IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 11, November 2019

Evolution of Scale-Out Storage Compare to Scale-Up Storage

Punam Jagannath Desai

M. Tech Student, Department of Computer Engineering, Bharati Vidyapeeth College of Engineering, Pune, India.

Abstract: Infrastructure scalability handles the changing needs of an application by statically adding or removing resources to meet changing application demands as needed. In most cases this is handled by scaling up or scaling out. There have been many areas of how it works and architecting for emerging cloud native applications. Scale up is done by adding more resources to an existing system to reach a desired state of performance. More compute, memory, storage or network can be added to that system to keep the performance at desired levels. Scaling-up can also be done in software by adding more threads, more connections or in case of database applications, increasing cache size. Scale-out is a Network-Attached Storage (NAS) architecture in which the total amount of disk space can be expanded through the addition of devices in connected with their own resources.

In cloud storage deployers deal with multiple parameters like scalability, cost, throughput, reliability, latency and resource utilization. This paper examines how scale out storage is more efficient than scale up with respect to some of these parameters.

Infrastructure scalability handles the changing needs of an application by statically adding or removing resources to meet changing application demands as needed. In most cases this is handled by scaling up or scaling out. There have been many areas of how it works and architecting for emerging cloud native applications. Scale up is done by adding more resources to an existing system to reach a desired state of performance. More compute, memory, storage or network can be added to that system to keep the performance at desired levels. Scaling-up can also be done in software by adding more threads, more connections or in case of database applications, increasing cache size. Scale-out is a Network-Attached Storage (NAS) architecture in which the total amount of disk space can be expanded through the addition of devices in connected with their own resources. In cloud storage deployers deal with multiple parameters like scalability, cost, throughput, reliability, latency and resource utilization. This paper examines how scale out storage is more efficient than scale up with respect to some of these parameters.

Keywords: Cloud, Resources, Storage, Throughput, Virtualization

I. INTRODUCTION

Storage capacity is the most fundamental criteria of a storage device. The second most fundamental storage criteria is what happens when you need to add capacity. The prospect of adding capacity keeps storage managers up at night. When we run out of capacity, there are two options. These options may be as simple as adding a new shelf of drives, or as complex as building a new data center. Obviously, simpler is better. And the key architectural difference that drives simplicity is whether your system design is scale-up or scale-out.

“Scale-up” architecture has been a long-running standard for storage. However, as data volume grows, it becomes increasingly difficult to ignore the many limitations and flaws of scale-up storage. The solution may be “scale-out” architecture.

Scale-up is the most common form of traditional block and file storage platforms. The system consists of a pair of controllers and multiple shelves of drives. When we run out of space, we add another shelf of drives. Scale-up architecture is limited to the scalability limits of the storage controllers.

Once the performance and/or capacity limits of the storage controllers are reached, then the only option is to add a new system to sit alongside the existing one. At this point, workload grows as we migrate storage and manage the load between the two independent silos of storage.

As an organization’s data volume grows, completely new systems need to be added to cope with the additional demands. And this is called scale out storage where total amount of disk space can be expanded through the addition of devices in connected with their own resources.

Hence we can say that Scale-up machines perform better for jobs with small and median (KB, MB) data sizes, while scale-out machines perform better for jobs with large (GB, TB) data size.

A major benefit to these emerging scale out solutions is that the cost of upgrades is generally low; additional nodes are generally the same price as the first nodes, and there is very little operational impact to adding nodes. This means that it is sensible to start with enough capacity and performance for current demand plus a few months of growth, buying new nodes as consumption increases. This also means that it is quick and fairly easy to add capacity to cope with the large new user count for a new project, reorganization or company acquisition.

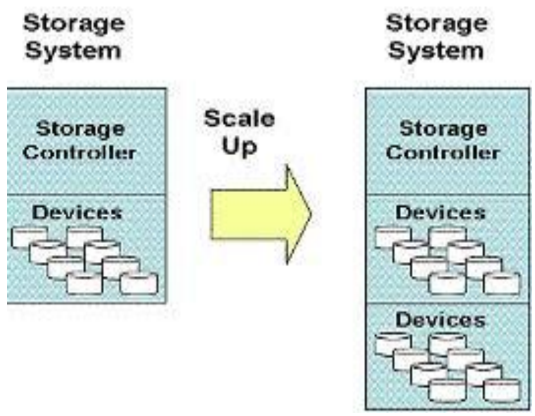
Another benefit of scale out is considering the parameter resource utilization. We can use different techniques like scheduling of resources or virtualization in scale-out system to improve performance of system and so that it becomes more useful than the scale-up storage.

Storage virtualization is a technology to create logical abstraction that isolates application from physical storage systems. Storage virtualization enables storage system to overcome physical barriers between resource and thereby optimize resource utilization and increase flexibility.

Scalability is another parameter that affect on the performance of overall system. While evolution from scale up to scale out this parameter must be considered. As the client to the object storage system addresses a virtual FQDN address, it is simple to add further nodes to the cluster, the object storage software including the new nodes into the cluster and redistributing the data across these nodes. For every new node added, it adds not only additional disk storage capacity (and performance) but additional RAM, CPU, and networking resources to improve overall performance of the cluster. This method of scaling avoids the limitations of the traditional scale-up storage architecture where every IO has to be processed by one of only two storage controllers.

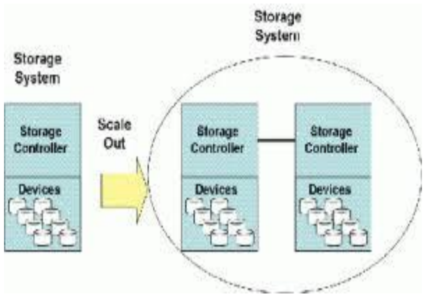
II. RELATED WORK

A. What is Scale-Up Storage?



Scale-up is the most common form of traditional block and file storage platforms. The system consists of a pair of controllers and multiple shelves of drives. When we run out of space, we add another shelf of drives. Scale-up architecture is limited to the scalability limits of the storage controllers. At first, adding more disks improves performance because disk throughput was probably the limiting factor from a performance standpoint. However, as the load on the array climbed — a situation often driven by virtualization — and more disks were added, the two controllers themselves became a bottleneck as each began to require more and more CPU for RAID calculations. Eventually, enough disks were added that the controllers were simply saturated and could do no more. Adding more and faster disks behind an overloaded controller pair simply places more overload on the controllers. In these systems once the controllers are the bottleneck there is little we can do, apart from buying an additional new array with its own pair of controllers and moving some of the workload (VMs) onto the new array.

B. What is Scale-Out Storage?



Scale-out is a Network-Attached Storage (NAS) architecture in which the total amount of disk space can be expanded through the addition of devices in connected with their own resources. When a scale out storage reaches to its storage limit, another array can be added to expand the system capacity.

In a scale out system new hardware can be added and configured as the need arises. When a scale-out system reaches its storage limit, another array can be added to expand the system capacity. Scale-out storage can harness the extra storage added across arrays and also use added devices to increase network storage capacity, adding performance and addressing the need for additional storage. In the diagram above, the scale-out system illustrated only includes a single added cluster. In reality, clusters can be added almost without limits as requirements dictate. Each device (node) includes storage capacity, which may be in the form of multiple drive spindles, and may have its own processing power and I/O (input/output) bandwidth. The inclusion of these resources means that as storage capacity increases, performance also increases.

III. CONCLUSION

This paper discusses about the key factors, using which decides how evolution of scale out storage over scale up storage takes place. Scale-out storage can eliminate scalability and performance bottlenecks and allow disk array growth to match the load being applied. Using some factors like capacity, cost, resource utilization and scalability it can be decided that scale out storage is more efficient than scale up. These factors elaborated in the above section will guide us in designing and developing system using the scale out storage.

REFERENCES

[1]. Desai, P. and Jayakumar, N., An Extensible Framework using Mobilityrpc for Possible Deployment of Active Storage on Traditional Storage Architecture.
[2]. Desai, P.R. and Jayakumar, N.K., A Survey on Mobile Agents.
[3]. Jayakumar, N. and Kulkarni, A., 2017. A Simple Measuring Model for Evaluating the Performance of Small Block Size Accesses in Lustre File System. *Engineering, Technology & Applied Science Research*, *7*(6), pp.2313-2318.
[4]. RishikeshSalunkhe, N.J., 2016. Query Bound Application Offloading: Approach Towards Increase Performance of Big Data Computing. *Journal of Emerging Technologies and Innovative Research*, *3*(6), pp.188-191.
[5]. Salunkhe, R., Kadam, A.D., Jayakumar, N. and Joshi, S., 2016, March. Luster a scalable architecture file system: A research implementation on active storage array framework with Luster file system. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)* (pp. 1073-1081). IEEE.
[6]. Jayakumar, N., Iyer, M.S., Joshi, S.D. and Patil, S.H., A Mathematical Model in Support of Efficient offloading for Active Storage Architectures.
[7]. Naveenkumar, J. and Joshi, S.D., 2015. Evaluation of Active Storage System Realized through MobilityRPC.
[8]. Jayakumar, N., Bhardwaj, T., Pant, K., Joshi, S.D. and Patil, S.H., A Holistic Approach for Performance Analysis of Embedded Storage Array.
[9]. Naveenkumar, J., Makwana, R., Joshi, S.D. and Thakore, D.M., 2015. Performance Impact Analysis of Application Implemented on Active Storage Framework. *International Journal*, *5*(2).
[10]. Naveenkumar, J. and Joshi, S.D., 2015. Evaluation of Active Storage System Realized Through Hadoop.
[11]. Zaeimfar, S.N.J.F., 2014. Workload Characteristics Impacts on file System Benchmarking. *Int. J. Adv*, pp.39-44.
[12]. Jayakumar, D.T. and Naveenkumar, R., 2012. SDjoshi,“. *International Journal of Advanced Research in Computer Science and Software Engineering," Int. J*, *2*(9), pp.62-70.
[13]. Suryawanshi, A.U. and Naveenkumar, J., Privacy-Preserving for a Secure Data Storage on Cloud Using Public Auditing Technique.
[14]. Suryawanshi, A.U. and Kumar, J.N., A Multiuser Model of Privacy-Preserving Auditing for Storing Data Security in Cloud Computing.
[15]. Naveenkumar J and Raval K.S, Clouds Explained Using Use-Case Scenarios.
[16]. Sagar S Lad, S.D Joshi, N.J, Comparison study on Hadoop’s HDFS with Lustre File System. International Journal of Scientific Engineering and Applied Science, 1(8), pp.491–494, 2015.
[17]. Jayakumar, N., Bhardwaj, T., Pant, K., Joshi, S.D. and Patil, S.H., 2015. A Holistic Approach for Performance Analysis of Embedded Storage Array. *Int. J. Sci. Technol. Eng*, *1*(12), pp.247-250.
[18]. Kumar, N., Kumar, J., Salunkhe, R.B. and Kadam, A.D., 2016, March. A Scalable Record Retrieval Methodology Using Relational Keyword Search System. In *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies* (p. 32). ACM.

BIOGRAPHY

Punam Jagannath Desai
Department of Computer Engineering, Bharati Vidyapeeth College of Engineering, Pune.