Proceedings of the VIII International Conference "Distributed Computing and Grid-technologies in Science and Education" (GRID 2018), Dubna, Moscow region, Russia, September 10 - 14, 2018

DISTRIBUTED VIRTUAL CLUSTER MANAGEMENT SYSTEM

V.V. Korkhov [1,a], S.S. Kobyshev [1], A.B. Degtyarev [1], A. Cubahiro [2], L. Gaspary [3], X. Wang [4], Z. Wu [4]

[1] *Saint Petersburg State University, 7/9 Universitetskaya nab., St. Petersburg, 199034, Russia*

[2] *Saint-Petersburg Electrotechnical University LETI, 5 prof. Popova, St. Petersburg, 197376, Russia*

[3] *Universidade Federal do Rio Grande do Sul, Av. Bento Gonçalves, 9500, Campus do Vale - Bloco IV, Caixa Postal 15064, 91501-970, Porto Alegre, Brazil*

[4] *Beijing Normal University, 19 Xinjiekouwai St, BeiTaiPingZhuang, Haidian Qu, Beijing Shi, 100875, China*

E-mail: [a] v.korkhov@spbu.ru

An effective cluster management system is the key to solving many problems that arise in the field of distributed computing. One of the major problems is to map applications and available computing resources in such a way that optimal application performance is achieved and resources are used efficiently. The traditional way is to tune applications to the resources which is not always easy or effective. We consider the opposite way that implies configuring resources to the application needs. This leads to the necessity of abstracting resources which can be done with virtualization technologies. In this paper we consider the execution of distributed applications in virtual clusters that are configured specifically to match application requirements to resources. In particular, we investigate performance of NAS Parallel Benchmarks and Hadoop benchmarking suites in different resource sharing scenarios.

Keywords: Cloud Computing, High Performance Computing, Virtualization, Virtual cluster.

1. Introduction

Container-based virtual infrastructures bring new possibilities to enhance parallel and distributed applications. We have already addressed approaches on flexible configuration of light-weight virtualized computing and networking resources in earlier works [1,2,3]. Such possibilities can increase application performance and resource utilization with minimal impact on simultaneously running applications sharing common set of resources. In this paper we give a short overview of results focusing on deployment and execution of distributed applications and data processing frameworks in virtual container-based clusters that are configured specifically to match application requirements to resources. In particular, we investigate performance of NAS Parallel Benchmarks and Hadoop benchmarking suites depending on resource restrictions and existence of other simultaneously running applications.

2. Motivation and approach

The general motivation of the approach we promote is the following:

- Make distributed computing system easier to use and to manage in order to deliver optimal performance to applications and efficiently utilize resources;
- Allocate as much resources as needed by applications without overprovisioning;
- Enable controlled concurrent use of shared resources with minimal impact on application performance;
- The overall goal: provide user applications with access to just as much resources as needed, and try to optimize shared resource usage

We consider following major classes of applications that can benefit from our approach:

- Parallel applications in cloud-based distributed systems, e.g. MPI applications
- Frameworks for distributed data processing, e.g. Apache Hadoop

The approach is built upon following assumptions:

- Each application gets its own tailored virtual computing environment to achieve optimal performance. We tune the computing infrastructure to optimize application performance and optimally distribute virtualized physical resources between applications, which represents the application-centric approach. Different applications do not compete for shared resources as the resources are isolated by virtualization technologies.
- Virtualization of resources is used as an underlying technology to abstract virtual computing environments from real resources: it helps create virtual clusters that match application profiles (configurable CPU, memory, network). Light-weight virtualization with less overhead is preferred, however there is no limitation on virtualization technology; flexible configuration of infrastructure is possible to match actual application requirements;
- Different applications have different profiles and requirements; these requirements can be complementary and can be matched in such a way that applications share single resource without hampering each others performance.

Virtual clusters are used as the main abstraction of computing resources configured according to the needs of parallel applications. These are the important features of virtual clusters (VC):

- VC is a collection of virtual nodes working together to solve a computational problem;
- Virtual nodes can be provided by different applicable virtualization technologies: light-weight virtualization (containers), full or para-virtualization;
- VC can be configured either by advanced users - they know exactly what they want (CPU, memory, IO, network) and can precisely control allocated resources (CPU, memory, etc); or automatically using special methods of application analysis to discover application requirements;

- VC can be flexibly adjusted to the needs of an application both before and during application run-time; capacity of unclaimed resources remains available for other applications on a limited set of hardware

3. Experimental use cases

There are two major use cases that we considered during the experiments:

1. High-performance parallel applications, e.g. MPI programs, typically computation-intensive
2. Distributed data processing frameworks, e.g. Apache Hadoop, typically data-intensive

The first use case is investigated with help of NAS Parallel Benchmarks (NPB) suite. These benchmarks are derived from computational fluid dynamics (CFD) applications and consist of several kernels and pseudo-applications. We selected the following application kernels for the experiments:

- FT - discrete 3D fast Fourier Transform, all-to-all communication
- CG - Conjugate Gradient, irregular memory access and communication
- MG - Multi-Grid on a sequence of meshes, long- and short-distance communication, memory intensive

Figure 1 illustrates different scenarios of resource usage: fig. 1a depicts the concurrent scenario when applications work in separate isolated containers preconfigured to provide limited amount of resources to each application; fig. 1b depicts the shared scenario when all applications share resources without isolation.

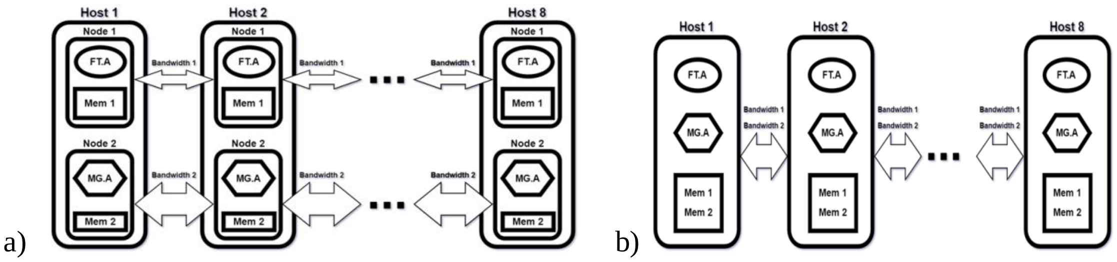


Figure 1. Service high level architecture: a) concurrent; b) shared

The following experimental testbed was used: 8 nodes of MS Azure resources (instance type A1, 1 core, 1.75M RAM) and Docker Swarm for managing container clusters. We evaluated concurrent and shared execution of two application kernels, MG and FT, to ensure that concurrent execution of both applications does not affect their performance in case container clusters are configured to meet the individual requirement of the applications. Figure 2 shows experimental comparison of concurrent and shared execution time depending on throttled networking bandwidth. It is clear that on low bandwidth application kernels compete for shared resources in the second case which results in overall performance degradation.

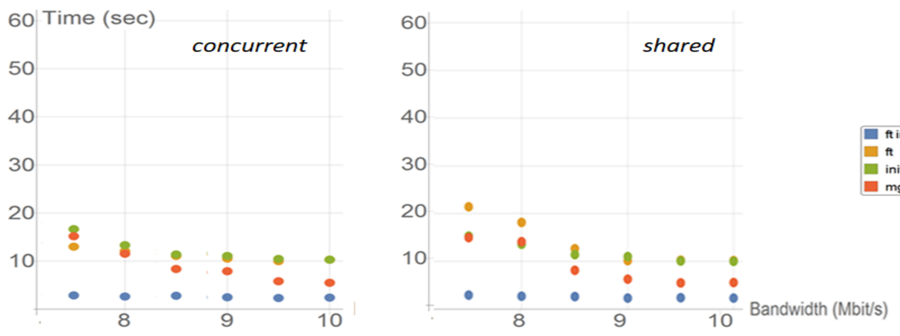


Figure 2. FT and MG NAS benchmarks performance

The second use case concerns using distributed data processing frameworks over a set of resources running concurrent virtual clusters. We considered Apache Hadoop as an example of such framework and investigated the performance of several Hadoop benchmarks:

- TestDFSIO
  - read and write storage throughput test for HDFS
- TeraSort suite (TeraGen + TeraSort + TeraValidate)
  - performs significant computation, networking, and storage I/O workloads;
  - combines testing the HDFS and MapReduce layers of a Hadoop cluster;
  - often considered to be representative of real Hadoop workloads;
  - divided into three parts: generation, sorting, and validation.
- MRBench
  - runs small jobs a number of times and checks whether small jobs are responsive

Figure 3 illustrates the experimental setup. Concurrent virtual clusters with separate Hadoop deployments were used over Amazon AWS t2.large virtual machines (2 vCPUs, 8GB RAM).

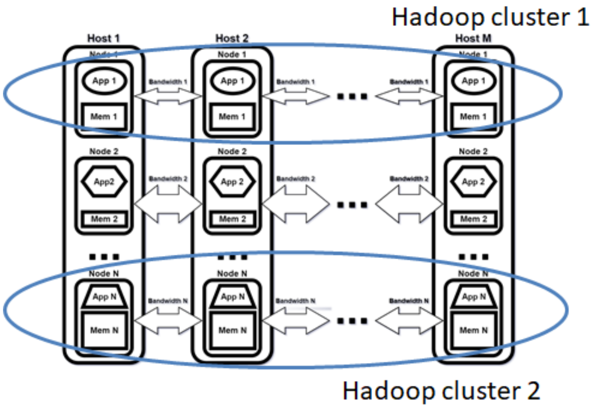
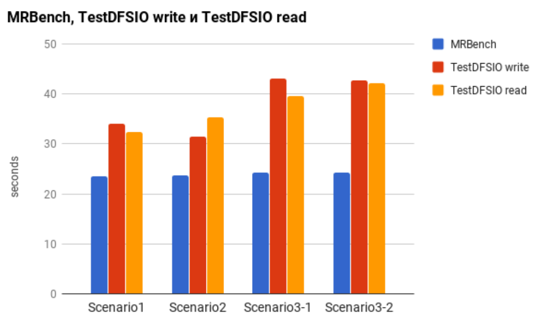


Figure 3. Experiments with Hadoop

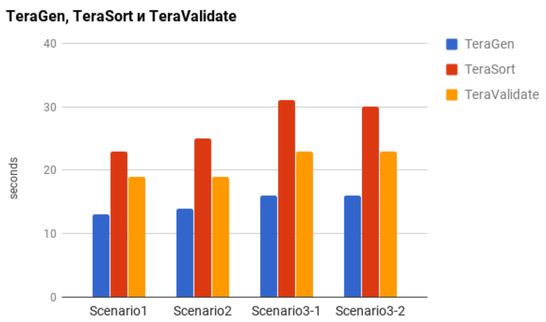
Experiment scenarios:

<u>Scenario1:</u> Every VM runs a single Docker container that uses full VM resources without constraints; <u>Scenario2:</u> Every VM runs a single Docker container constrained to use only 4GB RAM; <u>Scenario3:</u> Every VM runs two Docker containers, each constrained to use only 4GB RAM; two Hadoop clusters are deployed in parallel on containers, thus every VM is shared between two simultaneously running Hadoop clusters.

The results for different scenarios are presented in Figure 4 (Scenarios 3-1 and 3-2 correspond to each VC in Scenario 3). MRBench performance does not depend on the scenario since it focuses on MapReduce without much use of distributed file systems and only uses CPU. TestDFSIO significantly depends on the scenario: in Scenario 3 both read and write tests perform significantly slower than in Scenarios 1 and 2, though not twice as slow, which demonstrates the efficiency of using parallel clusters. TeraSort shows only a slight decrease of performance in Scenario 3: we managed to process twice as much as the original TeraSort workload increasing the overall processing time just for about 15 percent.



a)



b)

Figure 4. Experimental evaluation of MRBench, TestDFSIO, TeraSort Hadoop benchmarks

4. Conclusion

In this work we analysed experimental usage of configurable virtual clusters on different workloads: parallel applications (NAS Parallel Benchmarks) and distributed data processing frameworks (Apache Hadoop benchmarks: MRBench, TestDFSIO, TeraSort). We demonstrated that efficiency of using distributed resources can be increased - even in case of utilizing cloud resources - by simultaneous execution of light-weight virtual clusters on a single set of resources. We showed that flexible configuration of container clusters with standard tools helps allocate proper amount of resources and control free available resources. We still need to profile (or model) applications to specify realistic requirements depending on input data, which is our planned future work.

Acknowledgement

The research was partially supported by the National Key Cooperation between the BRICS Program of China (No.2017YE0100500) and Beijing Natural Science Foundation of China (No.4172033); by CAPES and CNPq in Brazil, Project Number 441892/2016-7, Call CNPq/MCTIC/BRICS-STI No 18/2016; by the RFBR project 16-07-01111 and grant of Saint Petersburg State University no. 26520170 (Russia).

References

[1] Korkhov, V., Kobyshev, S., Krosheninnikov, A. Flexible configuration of application-centric virtualized computing infrastructure. Lecture Notes in Computer Science, vol. 9158, pp. 342–353. Springer, Cham (2015). DOI: 10.1007/978-3-319-21410-8 27

[2] Korkhov, V., Kobyshev, S., Krosheninnikov, A., Degtyarev, A., Bogdanov, A. Distributed computing infrastructure based on dynamic container clusters. Lecture Notes in Computer Science, vol. 9787, pp. 263-275. Springer, Cham (2016). DOI: 10.1007/978-3-319-42108-7_20

[3] Kuchumov, R., Petrunin, V., Korkhov, V., Balashov, N., Kutovskiy, N., Sokolov, I. Design and implementation of a service for cloud HPC computations. Lecture Notes in Computer Science, vol. 10963, pp. 103-112, Springer, Cham (2018). DOI: 10.1007/978-3-319-95171-3_9