IBM

Building a Compute Grid on Apache Hadoop using Cloud Computing

A case study at the University of Pretoria

Executive sponsor: Willy Chiu, Vice President, High Performance On Demand Solutions (HiPODS), IBM Software Group

Management contact: Jeffrey L. Coveyduc, coveyduc@us.ibm.com

Technical contacts: Stefan van der Stockt, stefanvd@za.ibm.com
Andrew Hately, hately@us.ibm.com
Justin Coetsee, justinco@za.ibm.com
Satish Babu, r1sbabu@in.ibm.com,

Web address: www.ibm.com/developerworks/websphere/zones/hipods/

Date: 1 December 2008

Status: Version 1.0

©IBM Corporation 2008

Abstract: Computational intelligence (CI) algorithms, such as those developed by the University of Pretoria, South Africa, require a lot of computing resource to complete simulation and development in reasonable time. In an environment with many students, where each student's workloads can take days to complete, managing workloads and resources can become a full-time challenge.

Cloud computing dynamically provisions, configures, reconfigures, and deprovisions resources as required. This paper discusses how cloud computing was applied to solve the scheduling and resource management requirements of the University of Pretoria, as well as provide grid computing capability for testing CI algorithms.

Executive summary

The Computational Intelligence Research Group (CIRG) at the University of Pretoria, South Africa, consists of fifty PhD, masters, and honors degree students. CIRG specializes in researching, developing, and performance testing new computational intelligence algorithms and applications.

A single computational intelligence workload can take a few days to weeks to complete on a high-end workstation. Each student has a number of these workloads to conduct and analyze before finalizing their degrees. CIRG faced a number of challenges:

- Students did not have dedicated hardware, resulting in students trying to 'borrow' workstations from each other in a haphazard and ad hoc manner.
- With roughly one workstation for each student, it was not possible for multiple students to reliably run workloads concurrently, as there was no central management of which student's workloads were running on which machines.
- Students had no way of booking how many machines they would require, the date, or the period of usage. All workstations were requested and used on good faith.
- No data management was present. Students had to manually collect experimental data results from each of the workstations they managed to obtain access to.
- No central workload management was present as students manually started workloads on each machine. If a workload that took three days to complete finished execution at say 9:00 PM, no new workload would be started until the next morning, resulting in the loss of roughly half a day's potential execution.

The Africa Software Solution Laboratory solved the problems that CIRG faces by deploying an IBM cloud at the University of Pretoria. The IBM cloud solution is based on the IBM® System x® BladeCenter® technology and has a starting capacity of forty virtualized images on a fully loaded BladeCenter chassis supported by 5TB of storage for research needs. One of the specialized workloads on this cloud environment includes IBM Idea Factory for the collaboration needs of the CIRG research community. Benefits to the University of Pretoria include:

- Reduced research computational time from weeks to days
- The solution provides a platform that allows students to rapidly deploy grid applications
- A new mechanism for the research team to use open ideation
- The ability to easily manage and rapidly deploy new capacity to their infrastructure

Introduction

The Computational Intelligence Research Group (CIRG) of the University of Pretoria, South Africa, focuses on developing, testing, and implementing computational intelligence (CI) algorithms. CIRG consists of about fifty PhD, masters degree, and honors degree students from all over the world. The main focus areas of CIRG are new algorithm design, algorithm analysis, and applications of CI algorithms to solve real-world problems, with students obtaining a PhD, masters, or honors degree in their chosen field. Each student has a defined hypothesis or problem that they want to solve using CI techniques.

Computational intelligence is a subset of artificial intelligence, consisting of algorithms and techniques that combine elements of learning and adaptation to create programs that are, in some sense, intelligent. The main subfields of CI include artificial neural networks, evolutionary computation, artificial immune systems, swarm optimization, and fuzzy logic. CI approaches have been used to solve a variety of problems, ranging from financial prediction, RNA protein folding prediction, engineering process optimization, telecommunications routing, medical applications, medical research, economic applications, IT optimization, data mining, and many others.

CIRG has developed an asset called the computational intelligence library, or CIlib. CIlib is an open-source Java™ framework comprised of CI algorithms and problems. The library is designed to be 'parallel computation ready,' but does not provide the capability to do this as part of its scope. To date, CIlib is already being used by many other universities in Asia and the United States.

The challenge facing computational intelligence

Part of each student's degree is to develop and test a CI solution that solves the problem being investigated. As CI algorithms are based on heuristic search, this involves searching through so-called 'solution space' to try to find the optimal solution to the problem. The CI algorithm dictates how to proceed in this search for the optimal solution – the better the algorithm, the better and quicker the search result.

The challenge that students face is that the problems they are trying to solve are not trivial. This means that the search space for the CI algorithms can become extremely big, resulting in very computationally expensive workloads. To achieve statistical significant results, each student's workload needs to include in the order of a thousand experiments, using different parameters, inputs, problem types, etc. Each student's workload could potentially take days or weeks, and some extreme cases even months to compute on a single workstation running 24 hours a day. Furthermore, this process needs to be performed by each of the fifty students.

Students from CIRG attempt to solve this challenge by running their experiments on more than one workstation simultaneously. This provides some improvement in terms of throughput, scalability, and failover (if one machine fails, others are still running). A drawback of this method is that students attempt to run experiments by haphazardly trying to 'borrow' machine capacity from each other. This presents a scheduling and management nightmare, as each student requires at least five to ten machines to complete their experiments in a feasible amount of time. There is roughly one workstation for each student, which presents a major challenge if more than five workloads have to run concurrently.

Another drawback is that a fairly large part of the students' time was being used to optimize their workstation performance, as even a performance improvement of 5% could save days of computation. This started to detract the students from their actual work, which was researching and developing CI algorithms and applications.

Automated system management

Cloud computing is defined by Quan et al [QUAN2007] to describe both a platform and a type of application. A cloud computing platform dynamically provisions, configures, reconfigures, and deprovisions servers as needed. Servers in the cloud can be physical machines or virtual machines. Advanced clouds typically include other computing resources such as storage area networks (SANs), network equipment, firewalls, and other security devices. Figure 1 illustrates how Xen virtualization is used to segment a physical machine into multiple virtual machines that the IBM cloud can utilize.

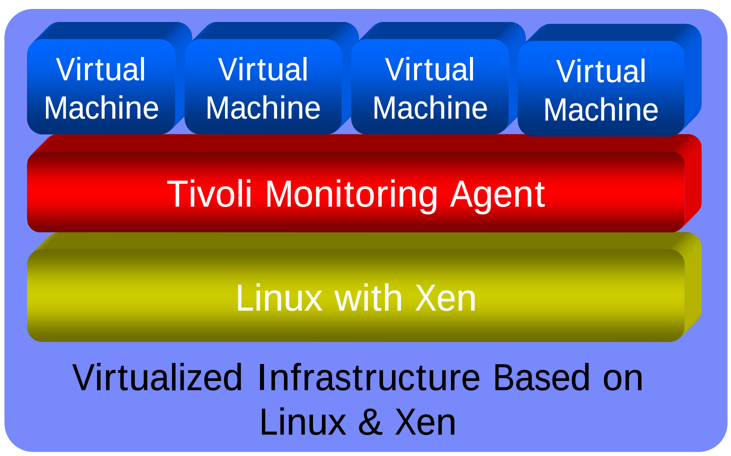

Figure 1 Linux with Xen Virtualization as an enabler for the IBM cloud solution

A cloud is a pool of virtualized computer resources that can:

- Host a variety of different workloads, including batch-style back-end jobs and interactive, user-facing applications
- Allow workloads to be deployed and scaled-out quickly through the rapid provisioning of virtual or physical machines
- Support redundant, self-recovering, highly scalable programming models that allow workloads to recover from many unavoidable hardware/software failures
- Monitor resource use in real time to enable rebalancing of allocations when needed.

In a cloud, a Web-based interface is used to book, schedule, and manage machines. The Web interface gives an overview of the cloud infrastructure, including which projects are currently defined, how many resources each project uses, or will use in the case of future projects. A consolidated view of available and utilized resources in the cloud is also shown through the use of real-time monitoring. Administrators can also drill down into projects and see detailed monitored information for each machine. Users can request new machines to be provisioned by merely selecting the desired CPU, memory, storage, operating system, and any additional software packages to be deployed. After the request is submitted, an administrator can approve, reject, and/or modify the request as required. After the request is approved, the cloud system automatically creates and provisions the requested machines in about an hour.

The IBM cloud solution dramatically simplifies the management, scheduling, and booking of computing resources. For a detailed overview and further benefits of cloud computing, refer to [QUAN2007].

Grid applications with Apache Hadoop

Apache Hadoop [HADOOP2007] is an open-source framework for running parallel computing applications on large clusters of commodity hardware. Apache Hadoop is based on the MapReduce algorithm [MAP2004]. MapReduce is a programming model that allows a large task to be broken down (or mapped down) into multiple smaller tasks that can be processed as individual jobs. The reduce function combines the output of all the smaller jobs in a specified manner to provide the output of the original large task. The Apache Hadoop framework takes care of job management aspects such as keeping track of which jobs run on which nodes, which jobs complete successfully, which jobs need to be restarted due to failures, and other tasks.

As Hadoop jobs run on a distributed cluster, data management is of key importance. Apache Hadoop uses the Hadoop distributed file system (HDFS) to create multiple replicas of data items across different nodes in the cluster. Using this approach, data reliability is increased through redundancy. Data is also kept close to the computing resource that uses it, which increases performance.

The combination of the MapReduce algorithm and HDFS enables parallel grid applications to be developed and deployed rapidly and easily, with minimal development time being spent on grid management aspects.

Cloud computing at CIRG

The IBM cloud solution from IBM provides the CIRG students with the capability to schedule and reserve one or more machines for use by authorized students. The system automatically provisions and configures the requested machines as stipulated by the reservation period. The IBM cloud solution forms the platform upon which students can deploy their workloads.

The Apache Hadoop framework running in a cloud computing environment is a natural fit to solve the challenges of the University of Pretoria. Hadoop provides the framework for students to submit a single job, which automatically gets broken down into multiple smaller jobs. Hadoop manages the jobs, which are executed on the available machines and collates the results to provide the student with a a single output result.

The combination of Apache Hadoop running on the IBM cloud infrastructure provides CIRG with a rich platform to run and manage their complex workloads.

Automatically managing machine availability

The IBM cloud solution directly solves CIRG's challenges of workstation scheduling and availability.

Challenge:	Solution:
Students did not have dedicated hardware, resulting in them 'borrowing' workstations from each other on an ad hoc basis.	**The cloud infrastructure is deployed on an IBM Blade Center filled with HS12 Intel Xeon quad core blades with 10GB of memory. This provides students with a fair amount of computation power that can be reserved through the cloud interface. Up to four Xen virtual machines can be deployed per blade.**
Students had no formal way of knowing which workstations were available, for how long, or if anybody else was busy using them.	**The cloud allows students to reserve a certain amount of machines for a specified period of time. The central management console gives a view of what the available capacity is at any given moment, and which projects are using the cloud resources.** **In-depth Tivoli® monitoring provides a single view into the infrastructure, allowing administrators to have a detailed view of which machines are being utilized fully.**
Students had to manually make sure that each borrowed workstation had the correct level of operating system, software, Java version, and CIlib library installed.	**When a student's scheduled timeslot arrives, the requested servers are automatically provisioned, along with any required software libraries such as the correct Java version, CIlib libraries, and any other requested packages. The student is provided with clear log-in details for each requested server, which allows the student to start work immediately.**
Students had to ensure that the workstations they borrowed were returned to them in an orderly fashion after execution. This often meant long manual cleanup cycles per workstation.	**After students finish working on their servers, the cloud management system automatically deprovisions the servers and allocates the resources to the next student (if a project is scheduled), or simply adds the resources back to the pool of available resources.**

The IBM cloud computing offering provides a clean solution to the requirement of CIRG to allow students to book and secure computation resources in a managed and effective way.

Building a grid on cloud

The IBM cloud computing solution alleviates the pain of booking computational resources, provisioning and configuring the resources according to students' needs, and providing a full management console to allow CIRG to see exactly how each resource is utilized. However, the cloud solution does not solve the challenge that students face regarding managing the execution of their workloads, nor does it address the manual data collection and management aspects.

Previously, each student had to manually start the workload on each workstation that he/she had access to. After execution, the generated data had to be meticulously and manually collected from each machine for post-processing. The IBM cloud solution provides a clean way to allocate and provision machines (virtual or physical) to students, but does not make job management and data manipulation any easier. Students still have to manually start jobs and collect data afterwards.

The solution was to investigate grid computing running on the IBM cloud. CIlib was designed from the start to be parallel computation ready, but CIRG had no way to run CIlib as a grid application. The Apache Hadoop framework provides the capability to run a single CIlib workload on multiple machines at once. The MapReduce capability of Hadoop automatically creates and manages multiple jobs that are based on the smallest unit of work in CIlib, namely a simulation.

Students have to stipulate the makeup and parameters for an experiment in a CIlib XML setup document. Each experiment consists of multiple simulations (usually 30 to a 100 simulations). Each simulation takes between thirty minutes to a couple of hours to complete, with extreme cases taking up to 24 hours to complete. In a nongrid environment, each simulation is run sequentially on a single machine. Using Hadoop, each simulation is broken down into individual jobs that can be executed simultaneously on multiple Xen virtual machines in the cloud. Job management, reliability, and failover are automatically managed by Hadoop, and require no extra development from CIRG students.

The output of each Cilib experiment is a set of measurements, as defined by the XML setup document. In a non-grid environment, individual results have to be manually collected and merged using data manipulation applications such as gawk, gnuplot, and spreadsheets. Hadoop uses the reduce function of MapReduce to combine the results of all jobs and present a single coherent result to the original submitted job. This greatly alleviates the pain of manual post-processing of data results.

Summary

Before using the IBM cloud, students at CIRG had to manually borrow, configure, and manage each other's workstations on an ad hoc basis. Students also did not have access to dedicated computing resources, which limited the number of workloads that students could run concurrently. The IBM cloud solution provides the ideal solution to allow the students of the Computational Intelligence Research Group of the University of Pretoria, South Africa, to effectively manage and automate their computing resources. The IBM cloud allows students to book, provision, and acquire computing resources using a central Web-based interface. The correct operating system, software, Java version, and CIlib version is automatically deployed on the specified number of Xen virtual machines.

Apache Hadoop is used on top of the IBM cloud infrastructure to provide a dynamic grid-based execution environment upon request. Hadoop uses the MapReduce algorithm to break a large CIlib workload into many smaller jobs. All jobs are centrally managed by Hadoop.

Using the IBM cloud and Hadoop, the CIRG students at the University of Pretoria realize a number of benefits:

- Reduced research computational time from weeks to days
- The solution provides a platform that allows students to rapidly deploy grid applications
- A new mechanism for the research team to use open ideation
- The ability to easily manage and rapidly deploy new capacity to their infrastructure

References

QUAN2007 Cloud Computing,
Greg Boss, Padma Malladi, Dennis Quan, Linda Legregni, Harold Hall
IBM HiPODS,
www.ibm.com/developerworks/websphere/zones/hipods/

HADOOP2007 Apache Hadoop Project,
http://hadoop.apache.org/

MAP2004 MapReduce: Simplified Data Processing on Large Clusters
OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004.
http://labs.google.com/papers/mapreduce.html

Acknowledgements

We acknowledge this paper's major supporters, contributors, and reviewers:

- Willy Chiu, Vice President, High Performance On Demand Solutions, IBM Software Group Strategy
- Jayashree Subrahmonia, Director, Advanced Customer Solutions, IBM Software Group Strategy
- Linda Legregni, Web 2.0 Strategy, IBM Software Group Strategy

Notices

Trademarks
The following are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM, IBM logo, BladeCenter, System x,Tivoli

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other counties, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrio logo, Celeron, Interon, Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Special notice
The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While IBM may have reviewed each item for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Anyone attempting to adapt these techniques to their own environments do so at their own risk.

While IBM may have reviewed each item for accuracy in a specific situation, IBM offers no guarantee or warranty to any user that the same or similar results will be obtained elsewhere. Any person attempting to adapt the techniques contained in this document to their own environment(s) does so at their own risk. Any performance data contained in this document were determined in various controlled laboratory environments and are for reference purposes only. Customers should not adapt these performance numbers to their own environments as system performance standards. The results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.