Computer Engineering and Applications Vol. 1, No. 2, December 2012

Monitoring and Resource Management in P2P GRID-based Web Services

Djafri Laouni and Mekki Rachida
Department of Computer Science, Faculty of Science, USTO University
djaafri29tp@gmail.com
mekki@univ-usto.dz

ABSTRAKSI

Komputasi grid baru-baru ini muncul sebagai respon dari meningkatnya permintaan untuk sumber daya (kekuatan pemrosesan, penyimpanan, dll). ditunjukkan oleh aplikasi - aplikasi ilmiah. Namun, dikarenakan peningkatan ukuran jaringan, kebutuhan dari pengorganisasian mandiri dan pengaturan ulang yang dinamis menjadi semakin penting. Karena sifat tersebut ditunjukkan oleh sistem P2P, konvergensi komputasi grid dan komputasi P2P tampak alami. Namun, dengan menggunakan sistem P2P (biasanya berjalan di Internet) pada infrastruktur jaringan (umumnya tersedia sebagai federasi SAN berbasis cluster interkoneksi oleh high-bandwidth WAN) dapat mengangkat masalah kecukupan mekanisme komunikasi P2P. Diantara sifat – sifat yang menarik dari system P2P adalah votalitas dari bagian – bagian yang menyebabkan kebutuhan dari integrasi terhadad system toleransi kesalahan. Dan layanan load balancing. Sebagai solusi, kami menawarkan sebuah mekanisme dari toleransi kesalahan dan model load balancing yang diadaptasi untuk model P2P grid. Yang dinamakan SGRTE (Monitoring dan Manajemen Sumber Daya, Toleransi kesalahan dan Load Balancing).

Kata Kunci: perhitungan grid, system peer to peer, grid p2p, toleransi kesalahan, load balancing, layanan web

ABSTRACT

Grid computing has recently emerged as a response to the growing demand for resources (processing power, storage, etc.) exhibited by scientific applications. However, as grid sizes increase, the need for self-organization and dynamic reconfigurations is becoming more and more important. Since such properties are exhibited by P2P systems, the convergence of grid computing and P2P computing seems natural. However, using P2P systems (usually running on the Internet) on a grid infrastructure (generally available as a federation of SAN-based clusters interconnected by high-bandwidth WANs) may raise the issue of the adequacy of the P2P communication mechanisms. Among the interesting properties of P2P systems is the volatility of peers which causes the need for integration of a service fault tolerance. And service Load balancing, As a solution, we proposed a mechanism of fault tolerance and model of Load balancing adapted to a grid P2P model, named **SGRTE (**Monitoring and Resource Management, Fault Tolerances and Load Balancing**)**.

Keywords: *Grid computing, PeerToPeer systems, gridP2P, faults tolerance,* Load balancing, web services.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

1. INTRODUCTION

Recent years have seen the emergence of IT structures of large sizes through cooperation between a large number of machines. These environments, by aggregating the resources of thousands of computers, applications offer new opportunities in terms of computing power, storage capacity and distribution programs. Two types of software infrastructures have emerged: the grids (GRID) which consist of the interconnection of clusters (Cluster) geographically distributed and architectures Peer to Peer (Peer to Peer) where a set of sites cooperate as equal to equal, hence the birth of the grid-P2P (P2P Grid).

The occurrence of faults and blockages in these systems is inevitable, treatment of these faults, reduce communication costs induced labor migration, are among the most difficult problems. Fault tolerance is among the best solutions for processing errors in these environments. These systems must also provide redundancy mechanisms, detection and localization errors and reconfigure the system to retrieve the error case.

The load balancing is used to distribute the requests to a service across multiple physical nodes providing this service. This function is typically used for network services such as Web servers, email, DNS ... It is part of a product as high availability load balancing does not redirect requests to a failed server.

The interest of our solution by integrating fault tolerance algorithms and load balancing algorithms in a common source code (code coupling) to give an ideal architecture in distributed systems (there is no application started on a compute node when a failure is detected, and there is no time saving when a peer is saturated).

We develop a model of p2p grid to manage fault tolerance and load balancing based on a structured grid into groups consisting of a set of peers that can submit and execute services (web services) in the grid to contrast to client / server architectures, plus this model has a mechanism for fault tolerance and load balancing inside and outside groups, and we study how a P2P grid and then tackles the problem of fault tolerance and also the problem of load balancing in grid P2P model. and it concludes with prospects (as proposed) to "cloud computing".

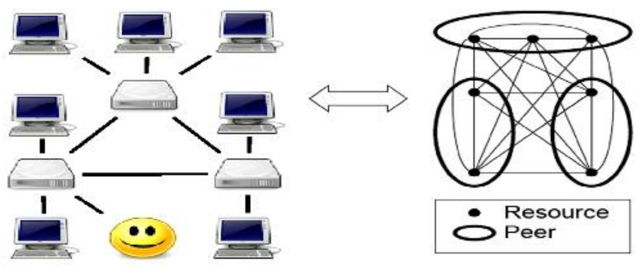

Figure 1. Peer-to-Peer Grid of resources

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

2. STATE OF THE ART

2.1. CONVERGENCE OF GRID AND PEER TO PEER[1]

Today, the pure number of desktop systems is the potential benefits of interoperability between desktops and servers in a single grid system very powerful. While a parallel situation exists, the differences between grid computing and P2P computing from their uses. Grids computing have been used for scientific computing, while the P2P computing gained importance in the context of the exchange of multimedia files. P2P is to mass collaboration computing devices. There are no such constraints on its architecture, which makes P2P computing more flexible. In fact, it uses the computing power to limit rather than a connection in the network. The client / server architecture does not exist in a P2P system. Instead, the peer nodes act as clients and servers, their roles are determined by task characteristics and system status. This architecture reduces the workload per node, and optimizes the use of all computing resources from the network.

Unlike the scope of Grid computing in scientific research, primarily offers P2P file sharing (eg Napster and Bittorrent), distributed computing (SETI @ home, for example), and anonymity (for example, Publius). Although two types of computing architectures have both a conceptual and practical distinction, their convergence is significant, "The vision behind both P2P and grid computing-that of a computer world where access to resources and services can be negotiated if needed will come to pass if we have succeeded in developing a technology that combines elements of what we now call P2P and Grid Computing.

Although the grids computing are widely accepted in the domain of scientific research, architecture-based server in the local context and nature of middleware in the (global)overall context limits its applications in open environments, where nodes are highly autonomous and heterogeneous, and their availability varies from time to time. An example of an open environment is the Internet, where vast resources are unoccupied, that are not normally organized in terms of computing power for a certain purpose.

2.2. USE OF TECHNIQUES FOR BUILDING P2P GRIDS [2]

Today, scientific applications require more and more resources, such as processors, storage devices, network links, etc.. Grid computing provides a response to this request by the treatment of aggregation and storage of resources made available by various institutions. As their sizes grow, grids express an increasing need for flexible distributed mechanisms allowing them to be managed effectively. These properties are exposed by P2P systems, which have proven their ability to efficiently manage millions of interconnected resources in a decentralized manner. Moreover, these systems support a high degree of volatility of resources. The idea of using P2P approaches to resource management of the grid has emerged naturally.

To our knowledge, very few attempts performed, convergence between P2P and grid computing have taken two different paths. One approach is to implement services on P2P software construction based on grid technologies (eg the use of grid services as a communication

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

layer). Instead, libraries of P2P can be used on grid infrastructures underlying physical layer as a service grid highest level. It's a way to leverage evolutionary mechanisms for P2P resource discovery, resource replication and tolerance faults. Grid applications often have significant performance constraints. In most cases, the grids are built as cluster federations. System-Area Networks (SANs), such as Giga Ethernet or Myrinet (which typically provide Gb / s of bandwidth and latency of a few microseconds), are used to connect nodes in a cluster of high performance, while wide-area Networks (WANs), provide a type of bandwidth of 1 Gb / s, but higher latency (typically of the order of 10-20 ms), are used to connect clusters. Therefore, sending a message between two nodes in the same SAN can be 1,000 times less expensive then do the same operation through a WAN. Such a discrepancy cannot be neglected, as the efficient use of network characteristics is a crucial issue in the context of the performance of scientific applications. However, P2P applications have not generally performance constraints important because they tend to focus on the edges of the Internet (with a low bandwidth and high latency, such as Digital Subscriber Line connections (DSL)). In this context, the latency between arbitrary pairs of nodes does not change much. Therefore, most published articles on P2P systems generally model of communication cost as distance based on the number of logical hops between communication entities, regardless of the underlying physical topology. When running P2P protocols on grid infrastructures, this factor must be clearly taken into account to effectively utilize the capabilities of existing networks to deliver the performance required by applications. Therefore, the use of libraries P2P Grid on infrastructures such as building blocks for grid services is a difficult problem, because it is clearly an unusual deployment scenario for P2P systems.

Therefore, it's important and legitimate to ask: does the P2P communication mechanisms are adequate for use in this context? Is it possible to adapt P2P communication systems to benefit from a high potential of these high-performance networks, to meet the needs of scientific grid applications?

2.3. FAULT TOLERANCE

Fault tolerance is one means of dependability whose principle is to respond to erroneous cases of a system and to prevent these errors lead to a malfunction visible to the user of the system considered [3].

In the domain of fault tolerance, fault is defined as a cause that causes an error, and error is a corrupted case in which is the system or application, such a case that can then cause a failure. Failure, in turn, manifests itself as a deviation from the behavior of the application from the correct behavior defined by the specification of the application. The construction of a system fault tolerance is mainly based on two mechanisms: the detection of a fault, then treating the fault detected [4].

2.4. LOAD BALANCING [5]

Load balancing is the technique to distribute work between different computers in a group unknown to the user. This technique provides very powerful features, and is used in large Internet servers to reduce response times, increase scalability and server availability.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

Load balancing is the technique to distribute work between different computers in a group unbeknownst to the user. This technique provides very powerful features, and is used in large Internet servers to reduce response times, increase scalability and server availability.
The traditional objective of a load balancing approach oriented system is to minimize the (global) overall execution time of applications and optimize the average response time of demands.

The application-level balancing, is to adjust system resources to the particular characteristics of a given application, in order to minimize the makespan (is an important performance criterion for parallel systems), regardless of the execution of other applications in the system.

The optimal exploitation of grid computing can be measured with two metrics: load balancing between resources of a grid and reduced communication costs.

The system -level Balancing**,** also called distributed scheduling is to maximize overall system performance, which is to minimize the overall response time of applications, assigning proper tasks to different resources of a system.

3. THE REALIZED WORK

we propose, as a solution:
Coupling of P2P networks and grid computing, and also the code coupling, to solve the problem of fault tolerance and load balancing problem, either in the peer or super peers.

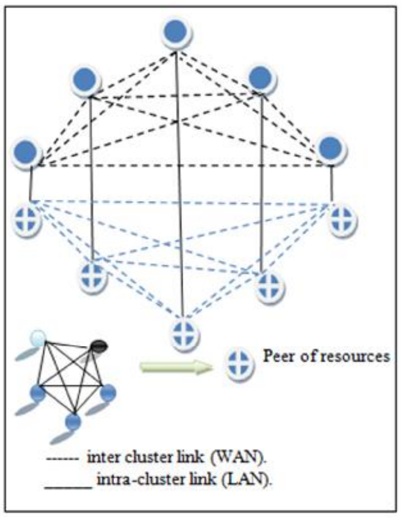
Peer of resources
------ inter cluster link (WAN).
______ intra-cluster link (LAN).

Figure 2. Proposed Model SGRTE(case General)

3.1 DESCRIPTION

The proposed grid model is formed of a set of groups interconnected by a WAN link (see Figure 2). The user can submit its web services from any peer group. Upon submission of the SW, the peer owner supports for his execution in the grid until its completion. and Submitting results of execution.

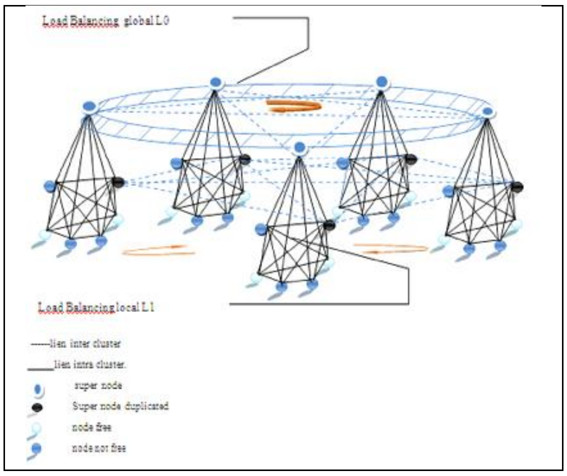

Figure 3. Proposed Model SGRTE (meshed network) Detailed

The positive things that distinguish the model ***SGRTE*** over other distributed architectures that currently exist are:

- The two levels L0 and L1 can become one level In the case of malfunction of all super nodes at the same time, with the operating system in its normal case.

-Group is inseparable from the other groups of ***sgrte***, unlike what is proposed by Yagoubi and meddber [6], where all groups are independent of the other groups, in the event of default level peers, local imbalance at the cluster level, same level of overall ***sgrte***. Because the peers that are independent of each other.

- Assign tasks to an outside group, in coordination with the super-peers. So the situation ***sgrte*** overall system is very balanced.
Our architecture is based on the following actors:

A.Peer

Upon receipt of a web service, you can run it locally as it can submit another peer of group, asking the Super Node list of peers available either locally or globally , and it selects a peer destination to start the web service remotely by direct exchange and receipt of the results the same way. Each peer is considered an owner or consignee, or even both at the same time. And each peer may perform services other peer cluster (inter / extra-cluster), if it is free. Property of our architecture, so we can have a single-level model.

Analysis:

In the peer level (Level 1): At this level, we find the elements for calculating the grid connected to their respective clusters. Each computing element at this level for role:

If you have a problem, **then** the peer selected continue the computation (update (backup to a log file periodically and momentarily)) no re-computation (minimizing response time).

Periodic collection of informations to its load, capacity, speed. **For** each compute node NCi of Cluster (intra-group) **Do** Sending to its current load Chari, CPi, Vi, its manager associated SN, and execute the load balancing decision. **End.**

PEER
(Private) Hote_name : chaine de caractères. *(Public) Peer_DB :Data base* *SW_List :Table* Id_SW : Chaine de caractère. Peer_Prop :Chaine de caractère. SW:chaine de aractères. Stat_SW *: « Pending », «Active », «Done », «Failed ».* *Table_SW_Done : Table* Id_SW *:Nombre entier.* Nb_Duplic : *Nombre entier.* Prop*: Chaine de caractère.* Chargi:double; Vi: double; CPi:double;
(Public) Procedure Lancer_SW(Id_SW, Peer_Prop,SW,Nb_Duplic) *(Public)* Procedure Collect_info_charge(Char*i*,CP*i*,V*i*) *(Public)* Procedure SW_Done(Id_SW, Peer_Prop,SW) *(Public)* Procedure Receive_SW(Peer_Prop,Id_SW,SW) *(Public)* Procedure Suppression_SW(Peer_Prop,Id_SW,SW) *(Public)* Procedure SW_NotSend() *(Public)* Procedure Receive_Resultat() *(Public)* Procedure Exe_Dec_Eq() *(Public)* Procedure SW_Send()

B. SuperNode (SN)
This is a manager; it contains all information's about its peer of group and other groups also.
a. If a peer wants to launch a remote web service, it requests the list of peers SuperNode available for it to launch its web service.

b. The SuperNode monitors the peers, if a peer fails "Crash", it detects the fault and tolerant the web services of the latter.

c. The SuperNode detects the fault of SuperNodeDupliqued, if the fault is "Crash", the SuperNode selects another peer to assign the directory and duplicated it sees it as a new SuperNodeDupliqued.

d. The SuperNode is responsible for tolerating faults in other groups of the grid in case of inability to tolerate them locally.

analysis:

At the super-peer (Level 0)
This level contains SN cluster, each node:
- Control Grid (detects faults, tolerate fault, update super node duplicated);

- manages the local load of its associated cluster;

- Participate in balancing the global load of the grid.

- To estimate the current load Charc, VG speed, capacity and GC Texe_m group G;

- Calculate the standard deviation ¾ G = μ.

- load balance with decision making and the saturation test.

- Transfer tasks to be performed.

- Send load information of other G SN grid (inter-cluster).

SUPER PEER
(Private) Hote_name : chaine de caractères.
(Public) Annuaire :Data base
Peers_Infs :Table
Peer :Chaine de caractéres.
Taille_maxFile d'attente :Nombre entier.
@IP : String.
Stat_Peer : « Connected », «Not connected».
Nb_services web:Nombre entier.
Taille_Disp :Nombre entier.
Fault_Type : « Crash », « Déconnexion ».
SND :Booléan. /* SND : super noeud dupliqué du group
SN_Gr : Boolean /* SN_Gr :super noeud d'un autre group
Verify_Dup :table
Peer_ Prop :Chaine de caractéres.
Id_service web :nombre entier.
Nb_Duplic :Nombre entier.
Table_Services web :Table
Peer :Chaine de caractères.
Id_service web :nombre entier.
Stat_service web : « En attente », «En exécution », «Done ».
Service web :Chaine de caractères.
+Prop :Chaine de caractères.
+Resultat : Chaine de caractères.
Table_service web_toler :Table
+Id_service web :nombre entier.
+Prop :Chaine de caractéres.
+service web :Chaine de caractéres.
+ offre:reel.
+demande:reel.
+se: nombre donné plus petit. Soit &

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

```
(Public) Procedure
Select_List_Peer_alive_Disp()
(Public) Procedure Receive_resultat()
(Public) Procedure
Receive_Confirmation()
(Public) Procedure charge_cour()
(Public) Procedure Clear_annuaire()
(Public) Procedure Update_SND()
(Public) Procedure Update_annuaire()
(Public) Procedure Detect_Fault()
(Public) Procedure Prise_decision()
(Public) Procedure
Determinate_NbDuplic()
(Public) Procedure Peer_Crash_ListSW()
(Public) Procedure List_SW_Tolerer()
(Public) Procedure Toler_Fault()
(Public) Procedure New_SND()
(Public) Procedure Toler_Fault_Peer()
(Public) Procedure Toler_Fault_SND()
 (Public)Procedure Trans_Tache()
```

C. SuperNodeDupliqued

is a peer group, furthermore he is responsible for monitoring the SuperNode, if he fails, he takes over, becoming the SuperNode group in turn means it among a group of peers SuperNodeDupliqued and former SuperNode on his return, he will be an ordinary peer group, and connect with other super node duplicated from other group.

➯ **Note:** The model SGRTE manage the grid simultaneously and in parallel from two Web Services Management (SGRTF & SGREC) and the following points will be assessed:

-high availability load balancing mechanism;

- Supported services (network services TCP / IP, etc.);

- Resources monitored and used in the criteria for dispatching (CPU usage, load, number of connections)

- Detect down nodes.

- Mechanisms supported (NAT, tunneling, direct routing).

3.2. EXPERIMENTS

We have identified the difficulty of fault tolerance and load balancing in the grid, it becomes necessary to control the errors to assess the mechanisms in place to cope. We have experienced our services load balancing and fault tolerance system **(SGRTE)** to evaluate the performance of our application. So we will evaluate for each number of programs (services) the level of fault tolerance and load balancing (outside the group / inside the group).
so we will evaluate for each number of programs (services) the level of fault tolerance and load balancing (outside the group / within the group).

Table 1 . Test results for the over cost of the components of group rate overload

#pgms	surcout	surcout snd	surcout/peer (node de calcul)	surcout peers	sn/peers	snd/sn	snd/peers
2	3411	1347	516	2064	15,13%	39,41%	38,31%
8	6822	2694	1o32	4128	14,66%	39,43%	38,21%
10	1023	4041	1548	6192	15,23%	39,45%	38,13%
15	1364	5388	2064	8256	15,10%	39,46%	38,32%
20	1705	6735	2580	10320	15,09%	39,49%	38,33%

Table 1 . test results for the over cost of the components of group rate overload (forwarded messages) is 15% between a peer group and SN, 39.49% between the SN and SND , 38, 31% between the SND and a peer group. we can say that the load of SN is 75% greater that a peer group and 60% compared to SND.

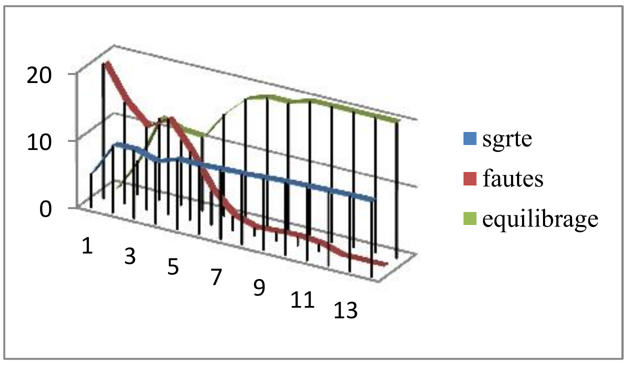

Figure 3. Result of dependency between faults and load balancing group.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

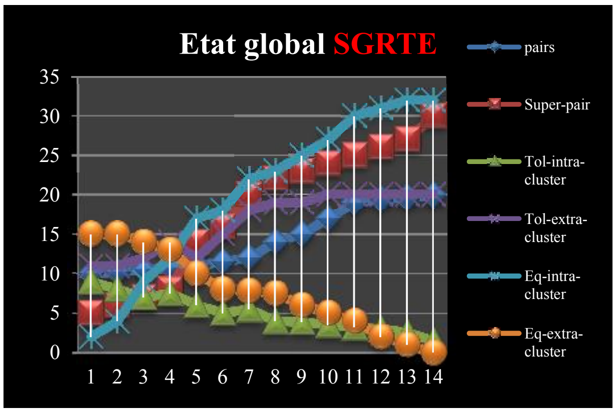

Figure 4. Results of the dependence between the number of peers and fault tolerance and load balancing on the outside and inside the group.

4. CONCLUSION

The coupling of models and P2P grid computing has made birth of a new grid model, which is P2P-Grid; this model, as it has its advantages of these two components, it inherits the disadvantages and the other appeared coupling.

In this paper, we have designed, implemented a service fault tolerance and load balancing strategy in a P2P grid model. The proposed P2P grid model can transform into a grid interconnected groups, each group consists of a SuperNode, a SuperNodeDupliqued, and peers. We implement a model ***SGRTE*** that can manage in a transparent fault tolerance and load balancing in the grid, the model reached a certain maturity, namely: the widespread use today of "farms servers' progress around applications called "online" (Web 2.0 and 3.0, ..), and finally as our future work is used to model ***SGRTE*** the spread of the Internet and the development of broadband networks, in this case it is the whole setup, including material that is leased to the request: server (s), storage, operating system, etc., can be managed by a cloud (cloud computing) applications such as Google , social networks ***Facebook, youTube and Twitter.***

REFERENCES

[1] I. FOSTER, and A. IAMNITCHI. *On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing. 2nd International Workshop on Peer-to-Peer Systems (IPTPS03)*, pages 118–128, 2003.
[2] Gabriel Antoniu, Mathieu Jan, David A. Noblet, *A practical example of convergence of P2P and grid computing: an evaluation of JXTA's communication performance on grid networking infrastructures* , September 2005.
[3] Laprie, J.-C., ed. *Le guide de la sureté de fonctionnement*. 1996, Cepadues.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

[4] Sara Bouchenak, Sacha Krakowiak, Noël de Palma *Tolérance aux fautes dans les grappes d'applications Internet*, INRIA, Projet Sardes – Montbonnot St Martin, 38334 St Ismier Cedex, France 8 avril 2005.
[5] François Taïani, Marc-OlivierKillijian, Jean-harles Fabre, *Intergiciels pour la tolérance aux fautes* ; Computing Department, Lancaster University LANCASTER LA1 4WA, Grande Bretagne, 2006.
[6]Distributed Load Balancing Model for Grid Computing Belabbas Yagoubi ,Meriem Meddeber , University of Oran Algeria Revue ARIMA, vol. 12 (2010), pp. 43-60 .

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)