Parallel Performance," 1 February 2012. [Online]. Available: https://software.intel.com/en-us/articles/granularity-and-parallel-performance. [Accessed 2016].

[30] R. C. Tryon and D. E. Bailey, Cluster Analysis, New York: McGraw-Hill, 1970.

[31] L. Rokach and O. Maimon, "Clustering Methods," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds., Boston, MA, Springer, US, 2005, pp. 321-352.

[32] P. Berkhin, "Survey of Clustering Data Mining Techniques," in *Grouping Multidimensional Data: Rescent Advances in Clustering*, J. Kogan, C. Nicholas and M. Teboulle, Eds., Berlin, Helderberg, Springer, Berlin, Helderberg, 2006, pp. 25-71.

[33] L. NCSS, "Hierarchical Clustering / Dendrograms," 2016. [Online]. Available: http://www.ncss.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Hierarchical_Clustering-Dendrograms.pdf. [Accessed 2016].

[34] "BCS The Chartered Institute for IT", Claud computing movint IT out of the office, British Informatics Society Limited, 2012.

[35] J. Joseph and C. Fellenstein, Grid Computing, Prentice Hall PTR, 2004.

[36] R. Hudyma and D. I. Fels, "Causes of Failure in IT Telecomunications Networks," in *Proceedings of SCI 2004*, Florida, 2004.

[37] P. Krzyzanowski, "Fault Tolerance Dealing with an imperfect world," 23 August 2010. [Online]. Available: https://www.cs.rutgers.edu/~pxk/rutgers/notes/content/ft.html. [Accessed 2016].

[38] D. Tang, "Failure analysis and modeling of a vaxcluster system," *Int. Symp. on Fault-tolerant computing*, pp. pp.1-8, 1990.

[39] B. Schroeder, C. M. University and G. A. Gibson, "A large-scale study of failures in high-performance computing systems," *International Conference on Dependable Systems and Networks (DSN'06),* pp. pp.249 - 258, June 2006.

[40] R. K. Sahoo, A. Sivasubramaniam, M. S. Squillante and Y. Zhang, "Failure data analysis of a large-scale heterogeneous server environment," in *Dependable Systems and Networks,* 2004.

[41] D. Oppenheimer, A. Ganapathi and D. A. Patterson, "Why do internet services fail, and what can be done about it?," in *USITS'03 Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems*, 2003.

[42] J.Gray, "A census of Tandem system availability," Tandem Computers Technical Report 90.1, 1990.

[43] J.Gray, "Why do Computers stop and what can be done about it?," 1986.

[44] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan and P. J. Maechling, " Pegasus, a workflow management system for science automation, QCG-OMPI: MPI Applications on Grids," *Future Generation Computer Systems,* pp. pp.1-42, 2014.

[45] T. Samak, D. Gunter, M. Goode, E. Deelman, G. Mehta and F. Silva, "Failure prediction and localization in large scientific workflows," in *6th workshop on Workflows in support of large-scale science,* New York, NY, USA, 2011.

[46] K. Plankensteiner, R. Prodan, T. Fahringer and P. K. A. Kert´esz, "Fault detection, prevention and recovery in current grid workflow," *Grid and Services Evolution,* pp. pp.1-13, 2009.

[47] N. Muthuvelu, J. Liu, N. L. Soe, S. Venugopal, A. Sulistio and R. Buyya, "A dynamic job grouping-based scheduling for deploying applications with fine-grained tasks on global grids," in *ACSW Frontiers '05 Proceedings of the 2005 Australasian workshop on Grid computing and e-research,* 2005.

[48] N. Muthuvelu, E. I. Chai, Chikkannan and R. Buyya, "On-Line Task Granularity Adaptation for Dynamic Grid Applications," in *Algorithms and*

Architectures for Parallel Processing, vol. vol.6081, Springer Berlin Heidelberg, 2010, pp. pp.266-277.

[49] W. K. Ng, T. Ang, T. Ling and C. Liew, "Scheduling framework for bandwidth-aware job grouping-based scheduling in grid computing," *Malaysian Journal of Computer Science 19(2)*, p. pp.117–126, January 2006.

[50] T. Ang, W. Ng, T. Ling, L. Por and C. Liew, "A Bandwidth-Aware Job Grouping-Based Scheduling on Grid Environment," *Information Technology Journal*, vol. vol.8, pp. pp.372-377, 2009.

[51] Q. Liu, W. U. o. T. W. Sch. of Inf. Eng. and Y. Liao, "Grouping-Based Fine-Grained Job Scheduling in Grid Computing," in *Education Technology and Computer Science, 2009. ETCS '09. First International Workshop on*, Wuhan, Hubei, March 2009.

[52] u. Singh, M.-h. Su, K. Vahi, E. Deelman, B. Berriman, J. Good, D. S. Katz and G. Mehta, "Workflow task clustering for best effort systems with pegasus," in *15th ACM Mardi Gras conference*, New York, NY, USA, 2008.

[53] R. Duan, F. Nadeem, J. Wang, Y. Zhang, R. Prodan and T. Fahringer, "A Hybrid Intelligent Method for Performance Modeling and Prediction of Workflow Activities in Grids," in *CCGRID '09 Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, Washington, DC, USA, 2009.

[54] H. Li, D. Groep and L. Wolters, "Efficient Response Time Predictions by Exploiting Application and Resource State Similarities," in *GRID '05 Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, pp.234-241 , 2005.

[55] W. Chen, R. F. d. Silva, E. Deelman and T. Fahringer, "Dynamic and Fault-Tolerant Clustering for Scientific Workflows," *IEEE Transactions on Cloud Computing* , vol. vol.4, no. no.1, pp. pp.49 - 62, 28 April 2015.

[56] W. Chen and E. Deelman, "Proceedings of the 6th workshop on Workflows in support of large-scale science," in *SC11 International Conference for High Performance Computing, Networking, Storage and Analysis*, New York, NY, USA, 2011.

[57] S. Kalayci, "Distributed and adaptive execution of condor," in *22nd International Conference on Software*, 2010.

[58] R. F. d. Silva, G. Juve, E. Deelman, T. Glatard, F. Desprez, D. Thain, B. Tovar and M. Livny, "Toward fine-grained online task characteristics estimation in scientific workflows," in *WORKS '13 Proceedings of the 8th Workshop on Workflows in Support of Large-Scale Science*, New York, NY, USA, 2013.