



Beowulf Cloud: A Scalable Cloud Computing Architecture

Vishal Garg

Department of IT, Government of NCT of Delhi, Delhi Secretariat, New Delhi, India

Abstract

Cloud based infrastructure is much popular in IT industry and most of the IT companies and organizations have already migrated or migrating to the cloud either public, private or hybrid. Most of the big organizations deploy their own private cloud, the medium sized organizations migrate to hybrid clouds which is a combination of private and public clouds and the smaller organizations migrate to public clouds. But, for open source organizations and other small scale or non-profit organizations or projects who cannot afford to have a cloud based infrastructure and need large amount of computation power to serve to large number of end users with less or no profit, cloud based infrastructure is a big challenge. Most of these projects either sold out or fails. Beowulf [2] cloud can be the solution to them as the computation power can be donated and shared between computers, clusters [4] or clouds from anywhere on the network.

Keywords: Cloud computing; Container data type; Computer cluster; Beowulf cloud.

Introduction

The Beowulf cloud is a technique to donate a fixed amount of computation power as a service. The organization which cannot afford to have online infrastructure like a cloud based infrastructure running, may ask for people to connect and donate their computation power. These days, containers (abstract data type) [3] are already very popular with clouds and much mature. The computer systems are having a good number of resources already and with the use of containers, we can create a cloud which can be run from worldwide. This kind of computing architecture is not completely distributed but a kind of cloud architecture that can be scaled and used for heavy loads. The organization building a cloud like this may ask people around the world to donate their part of computation power. Those who want to donate their computation power will install a utility program or a software control panel to donate their computation power. Using that software program, the donors can donate a fixed or variable amount of computation power like number of threads or number of cores or percentage of computation power to cloud. The software program will automatically download the containers according to their computation power and will execute those containers. The Beowulf Cloud will be having an additional container cluster controller for the distribution of containers and to connect with the donor nodes. For a cloud required for research work or special kind of tasks like graphics processing, the special kind of hardware can also be used and donate their amount of computation power like graphics cards for graphics processing and mathematical computational tasks.

Detailed Description

Requirements. Beowulf cloud can be implemented with any existing cloud, there is one extra node is required which will be the container cluster controller. A utility program for the donor machine to download and manage containers locally.

Beowulf Cloud Implementation. The Beowulf cloud is based on containers to share and manage computation power. We'll need to choose which containers to use. There are lot of options available as there are lots of sophisticated containers available including commercial and open source. The container cluster

controller needs to be implemented so a separate node or number of nodes are required according to the load. The container cluster controller will be able to manage all the tasks related to distribution of containers to donor nodes. The container cluster controller can be complex because the donor nodes cannot be same as the ordinary compute node. There are lot of restrictions and limitations for that including but not limited to the amount of RAM available for the containers, amount of computation power available, security of the containers, connectivity of node, availability of node. Every limitation is the task to be managed on container cluster controller and utility program for donor node.

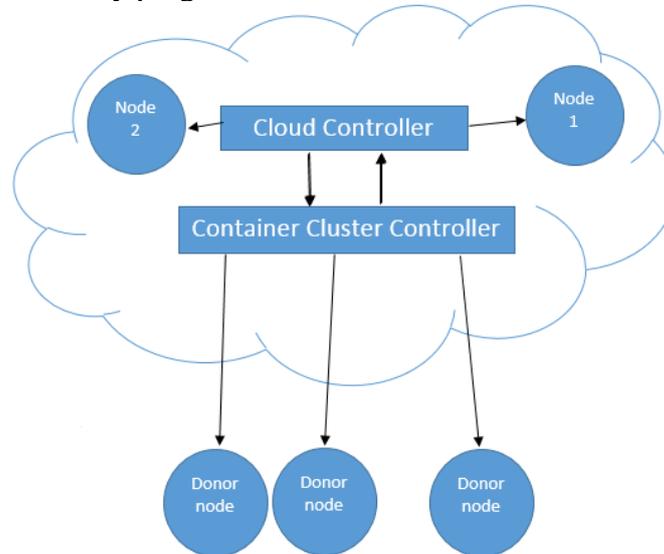


Fig. 1 Beowulf Cloud Architecture

The container cluster controller should create containers and those containers need to be as available containers. As there can be a number of compute nodes also available with the cloud which are managed by the cloud controller, the work will be distributed between the cloud nodes and the donor nodes with the help of container cluster controller and cloud controller. The container cluster controller will communicate with the cloud controller for containers available to execute. Those containers can be transferred to the donor nodes from container cluster controller. The choice of donor node to transfer the containers can be in various ways like based on priority i.e. the node with higher reliability, availability and scalability [1] will be having higher priority and the containers will be served to those high priority nodes first. The node will respond with the result of execution as soon as the container executes and container cluster controller will wait for specific time for the node to respond. After the timeout of the response of execution of container the container cluster controller will retransmit the same container to different node to execute. When the container cluster controller receives the response with container marked as success then that container will not be out of the execution queue and a new container will be transmitted to that node. To speed up the execution, the multiple containers can be transferred to a node. The donor nodes which have been dedicated to cloud can manually be marked as higher priority on the container cluster controller. The nodes with very poor reliability can be marked the lowest priority to make the Beowulf cloud better.

Donor Nodes

The donor node is the computation node for the Beowulf cloud on which the Beowulf cloud is based on. The donor node can be located anywhere on the network and can be having any configuration of hardware. A utility software is required to be built for the management of the node and to communicate with the cloud. The donor node having utility software should have the options to donate the amount of computation power

like in terms of threads or in terms of percentage or in terms of number of CPU cores. There can be other options to connect or disconnect the node from cloud. The utility program on donor node will download the containers according to the amount of computation power is donated and those containers will execute with

only the amount of computation power specified in the utility program. The response will be sent to the server with the success if, there is the connectivity issues the utility program will wait for a specific time and after that the container will be destroyed without sending the response. The utility program will also look for the tapping of containers for security also. The donor node can also be idle if there is no container available to execute and at the peak time, there will be the continuous execution of containers.

For the specific research or graphical rendering or mathematical computation applications, the specific kind of hardware can be used at the donor node like a graphics card. The utility program will be able to manage the hardware resources that can be donated. The graphics or sound cards can perform the specific task and the containers will also be communicated according to their work. A Beowulf cloud can be a specific or general kind of cloud in terms of kind of task it can perform.

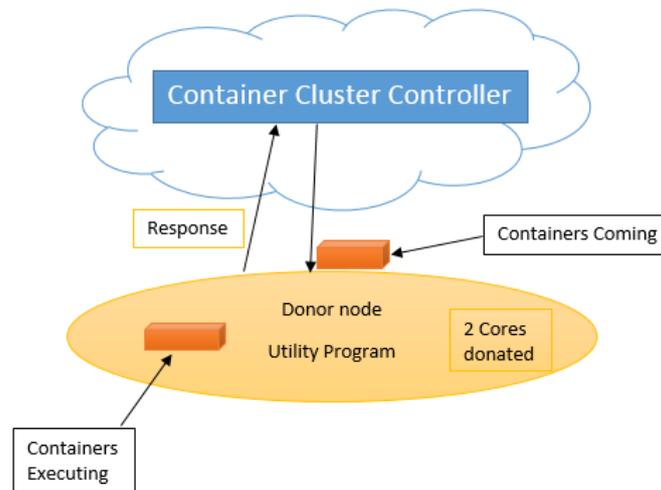


Fig. 2 Beowulf Cloud Data Flow

The utility program can manage and download the multiple containers to make the communication reliable and to continuously process the containers. To make the donor computer secure, the utility program will be restricted to few kind of resources of the donor computer. It is better if, there is a dedicated user created on the donor machine so that the user can be sure that his personal files are secure.

If, the organization running the cloud is paying donor nodes for the computation power then, the utility program must have a procedure to count the money against the computation power or the number of containers executed. There can be the local database as well as the container cluster controller will also manage the money against the computation power donated. This can be opposite to the cloud services we are having now, the cloud vendors charge for the services the users get and in Beowulf cloud they can give money for the services they have received. There are tools available in existing clouds to calculate the money against the services and these can be modified to be used with Beowulf cloud to pay donor nodes for their services.

Security

The container cluster controller will communicate with donor nodes and will also manage the security of communication and containers. The container should be secure enough to hide the data which should not be tapped. The secure protocols can be used to communicate between container cluster controller and the donor node. If, containers are only having the computational instructions and there is no related data processing which is confidential then the cloud can be secure and the data processing can be done on local or specific nodes.

The authentication procedures can be used like existing clouds to authenticate. The authentication can be done on a separate module or can be integrated with container cluster controller. The billing for paid donors can be done on the separate module or can be integrated on the container cluster controller but, it is best practice to separate the modules to improve performance and manage easily.

Future Trends

The Beowulf Cloud technique may be used in wide range of applications in the future and as seen already. Likewise in different shared processing applications, the technique of sharing processing power is already in use. With the use of the sharing of computation power, there can be the mobile cloud based infrastructure also in the near future. The hardware resources are getting much more powerful either in desktop workstations or mobile devices, the processing power can be shared everywhere in the near future with this technique to make possible the virtual infrastructure.

There can be the virtual parallel systems also like virtual supercomputers, which can be built using Beowulf Cloud technique without the requirements of huge resources like cost, time, hardware or human. There can also be the service opposite of the service we are currently running that is computation as a service which can be served by public to organization not organization to public.

Conclusion

Cloud computing is becoming so popular and with the popularity and trend, every organization is adopting the cloud architecture. The small organizations are migrating to public clouds, the medium sized organizations are migrating to hybrid cloud and large organizations are migrating to private clouds. The micro or open source organizations who want to migrate to private cloud and can't afford it, can try to build a Beowulf cloud using containers to adopt new technology and great scalability. The Beowulf cloud can be the hot trend and successful architecture in small organizations as well as large organizations.

References

- [1] Hill, Mark D. (1990). "What is scalability?". ACM SIGARCH Computer Architecture News 18. doi:10.1145/121973.121975. and Duboc, Leticia; Rosenblum, David S.; Wicks, Tony (2006). A framework for modelling and analysis of software systems scalability. Proceeding of the 28th international conference on Software engineering - ICSE '06. pp. 949. doi:10.1145/1134285.1134460. ISBN 1595933751
- [2] Becker, Donald J and Sterling, Thomas and Savarese, Daniel and Dorband, John E and Ranawak, Udaya A and Packer, Charles V, "BEOWULF: A parallel workstation for scientific computation", in Proceedings, International Conference on Parallel Processing vol. 95, (1995)
- [3] Paul E. Black (ed.), entry for data structure in Dictionary of Algorithms and Data Structures. US National Institute of Standards and Technology. 15 December 2004.
- [4] Bader, David; Robert Pennington (June 1996). "Cluster Computing: Applications". Georgia Tech College of Computing.