

Condor Cloud New Solutions

Ali Hameed Yassir Ali Fattah Dakhil
College of Computer Science and
Information Technology
Sumer University
alihameed_48@yahoo.com allee.fattah@gmail.com

Wafaa Mohammed Ali
College of Education for
pure sciences
University Thiqar
wafaam20142015@gmail.com

Received : 16\12\2017

Revised : //

Accepted : 8\1\2018

Available online : 20/2/2018

DOI: 10.29304/jqcm.2018.10.2.369

Abstract:

In this article author proposes a new solution in the field of distributed computing needs, humans' requirements of new ways for developing technologies that increase information systems distribution horizontally and decrease the power consumption and delay time vertically. The author presents in detail the distributed computing concepts, types and some examples like Condor system of distribution and the proposal for developing.

Keywords: condor, solution, distributed computing, power, open source, rmi, cloud.

1- Introduction

The earliest human ate food gathered by them using only the tools that they made themselves. We have continued the efforts made by humans for the disengagement between production and consumption for thousands of years and eventually led to the emergence of intensive agriculture, manufacturing, and distribution of electrical energy. As a consequence, it is the privatization of labor and economy and new technologies as determined by the modern world that give me the freedom to sit in a cafe drink a cup of coffee and write this article on my laptop, without drawing the attention to the sources of the things that I consume or use, including coffee and water, electricity and bandwidth wireless network.

This is the availability and accessibility of these sources to be fit, for example, on the concept of default virtualization, which indicates (in computer science) to withhold some useful functions behind the interface and hides the details about how to put these functions into effect. For example, when a waiter opens the cafe tap water, it's akin to scooping water from the barrel inexhaustibly. The same phenomenon occurs when I connect my laptop power plug in the wall, the presence of a huge electric switch invisible behind the plug, who knows when and how this electric energy was is generated. The computing itself is not characterized by the default character. Laptop, desktop or even the sophisticated computer centers of information for one of the companies, are systems with a great deal of independence and sufficiently implement programs stored

Locally. Since we do not accept a situation where each institution or a commercial company is obliged to run the power station, so why accept it for computers. This situation pushed the computer scientists to look for other alternatives more effectively. Digital networks increase speedily day after day [1], why do not we 'compile' computers on an ongoing basis so that users can request resources, storage, data and software regardless of location and appropriate source to provide. In other words, why not make public virtual computer services? Such a 'networking' computation would be beneficial and useful like electric networking. This networking can work like institutions of e-commerce, for example, Virtual tour of tourist attractions, so it can be adapted to their

information systems and their computers depending on the demand as it works at the same time to connect these systems with partners, suppliers and customers in order to expand the network of services they provide. For example Adventure Travel Company provides to its customers trips to explore the virtual reality of its existing sites under the sea. Once the potential customers start sailing in the virtual water outside the tourist resort (for example Tioman island), the company is obliged to access to some databases and software required to retrieve geographic information related to the trip, and convert the data, write the appropriate information and integrate it all in real time video pictures coming from cameras that have been installed in the same location [1], [2]. As it is impossible for any travel company to run this potential alone, but with the available networking technologies, it could assemble the necessary resources from several sources, each one of them to meet the needs of a large number of customers, and achieve economic feasibility. In addition, the systems networking carry great potential for a number of other areas where work is limited to a few categories. In the field of medicine, for example, it allows access to computers and remote databases to a doctor comparing the radiographic images of millions of other patients, which may be able, for example, to discover a malignant tumor early. It may allow biological chemists at different places to screen 10,000 candidate for drugs with hours rather than days, or may allow civil engineers to test and design earthquake-resistant bridge within a few days instead of several months [4].

This concept for the Distributed computing and networking is a natural extension of the current web. Internet has added advantage on the nature of communication, allowing any person to call someone or one device to call another device regardless of location or the means used to do so. This has resulted from the innovation of e-mail, World Wide Web applications and peer peer-to-peer, including systems for file-sharing that provide access to any person or another device, and so this is not inferior to other groups varying from those services and is safe and reliable.

2- Distributed Computing

Distributed Computing is a field of computer engineering and studying distributed systems. Distributed systems are composed of a group of computers that communicate with each other through a network connected with each other. [3] Computers interact with each other to achieve a common goal called computer programs that run on distributed systems. There are two types of distributed computing, [8] cluster and grid computing besides two other common types, distributed information system and pervasive distributed computing. In distributed computing, the problem is divided into small problems, and then is distributed to other computers for processing. Figure 1 represents grid computing [3], [4].

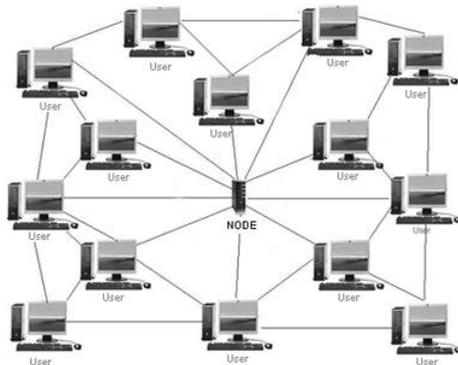


Fig. 1. Grid Computing

The idea behind distributed computing often does not use all the processing power available. In normal circumstances, [5] using certain number of program requires a certain amount of the processing power and a large amount of processing remains unused. [6] The idea behind distributed computing is to use the idle power available in some way. Simply linking to a certain number of computers in a grid needs the presence of a particular program on each machine within the network and some of the tasks are distributed on devices and not used by the main user.

The processors of these devices in turn address the jobs sent to it through this network by a user of the master device [5], [6] and [9]. Distributed systems characterized many advantages, including:

1. There are no common clock pulses between devices; each device has an independent clock pulse from other devices.

2. There is no shared memory; information is exchanged by passing messages.
3. The ability to withstand failure tolerates failed.
4. The ability to expand in the structure of the system (the network, the number of devices).
5. Each device or node sees a fraction of the system, and deal with a specific part of the income (a specific type of data).

3- Condor is a Distributed Computing

Condor is a computing infrastructure that utilizes unused compute cycles from common workstations was developed by the University of Wisconsin-Madison. This system called Condor has been instrumental in the development and testing of physics-based algorithms requiring massive computations.

The initial use of the Condor was to produce atmospheric databases through a parameter sweep of key inputs into the MODTRAN atmospheric propagation code. These databases were initially used as part of an atmospheric compensation algorithm optimizing for atmospheric visibility, [8] water vapor content, and target surface elevation served in a model predicted vs. real sensor reaching radiance comparison [8]. These selected atmospheric inputs were used as the basis for converting a sensor reaching radiance scene to surface spectral reflectance. The same type of database has been used as part of a target detection scheme where a target spectrum is propagated through a range of atmospheric parameters. This results in a set of estimates of how a sensor would image this target through various atmospheric conditions which generates a subspace on which target detection schemes can be applied. Because it is possible to finely sample the solution space of numerical models such as MODTRAN, statistical descriptions of these atmospheric conditions can be devised as a more compact description of an entire. Condor runs on Linux, UNIX, Mac OS X, FreeBSD, and contemporary Windows operating systems [7]. Condor can seamlessly integrate both dedicated resources (rack-mounted clusters) and non-dedicated desktop machines (cycle scavenging) into one computing environment.

Condor is developed by the Condor team at the University of Wisconsin–Madison and is freely available for use. Condor is an open source (it's licensed under the Apache License 2.0). It is also available on other platforms, like Ubuntu from the repositories. Some applications like the NASA Advanced Supercomputing facility (NAS) Condor pool consists of approximately 350 SGI and Sun workstations purchased and used for software development, visualization, email, document preparation, etc. Each workstation runs a daemon that watches user I/O and CPU load. Condor scheduler performs network admission control to ensure that network resources are not oversubscribed. The scheduler allocates network capacity up to a configured limit for each subnet. The administrator typically configures this limit to less than the subnet's full capacity to reserve capacity for other network users. The scheduler places jobs on remote CPUs only when the network capacity limits would not be exceeded by the jobs' checkpoint and data transfers. If capacity would be exceeded for a given subnet, jobs which may have run on CPUs on that subnet will instead run in other subnets (where network capacity is available) or remain idle.

When a workstation has been idle for two hours, a job from the batch queue is assigned to the workstation and will run until the daemon detects a keystroke, mouse motion, or high non-Condor CPU usage [9]. At that point, the job will be removed from the workstation and placed back on the batch queue. Condor can run both sequential and parallel jobs. Sequential jobs can be run in several different "universes", including "vanilla" which provides the ability to run most "batch ready" programs, and "standard universe" in which the target application is re-linked with the Condor I/O library which provides for remote job I/O and job checkpointing. Condor also provides a "local universe" which allows jobs to run on the "submit host". In the world of parallel jobs, Condor supports the standard MPI and PVM. In addition to its own Master Worker "MW" library for extremely parallel tasks. Condor-G allows Condor jobs to use resources not under its direct control. It is mostly used to talk to Grid and Cloud resources (Cloud computing and grid computing are the words that confuse a lot of people because of the similarity in theory. Cloud computing and grid computing infrastructure contain a huge computer

network [10]. On the front end, cloud computing and grid computing concepts are newest compared by many other computing. Both concepts have been developed in order to distribute computing as the example, the computing element over a wide area. Literally, [11] the computers are separated from each other or by other means, In another word, there are a lot of people who prefer to use distributed computing on a single processor computing, and here are the reasons:

1. Distributed computing are sources that provide simultaneous or parallel computing for users by decreasing the delay time. Applications do not need to wait in a queue to be served one after the other.
2. The cost for Distributed computing is also not high since it is better to have several low-cost devices rather than one high-cost computer.
3. Distributed computing is much easier to grow or increase capacity; designers can add other computers in the event of the increased number of users.

Undistributed computing needs to increase the computing power in the system which will require a high cost, To understand the basics and details between grid computing and cloud, It should to explain both techniques, "Cloud Computing" Cloud is basically an extension of the concept of abstraction code, [6] An extension of the object-oriented programming concept of abstraction, it means the Internet cloud. End-users are only given products or input data for the whole process, which leads to the outputs which are simply hidden and shown on computing and fake sources that are present in multiple servers or the servers that work in groups. Also in the family "cloud computing" There is what is known as "software as a service" in English "Software as a Service" or SaaS applications [3]. These customers' clients always install light programs on the server but leaving all the heavy work using the infrastructure to someone else. Cloud computing eliminates the cost, and the complexities of purchase, configuration, and management of the hardware and software required to build and deploy applications: These applications come as a service on the Internet (cloud). "Grid computing" Networking systems are designed for cooperation in post sources and can also be perceived as a distributed computing on a large scale and is mainly used by the processing capacity in several PCs to handle one task [6].

The task is divided into several sub-tasks; each device in the network client has a sub-task. Once tasks are processed, they are sent back to the main system which supervises all the tasks and is grouped together or united as directed.

4- The Open Source Condor for Cloud

Red Hat Expanded Messaging, Real-time, and Grid Technology Capabilities to advance cloud it announced the availability of Red Hat Enterprise MRG 1.3, [16] including updates to the product's Messaging, Real-time and Grid technologies, which provide a key technology base for Red Hat Cloud Foundations, a solution set that offers a comprehensive set of tools to build and manage a private cloud Figure 2. Red Hat Enterprise MRG provides an integrated platform for high performance distributed computing.

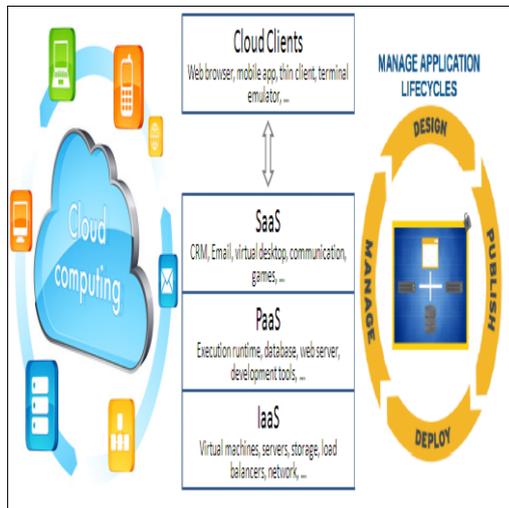


Fig. 2. Red Hat Enterprise MRG.

Enterprise MRG first released was in June 2008, Enterprise MRG has since enabled customers around the world to meet their messaging, real-time and grid computing needs, offering: Enterprise MRG's Grid functionality, based on the Condor Project brings the advantages of flexible deployment to a wide range of customers. By using software methodology for design and implement an evaluation let the developers using PaaS (Platform as a Service) to modify the functions or task, the network architects IaaS (Infrastructure as a Service) to get the scalability to provide the high-performance services to the End users SaaS (Software as a Service).

5- Applications and Workloads

With Grid, customers can build cloud infrastructures to aggregate multiple clouds. It provides integrated support for virtualization and public clouds and easier aggregation of multiple cloud resources into one compute pool. In addition, [11], [14] it provides more streamlined and flexible computing across remote grids with servers, clusters and cycle-harvesting from desktop PCs as well as across private, public and hybrid clouds. MRG Grid is a key base component of Red Hat Cloud Foundations see Figure 3.

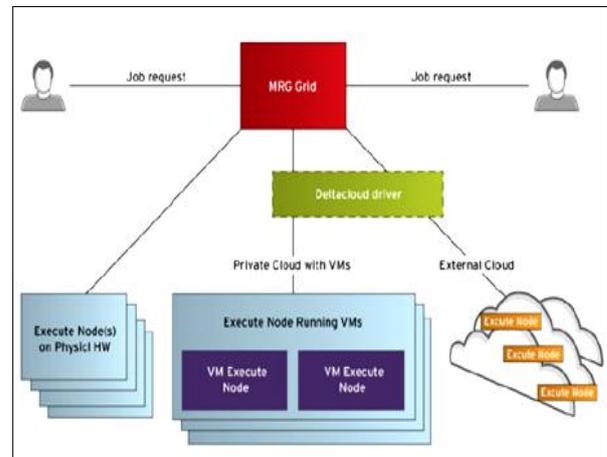


Fig. 3. VM and Cloud provisioning with delta cloud.

6- Condor RMI for Overlay Resource Manager

Ten years ago condor has introduced the concept of Condor glide-ins as a tool to support 'just in time scheduling' in a distributed computing infrastructure that consists of resources that are managed by (heterogeneous) [12] autonomous resource managers as example glideinWMS Figure 4. By dynamically deploying a distributed resource manager on resources allocated (provisioned) by the local resource managers, the overlay resource manager can implement a unified resource allocation policy. In other words, the use remote job invocation to get resources allocated.

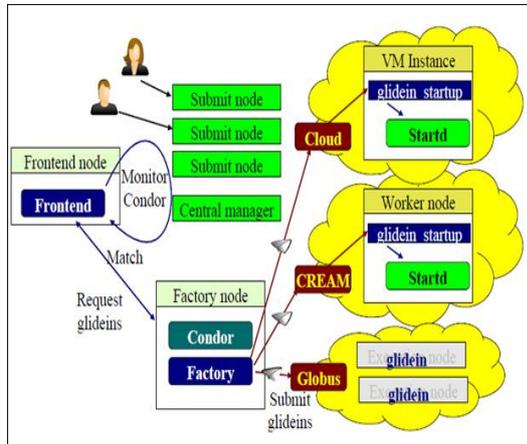


Fig. 4. glideinWMS Architecture.

7- Conclusion, Solutions and Suggestions

Cloud services are provided by grid computing. In fact, almost all the online services that users get are from the cloud, for example, web hosting, multiple operating systems, and database support. It tends to have more leniencies in conjunction with heterogeneous and geographically scattered computing systems; Condor is a powerful tool for scheduling jobs across platforms, both within and beyond the boundaries of clusters [13]. Through its unique combination of both dedicated and opportunistic scheduling, Condor provides a unified framework for high-throughput computing. By integrating the basic Condor commands into a GUI-based program, is to make Condor become more user-friendly and more convenient to use. An integration of more advanced Condor commands into the GUI-based program, developed and making Condor serve as the key schedule of a large grid computing system based on WSRF.NET[15]. Distributed computing success depends on high numbers of its adopters. Among the proven strategies to overcome the traditional position baffling is to secure free access to basic technologies. From here, it is important not only to have the availability of all the major specifications of communication free-for-all but also to have an open and easy way to implement these specifications. The Globus toolkit meets these requirements condor is one of these toolkits. However, any further progress depends on the contributions of academics, industrialists, and sellers; it also depends on the inclusion of the concepts of communication within the business software as well as the training of the other parties.

References

1. Douglas Thain, Todd Tannenbaum, and Miron Livny, "Distributed Computing in Practice: The Condor Experience" *Concurrency and Computation: Practice and Experience*, Vol. 17, No. 2-4, pp. 323-356, February-April, 2005.
2. Alain Roy and Miron Livny, "Condor and Preemptive Resume Scheduling", Published in *Grid Resource Management: State of the Art and Future Trends*, pp. 135-144, Fall 2003.
3. Rajesh Raman, Miron Livny, and Marvin Solomon, "Policy Driven Heterogeneous Resource Co-Allocation with Gangmatching", *Proceedings of the Twelfth IEEE International Symposium on High- Performance Distributed Computing*, June, 2003, Seattle, WA.
4. Nicholas Coleman, Rajesh Raman, Miron Livny and Marvin Solomon, "Distributed Policy Management and Comprehension with Classified Advertisements", *University of Wisconsin-Madison Computer Sciences Technical Report #1481*, April 2003.
5. Nicholas Coleman, "An Implementation of Matchmaking Analysis in Condor", *Masters' Project report*, University of Wisconsin, Madison, May 2001.
6. Condor Team, University of Wisconsin-Madison "Condor Version 7.3.1 Manual", pp.29-129, May 20, 2009.
7. J. Basney and M. Livny. Improving goodput by co-scheduling CPU and network capacity. *International Journal of High Performance Computing Applications*, 13(3), Fall 1999.
8. J. Bester, I. Foster, C. Kesselman, J. Tedesco, and S. Tuecke. A distributed resource management architecture that supports advance reservations and co-allocation. In *International Workshop on Quality of Service*, 1999.
9. C. Lee, R. Wolski, I. Foster, C. Kesselman, and J. Stepanek. A network performance tool for grid computations. In *Proceedings of the Conference on Supercomputing*, 1999. High-throughput resource management. In I. Foster and C. Kesselman, editors, *The Grid: Blueprint for a New Computing Infrastructure*, chapter 13. Morgan Kaufmann Publishers, Inc., 1998.

10. R. Raman, M. Livny, and M. Solomon. Gang-matching: Advanced resource management through multilateral matchmaking. In Proceedings of the Ninth IEEE International Symposium on High Performance Distributed Computing, August 2000.
11. S. Vinoski. Advanced message queuing protocol. IEEE Internet Computing, 10:87–89, November 2006.
12. E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, “The cost of doing science on the cloud: The montage example,” in SC08, 2008, pp. 1–12.
13. D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. P. Anderson, “Cost-benefit analysis of cloud computing versus desktop grids,” in IPDPS, 2009, pp. 1–12.
14. D.Thain, T. Tannenbaum, and M. Livny, “Distributed computing in practice: the Condor experience.” Concurrency - Practice and Experience, vol. 17, no. 2-4, pp. 323–356, 2005.
15. “Sungridengine,http://gridengine.sunsource.net/RedH.

حلول جديدة سحابة الكوندور

وفاء محمد علي

كلية التربية للعلوم الصرفة

جامعة ذي قار، ذي قار، العراق

wafaam20142015@gmail.com

علي فتاح داخل

كلية علوم الحاسوب وتكنولوجيا المعلومات

جامعة سومر، ذي قار، العراق

alihameed_48@yahoo.com

المستخلص :

في هذه المقالة يقترح المؤلف حلا جديدا للاحتياجات في الحوسبة الموزعة، المتطلبات البشرية لطرق جديدة لتطوير التكنولوجيات التي تزيد من توزيع نظم المعلومات أفقيا وتقليل استهلاك الطاقة والتأخير في الوقت عموديا. يعرض المؤلف بالتفصيل مفاهيم الحوسبة الموزعة وأنواعها ويعرض الحلول مع الأمثلة مثل نظام سحابة كوندور الموزعة كاقترح للتطوير.