Goscinski, Andrzej, Brock, Michael and Church, Philip C. 2012, High performance computing clouds, in Cloud computing : methodology, systems and applications, CRC Press, Boca Raton, Fla., pp.221-259.

11

High Performance Computing Clouds

Andrzej Goscinski

School of Information Technology, Deakin University, Geelong, Australia

Michael Brock

School of Information Technology, Deakin University, Geelong, Australia

Philip Church

School of Information Technology, Deakin University, Geelong, Australia

CONTENTS

In recent times, cloud computing has gained the interest of High Performance Computing (HPC) for providing HPC environments on-demand, at low cost and in an easy to use manner. While the potential for HPC clouds exists, there are challenges (such as performance and communication issues) and some experts deem cloud computing inappropriate for HPC.

This chapter investigates what challenges exist when attempting to use clouds for HPC and demonstrates the effectiveness of HPC cloud computing through broad benchmarking and through a new prototype cloud called HPCynergy - a cloud built solely to satisfy the needs of HPC clients. This chapter demonstrates, via benchmarking comparisons between clouds such as EC2, HPCynergy, virtual and physical clusters that HPC clouds have a bright future and that most claims against the use of cloud computing for HPC are exaggerations.

11.1 Introduction

A recent trend in computing research is the use of clouds to support and/or provide complete environments for high performance computing (HPC) applications. The motivation for this trend is to support HPC applications without the excessive cost and complexity overheads. HPC applications are software used to carry out complex algorithms on large data set (often terabytes in size) in the following fields:

- Bioinformatics – to understand diseases and biological systems through DNA analysis
- Physics – to discover particles often using the Large Hadron Collider
- Engineering – to design better cars and planes and make environments green(er)

Traditionally, clusters have been used to run HPC applications: rather than run the whole application on a single system, subsets of the application are run concurrently across multiple cluster nodes so that results are returned within reasonable timeframes. However, clusters are not cheap to purchase and maintain and are usable only by computing specialists. Hence the attention of cloud computing for HPC is to reduce costs, offer on-demand services and simplify use.

Currently specialized High Performance Compute Clouds (HPC Clouds) are only offered by a few providers such as Amazon [60] and SGI [373]. These specialized cloud instances usually make use of nodes with HPC-specific hardware (Infiniband or Gigabit Ethernet interconnects between nodes, special processors, etc.), HPC middleware and some form of hardware monitoring.

While there is the promise of cheap, on-demand HPC, cloud computing is a completely different paradigm thus exhibits unexpected side-effects. Cost wise, there is no guarantee that paying more for additional cloud nodes improves performance [636]. As most clouds use virtualization, HPC software applications may execute abnormally and there is no guarantee that all cluster nodes in a cloud are within close proximity to each other.

Furthermore, cloud interoperability does not exist: whole computer systems created on one cloud are not transferable to another cloud. This issue is so significant that even Intel proposed it as a requirement in their vision of cloud computing [343]. Finally, HPC clouds still lack a number of important software features: (i) current cloud management tools focus on single servers, not whole clusters, (ii) tools for publication and discovery only focus on service functionality and ignore other parameters such as quality and current activity.

In general, while the potential of on-demand HPC via clouds exists, there is no solution that combines cloud computing and HPC together. The aim of this chapter is to present a review of publications on combining clouds and HPC, highlight existing challenges faced when using clouds for HPC, present a new HPC cloud, and demonstrate, via experimentation, the applicability of HPC clouds.

The rest of this chapter is as follows. Section 11.2 presents arguments by leading experts both against and in favor of using cloud computing for HPC. Section 11.3 provides a definition of HPC clouds and makes a comparison with current physical and virtualized HPC cloud offerings. Section 11.4 details various challenges faced (based on the definition presented in Section 11.3) when using clouds for HPC. In response the design and implementation of a new HPC cloud called HPCynergy is presented in Section 11.5. The feasibility of running bioinformatics and physics applications on HPCynergy (when compared to other cloud computers and dedicated HPC platforms) is investigated in 11.6. Conclusions and future trends are presented in Section 11.7.

11.2 High Performance Computing (HPC) vs. Cloud Computing

Recently, vendors and researchers have turned to cloud computing as a way to access large amounts of computational resources, storage, and software cheaply. In terms of clouds for HPC environments, experts are both for and against the idea.

Dr. Iordache rejects the possibility claiming that HPC is about performance while virtualization in clouds is about adding more latency (overhead) [344]. This claim is not entirely correct as virtualization provides a required platform independent of an existing physical platform. While latency exists, it is not such a significant problem now as it was when it first emerged. Another opponent to HPC clouds is Dr. Kranzmuller, who stated "*there are only very limited uses for the public cloud in HPC ... For applications with larger storage requirements or closely coupled parallel processes with high I/O requirements, clouds are often useless.*" [345].

Dr. Yelick, director of NERSC, noted about current developments in the Magellan cloud, "*there's a part of the workload in scientific computing that's well-suited to the cloud, but it's not the HPC end, it's really the bulk aggregate serial workload that often comes up in scientific computing, but that is not really the traditional arena of high-performance computing.*" [345].

Not all experts are against the idea. An initial study of the effectiveness of clouds for HPC shows that the opportunity is there and is compelling but "*the delivery of HPC performance with commercial cloud computing is not yet mature*" [736]. However, the computational environments used in the study (Amazon's EC2 and NCSA cluster) have inconsistent hardware and software specifications, and the experiment details are not specified precisely.

Professor Ian Foster stated in his blog when addressing these results that "*based on the QBETS predictions, if I had to put money on which system my* *application would finish first, I would have to go for EC2.*" He also added that when HPC clouds are to be assessed, optimization for response time should be considered rather than the utilization maximization used on supercomputers [278].

There are also other strong voices supporting HPC in clouds. In response to the increasing costs of IT infrastructure, use and maintenance, Microsoft's Dan Reed stated, "*fortunately, the emergence of cloud computing, coupled with powerful software on clients ... offers a solution to this conundrum.*" He adds that "*the net effect will be the democratization of research capabilities that are now available only to the most elite scientists.*" [346].

An analysis of HPC and clouds shows that (i) the concept of HPC clouds has been rejected by some researchers because clouds use virtualization; (ii) not all problems require powerful and expensive clusters; (iii) organizations, although they depend on storing and processing large data, they are not prepared to invest in private HPC infrastructures; (iv) HPC on clouds offer cost and scalability advantages to clients (users).

In support to Item (i), vendors such as SGI [373] and Penguin [201] do not offer virtualization in their HPC offerings; they provide customized servers.[1] In this context it is worth presenting the thought of Ms. Hemsoth, "*It's not really the cloud that everyone recognizes if you remove the virtualization, is it? At least not by some definitions. But then again, getting into complicated definitions-based discussions isn't really useful since this space is still evolving (and defining it ala the grid days) will only serve to stifle development.*" [343].

Overall, it is possible to offer HPC as a service: if there are many clusters that provide HPC services, cloud management software offers services transparently, and clients can discover and select a service that satisfies their requirements, then there is a satisfactory condition to say that there is a HPC cloud. While virtualization is often a common element for cloud computing, it should be noted that not all expert definitions of cloud computing [298, 300] consider virtualization as a vital element.

In support to Item (ii), there are very many bioinformatics, engineering, social sciences problems that could be successfully solved on less powerful clusters and on demand (without a need for waiting in a queue[2] for execution). Furthermore, many clients do not wish to purchase, deploy, run and maintain even smaller clusters.

In support to Items (iii) and (iv), organizations prefer to concentrate on their mission statement activities, and clients prefer to concentrate on their problems, research, and development. Clouds allow on demand resource usage: the allocation and release is proportional to current client consumption, thus clients only pay for what they use. This implies the availability of cloud resources is more important than performance. However, there is no guarantee

---

[1] We will show in this chapter the influence of virtualization on the execution performance of high performance applications on different clouds.

[2] Queues are used with clusters to hold jobs that require more nodes than currently available.

that paying for (renting) extra nodes on a cloud will guarantee significant performance improvement [636]. Finally, even when purchase, installation and deployment of a cluster is complete, the cluster still incurs ongoing costs (mostly electrical power for cooling) long after research projects are finished.

Besides the low cost and simplified access to specialized, high throughput systems, there is also the opportunity to improve collaboration between research projects. Instead of only making large datasets available to various research teams, it is now possible to expose whole research environments as on-demand services. Thus, it is clear that HPC clouds have a bright future.

11.3 Taxonomy of HPC Clouds

This section presents an accurate cloud computing definition based on current literature. Based on the presented definition, a taxonomy of current cloud and HPC solutions is presented. Problems are identified based on what properties from the supplied definition in Section 11.3.1 are not satisfied by existing clouds. If the means by which a property has been exposed is deemed inadequate, the poorly exhibited property is also considered a problem.

11.3.1 HPC Cloud Definitions

Neither cloud computing nor HPC cloud computing have a concrete and commonly accepted definition. According to Gartner [298], clouds are characterized by of five properties:

1. Service-based – all resources within the cloud system are to be made available via services that have well defined interfaces.
2. Scalable and Elastic – clients only use resources (and amounts thereof) they need.
3. Shared – all resources are used at their full utilization and not dedicated to any client.
4. Metered by Use – clients make use of clouds on a pay-as-you-go basis.
5. Uses Internet Technology – cloud services are built and accessed using Internet technologies, often the World Wide Web.

The NIST definition of cloud computing [508] furthers Gartner's definition in that clients have broad network access (the cloud is accessible via any network enabled device) and require clouds to monitor and reconfigure resources based on service usage.

In another study, 21 experts were asked for their definitions of cloud computing [300]. Each expert used at least two of Gartner's properties and some consider virtualization as a property of cloud computing. Some experts even considered that cloud computing offerings fall under one of three broad categories:

- Infrastructure as a Service (IaaS) – virtual servers are provided where clients are able to create their required platforms from the (virtual) hardware upwards.
- Platform as a Service (PaaS) – complete hardware and software configurations (platforms) are offered to clients and maintained by cloud providers.
- Software as a Service (SaaS) – clients are offered software applications/services and only have to focus their use.

In this chapter, cloud computing (based on the description by Gartner and current cloud offerings) is made possible through the combination of virtualization, Web services, and scalable data centers [145]. With the recent emergence of HPC in cloud computing (thus giving rise to the idea of HPC clouds [201, 373] and/or HPC as a service [60]) once again there is the question of what the term HPC cloud stands for and what properties need to be exhibited. Thus HPC clouds are viewed as having the following properties:

1. Web Service-Based – all resources from data storage to cluster job management are done via self-describing Web services with Web APIs for software processes and Web forms for human operators.
2. Use the Pay as You Go Model – all HPC clients are billed for the resources they use and amounts thereof.
3. Are Elastic – if a client has a task that consists of multiple jobs in a required sequence (workflow) HPC clouds are expected to allocate and release required services/resources in response to changes in the workflow.
4. Clusters Are Provided On-Demand – clients should be able to specify requirements and then (i) discover an existing cluster for immediate use or (ii) have an existing cluster reconfigured to satisfy client requirements.
5. Guaranteed Performance – typically, if cluster nodes are allocated to clients, all nodes are expected to be within close proximity to each other.
6. Virtualization – for flexibility, cloud computing will require the use of virtualization. However, virtualization should be made an optional feature to HPC clients: they may have applications that do not work in virtual environments due to high latency sensitivity.
7. Are Easy to Use – clusters (physical or virtual) can be complex thus need to be simplified. First, unintuitive user interfaces (such as command lines) can frustrate human users into finding work around. Second, some researchers (even experts) may wish to spend more

time conducting research rather than on HPC application and application configuration/modification.

These properties of the provided HPC cloud definition are used to propose a simple HPC cloud taxonomy that considered both major types of HPC cloud solutions: virtualized HPC clouds and physical HPC clouds.

11.3.2 Virtualized HPC Offerings in the Cloud

While cloud computing has much to offer HPC (such as low cost, availability and on-demand service provision), few cloud based HPC solutions exist. There are even vendors that claim to provide HPC cloud solutions but are simply resellers for other cloud computing vendors. CloudCycle [214] is such a vendor that sits on top of Amazon and sells HPC products to clients. This section shows what HPC product offerings currently exist from cloud computing providers and examines what properties they exhibit from the HPC cloud definition presented in Section 11.3.1.

11.3.2.1 Amazon Cluster Compute Instances

An IaaS cloud, Amazon's Elastic Compute Cloud (EC2) [59] offers virtual machines (instances) to clients who then install and execute all required software. EC2 offers a catalog of instance types that detail the (virtual) hardware specifications and range from small 32 bit systems to large multicore 64 bit systems [61]. There have even been successful attempts to create clusters within EC2 manually [259, 349] yet others have deemed EC2 unviable solely on the network communication performance [736].

Recently, Amazon has offered the Cluster Compute Instance for HPC [62] which offers a (combined) 23 gigabytes of RAM, 16 Terabytes of storage and use 10 Gigabit Ethernet. In terms of the HPC cloud definition, Cluster Compute Instances only exhibit the properties of being service based, clients paying for what they use, and the use of Gigabit Ethernet for performance. However, Cluster Compute Instances are still not adequate, as the property of ease of use is not exhibited. Clients are required to install cluster management software.

11.3.2.2 Microsoft Azure

A PaaS cloud, Microsoft Azure [515] provides complete hardware and software configurations on-demand. Thus, clients focus on the development and management of their own services and can make the services discoverable via the .NET Service Bus [661]. However, nothing more than a unique URI of each service is publishable.

In terms of support for HPC, Azure only acts as an "auxiliary" to clusters running Microsoft's HPC Server 2008 [516]. On site clusters lacking available resources can connect and off load tasks[3] to Azure. Overall, the only properties exhibited by Azure are the use of Web services, the pay-as-you-go model, and (through the use of clusters extending to Azure) elasticity.

11.3.2.3 Adaptive Systems' Moab

Adaptive Systems is a provider of HPC cluster management software and a provider who helped in the genesis of Amazon's Cluster Compute Instances [341]. While Adaptive does not provide a cloud perse, it is worth describing, as clusters are comprised of hardware and software. Of interest to this chapter are the Moab Adaptive HPC Suite [603] and the Cluster Management Suite which contains a Cluster Manager [604], Workload Manager [605] and Access Portal [602].

The Moab Adaptive HPC Suite allows each cluster node to automatically change its installed operating systems on-demand. For example, a cluster comprised of Linux nodes is able to change a required number of nodes to Windows, thus allowing the job to run. This feature, based on the definition in Section 11.3.1, exhibits the properties of clusters on-demand and ease of use.

The Cluster Manager offers the ability to partition clusters into virtual clusters, thus giving clients the illusion they are operating whole clusters dedicated to them. While Quality of Service functionality is offered, it is limited to functional aspects of cluster jobs such as the maximum amount of time the client will tolerate the job being in a queue. In general, this offering exhibits the cluster on demand property (as with the Adaptive HPC Suite).

In terms of workflows, orchestration of multiple services/executables, there is the Moab Workload Manager. It is a scheduler with a few extra features: it is possible for jobs to change their resource allocations during execution (for example, the first five minutes use a single node, the next five use two nodes, etc.) and integration with existing grids.

Finally, Access Portal provides job submission, monitoring and control to clients with a graphical view of the cluster job load based on the individual nodes. The problem with the portal is that getting jobs on the cluster is user unfriendly; the Web page is mostly a wrapper for a command line. While Adaptive Systems does make reference to HPC on cloud computing, they do not provide a tangible solution. All they offer is architecture for resource consolidation [200].

11.3.3 Physical HPC Offerings in the Cloud

With cloud computing's use of virtualization, performance overheads are often exaggerated by some experts. Irrespective, there are clients that simply prefer to use physical servers over virtual servers. The purpose of this section is

---

[3] Given Azure is a .NET based environment, it is assumed only .NET tasks can be offloaded.

to examine HPC cloud offerings that use physical servers instead of virtual servers.

11.3.3.1 Silicon Graphics Cyclone

Cyclone [373] is a cloud that does not use virtualization and offers itself to clients as both a SaaS and an IaaS cloud. As a SaaS cloud, Cyclone provides a wide array of applications for numerous scientific fields [374]: from BLAST for biology experiments to SemanticMiner for Web ontologies.

In terms of an IaaS cloud, all that can be learned is that clients are able to install their own applications as in EC2 — clients are provided with a server with an installed operating system as a starting point. However, the hardware specifications are extremely vague with the only detail published that the machines use Intel Xeon processors. As with previous cloud offerings, the pay-as-you-go billing model is exhibited. However, that is the only property exhibited.

11.3.3.2 Penguin Computing on Demand (POD)

A similar solution to Cyclone, Penguin Computing's POD [201] offers physical cluster nodes to clients as utilities. Physical servers are used to carry out application execution while clients access POD via allocated (virtual) login servers. While Penguin Computing claims security on its POD offering, an examination of the documentation shows that the security relies on best-practices (trust) [202, 203]. Furthermore, when a client submits a job to a compute node in POD, the nodes performing the execution become dedicated to the client. While advertised as a cloud, no HPC cloud property is exhibited by POD. In general, POD is a remotely accessible cluster — not a cloud by definition.

11.3.4 Summary

In general, the idea of hosting HPC applications on public clouds such as EC2 is possible. However, there is yet to be a solution that combines both the convenience and low cost of clouds with high performance. Some HPC cloud solutions are clouds only in name and only focus on making clusters accessible over the Internet. Thus, clients are significantly involved with the installation and management of software required by their applications.

Of all the properties presented in Section 11.3.1, only the properties of clouds being Web service based, using the pay-as-you-go model, elasticity and ease of use are exhibited. Other important properties such as transparent cloud management, and guaranteed performance, are not exhibited. The latter property is of significant importance — HPC is about running large software applications within reasonable time frames, thus performance is a must.

11.4 HPC Cloud Challenges

While the term HPC cloud is in use, current solutions only exhibit a minimal set of required properties. Besides requiring clients to (in the worst case) be involved in the infrastructure establishment of HPC environments (from finding and configuring data storage to building and configuring HPC applications for execution on clusters), literature shows [342, 636] that clients may also encounter issues with placing HPC software on cloud hosted clusters, interoperability if resources are spread across multiple clouds (or they wish to migrate), discovering required resources and even ensuring they get acceptable cost-to-performance value.

Based on the literature presented in Section 11.3, this section presents the challenges faced when using clouds to support HPC. While the focus is on HPC clouds specifically, many of the issues presented in sections exist in a same or similar form for cloud computing [733] — specifically the infrastructure aspects of clouds.

11.4.1 Interface Challenges

Most cloud and cluster solutions offer command line interfaces [201, 373] but rarely (if at all) present graphical interfaces. Even when provided, graphical interfaces tend to act as command line wrappers (as seen with Moab [602]). In some cases, the only Web interface is an API for Web-enabled software processes. Besides frustrating and confusing to general users, users who are computing experts may prefer to spend less time getting applications to work on clusters and more time conducting research.

The best cloud interface so far is that offered by CloudSigma [196]. Using a desktop-like interface, CloudSigma is easy and best suited for human operators as it uses a common interface approach. What is lacking is a facility to manage multiple servers at once. While the need for interfaces is mentioned in [733], it examined the need for having a private cloud communicate with a public cloud in the event of resource shortages. This supports the claims that most interfaces offered are programming/infrastructure oriented and often overlook the need for human oriented interfaces.

11.4.2 Performance Challenges

At the very least, a cluster (physical or virtual) hosted in a cloud should run applications as fast (or even faster) than existing physical clusters. Currently, there is no mechanism in place to evaluate performance factors such as current CPU utilization and network IO. Furthermore, there is no service that allows the evaluated factors to be published.

Immediately, this makes it impossible to discover servers and clusters based on quality of service (QoS) parameters. Also, it makes it impossible for the cloud to be intelligent (Section 11.4.4) — the cloud is not able to see that (for example) a virtual cluster is under performing due to high network latency thus will not make attempts to transparently reconfigure the cluster to use a different (idle) network.

Currently, the only parameter guaranteed with clouds is high availability (as seen with EC2 [62]). Similarly, this issue appears as a resource management and energy efficiency issue in [733]. While it is true that allocation of physical CPU cores to virtual machines can significantly improve performance, the approach can increase network latency as virtual machines are placed farther and farther apart so they have effective use of physical CPU cores.

11.4.3 Communication Challenges

A common issue with clouds is their network performance, e.g., EC2 [736]. Network performance cannot be solved by using an Infiniband network alone. There is still the issue of network topologies within clouds. Even if a cloud is hosted within a single location (organization), the virtualized nature of clouds may cause a single (virtual) cluster to be hosted across multiple systems (separate datacenters).

At the time of writing, no cloud could be found that details how or where client virtual machines are placed. As stated at the end of Section 11.4.2, communication overheads and performance are at odds with each other — spreading out virtual machines may improve (CPU) performance, but the overall performance may remain unchanged due to increased network latency caused by VM placement.

11.4.4 Intelligence Challenges

While cloud computing offers resources as a utility, they are not always intelligent when allocating resources to clients. For example, if a client needs additional (virtual) servers, the client has to invoke functionality in the cloud itself to create the new servers.

The other issue is the placement of (virtual) servers in relation to each other. While some HPC cloud offerings claim to use Infiniband, they do not give any guarantee that client nodes are kept in close proximity of each other.

11.4.5 Configuration Challenges

As well as providing clusters in clouds, it has to be possible to reconfigure the specifications of existing clusters for different types of cluster applications. The configuration of a cluster cannot become rigid, otherwise application developers will be forced to build the applications specifically for the cloud cluster and not the problem they intend to solve [636]. So far, the only such solution that allows a cluster to be dynamically reconfigured is the Adaptive

HPC Suite [603]. However, the solution is too course grained: it allows for the complete change of an installed operating system, not individual software libraries.

11.4.6 Publication and Discovery Challenges

If a cluster is deployed in a cloud, current information about the cluster has to be published in a well-known location. Otherwise, clients cannot find and later make use of the cluster. Currently, there is no cloud registry for clusters or other HPC resources. This is also an issue for other cloud and HPC related resources such as individual servers and data services.

Furthermore, clients should not be subjected to information overload. For example, if a client only wants a 10 node Linux cluster with 2 Gigabytes of RAM per node, the client should only be informed of clusters that satisfy those requirements and how. If the client does not care if the cluster is using Infiniband or Gigabit Ethernet, then the client should not have to be told about it. Overall, publication in clouds only varies between having static catalogs [61, 374, 632] to only tracking unique addresses of services [661].

11.4.7 Legacy Support Challenges

Often, the term legacy is used to describe software written years ago but still in use. However, it is possible for the legacy unit itself to be data generated from experiments years ago and stored using software systems current at the time. The challenge with such data is ensuring that it can be maintained in its original form (thus avoiding loss) but also making the data accessible to newer software (compatibility). In some cases, the data is too big to be converted to a newer format.

A similar challenge was faced with the CloudMiner [254]: the goal was to use cloud computing to provide data mining in an effective manner across heterogeneous data sources. The solution proposed was to have all data and data mining algorithms exposed via services that are published to well-known brokers. The end result is an environment where providers could publish their data and data mining techniques in a single, cloud based environment thus allowing clients to discover required data and run required algorithms over the data, irrespective of how the data is stored or where it is located.

11.4.8 SLA Challenges

One of the most difficult challenges is forming and maintaining agreements between clients and services. Computer systems (all types from standalone systems to clouds) are built using technological objectives. Even if the underlying computer system is the most effective at getting a given task done, it is ignored if it does not satisfy client (personal) requirements. For example, a database system may offer data storage using RSA encryption for strong security but the client (for whatever reason) may require Triple-DES. RSA is better but it is not what the client requires hence is not acceptable.

Besides satisfying client preference, mechanisms need to be in place to ensure that the terms outlined in the SLA at met for the life of the agreement and that violations of the agreement are properly addressed. For example, if the SLA states a job will be completed in two week but the job takes four, that is a violation, for which the provider must bear a penalty. Based on the current cloud offerings, SLAs only cover against loss of availability and state damages awarded (as seen with EC2 [62]). No other issues, such as security violations, are effectively covered.

While security and availability are presented in [733], possible resolutions to them are unconvincing at best. In terms of security, it is suggested that virtual VLANs (virtual, overlay, LANs that exist independent of physical LANs) could help improve security via traffic isolation. There is still the open issue of what international law applies when there is a security breach. However, when it comes to availability, [661] proposes that SLAs go beyond availability, need to consider QoS parameters and agreements properly formed between service providers and clients so that violations can be settled properly.

11.4.9 Workflow

The final challenge when creating HPC clouds is supporting client workflows. Not all HPC problems are processed using a single application; some problems require a chain of multiple different applications. The challenge with workflows is ensuring that each application in the workflow has all of its required resources before execution. Furthermore, it is unrealistic to assume that all resources will exist on a single cloud; the resources may be spread across separate clouds.

11.5 HPC Cloud Solution: Proposal

In response to the challenges presented in Section 11.4, this section proposes a new cloud solution called HPCynergy (pronounced HP-Synergy). The goal of HPCynergy is to provide an HPC oriented environment where HPC resources (from data storage to complete cluster environments) are accessible to clients as on-demand, easy to use services. While the HPCynergy solution aims to satisfy all properties presented in Section 11.3.1 the virtualization property is made optional for clients who have HPC applications that are sensitive to network/CPU latencies.

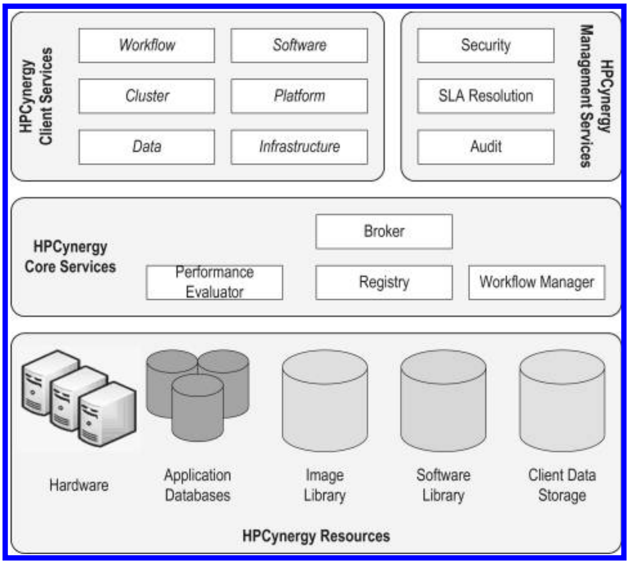


FIGURE 11.1
HPCynergy Structure.

11.5.1 HPCynergy Design

HPCynergy is designed as a collection of services, each service belonging to one of four broad groups. To address the Interface Challenge (Section 11.4.1) all services presented in this section have a Web API for software process clients and intuitive Web pages for human operators using Web browsers. Figure 11.1 shows the structure of HPCynergy with numerous resources at its base ranging from large hardware systems to various database services.

On top of the HPCynergy resources are the Core Services (Section 11.5.1.2): provided in response to the challenges of Publication and Discovery (Section 11.4.6), Configuration (Section 11.4.5) and Performance (Section 11.4.2). In relation to [733], the Cores Services are a realization of client on-demand resource provisioning and negotiation. Based on supplied requirements, clients learn what resources are available and decide for themselves which ones they will use. If matches cannot be found, clients retry with updated requirements.

On top of the Core Services are the HPCynergy Management Services (Section 11.5.1.3): provided in response to the Communication (Section 11.4.3) and SLA (Section 11.4.8) challenges. Security is also offered so that all clients only use services (and resources) intended for them.

Finally there are the HPCynergy Client Services (Section 11.5.1.4). The services provided come from either HPCynergy itself (such as cluster discovery) or services that have been provided by other HPCynergy clients. For example, when a HPC application completes and generates new results, the results can be kept in place in the Application Database thus supporting further projects.

11.5.1.1 HPCynergy Resources

HPCynergy provides a number of resources to users, which are described below:

Hardware: Hardware can exist as physical servers or virtual servers to support two scenarios for clients. First, a client may require a special purpose environment, hence will wish to start with raw (virtual) hardware and construct a suitable system from there. The other scenario is the client's requirements of an environment may be flexible, thus only a reconfiguration of existing services and resources (such as a pre-made cluster) are needed.

Application Databases: To support HPC applications, application databases (such as genome databases) are offered to any client. By providing the database the client can focus on the application to be run and avoid lengthy transmissions from remote locations to the cloud.

This element is similar to the Storage Virtualization feature in [733]. In that work, the focus was on providing a uniform means of data storage to support virtual machines — most likely how the virtual machines could be stored and how virtual hard drives could be provided to them. The HPCynergy realization is different — as well as providing a store of many application databases, there is a plan to offer the databases in a format clients require. E.g., if a client prefers SQL, all databases (regardless of type or format) are presented as SQL databases to the client. In effect, the aim is to provide the same uniform environment as CloudMinder [254].

Image Library: The Image Library contains numerous operating systems in both initial states (just the operating system installed and nothing else) and in pre-packaged states (have various combinations of installed software applications and libraries).

Software Library: The Software Library is provided so that HPC clients can create virtual machines and then either make use of or install required software to run their applications. In general, the Software Library keeps software as operating system specific installable (such as executable for Windows and packages for Linux). By keeping a library, clients will be able to install required software (along with any required dependent software).

Client Data Storage: The final element of the HPCynergy resources group is data storage for individual clients. When a client application is complete, the virtual machine needs to be disposed of so that the hardware resources can be released and used for new virtual machines. However, if a virtual machine is disposed, any data in it may be lost.

Thus, the Client Data Storage is provided so that clients can keep their data (results and log files) separate from their virtual machines. Furthermore, providing data storage to clients allows easy sharing and makes it possible for clients to share the results of the experiments with other clients. As with the

Application Databases, the Client Data Storage is a virtualization of data storage [733] which aims to provide all data in a required client format [254].

11.5.1.2 HPCynergy Core Services

Clients to HPCynergy are not limited to computing specialists needing to run an application on a (virtual) cluster. There is also the possibility where the client is a research team that intends to realize a new form of research environment. Thus such clients can be considered Service Providers and require facilities to make their new services known to other clients.

The purpose of the Core Services is to provide a set of generalized and reusable services to support the operation of HPCynergy and to help support clients who wish to provide their own HPC services to other clients. The goal of the Core Services is similar to the objectives behind CloudMiner [254]. CloudMiner allows any form of data mining algorithm to be carried out on any form of data source without any dependency between each other. HPCynergy allows any form of HPC application to be carried out using any HPC resource necessary without any dependencies between the two either.

Registry: The Registry allows for extensive publication of all services and resources (from databases to clusters) within HPCynergy. Having a registry system in a cloud is a significant feature. At the time this was written, all other clouds used catalogs for their services/resources instead of actual discovery systems ([61, 374, 632] to name a few).

Broker: Working with the registry, the Broker is a service that allows clients (human operators or other services within the HPCynergy) to find and select required services and resources [145]. Like the registry, such a service does not exist as yet in cloud computing hence its presence is a unique feature.

It could be considered that a Broker and Registry could be combined into a single service. While that may be true, the two services are separate to keep client request load low — high demand for the registry service will not affect attempts to discover, locate and arrange access to required services and resources.

Performance Evaluator: The Performance Evaluator service performs routine inspections on all services and resources within HPCynergy. By keeping current information about performance factors (considered quality of service parameters — QoS), it is possible to allocate high quality (thus high performance) resources and services to HPCynergy clients. Furthermore, other management services such the Workflow Management service can use the same information to avoid poor quality resources.

Workflow Management: Clients do not always require just a single service or resource. Clients may require a set of resources and services and (within that set) a required workflow. Like all resources and services, workflows also have requirements of their own: specifically, they all require a series of tasks to be carried out on specific services. Furthermore, each task within the workflow will have requirements on what data is passed in and what information comes out.

Thus, the Workflow Management service is a high level version of the Broker: the requirements going in will be the description of an overall workflow and the outcome is to be an orchestration that can execute the workflow within client specified limitations (for example, do not take longer than two weeks and do not cost the client more than $200).

At the time of writing, no adequate services for workflow management in clouds could be found. The closest solution found is the Moab Workload Manager [605] and even then it is ineffective, because jobs must be prepared before being ordered into a workflow.

11.5.1.3 HPCynergy Management Services

As with any service based environment, access to the services (and the resources offered by them) has to be monitored. Thus, a means of security has to be in place not just to prevent unauthorized client access to services, resources and data, but so that HPCynergy can comply with any regional laws regarding information security.

While good detail of management services/features are presented in [733], this chapter considers them in the Core Services group — they are fundamental to getting HPCynergy working. Thus in this group HPCynergy offers services that address difficult (and often nation spanning) issues.

Security: The role of the security service is to provide the three core elements of security: confidentiality — clients only access and use services, resources and data meant for them; integrity — data and communications between clients and services are not modified in transit; and authentication — to ensure that all entities (clients, services and resources) can have their identities verified.

SLA Resolution: If services and resources are to be presented to a client based on their requirements, then the parameters of the services when they were discovered need to be maintained. Thus, when services (and the resources behind them) are discovered and selected, a service level agreement (SLA) needs to be formed and enforced. That is the role of the SLA Resolution service. Once a SLA has been formed between a client and a service, any violation that occurs goes to the SLA Resolution service. This service seeks to settle disputes and carry out any penalty/compensation agreements outlined in the SLA.

This service is especially important when supporting client workflows. Workflows have very long lifetimes and involve numerous services and resources directly (may utilize them without services). Thus a single workflow will have numerous SLAs, one between each service for a given task and one for the complete workflow as a whole.

Audit: A counterpart to the SLA Resolver, the Audit service monitors all interactions with all services and ensures that clients pay for what they use. Another scenario addressed by the Audit service is where clients make use of services but fail to pay any fees. Thus, another role of the Audit service is to check clients wishing to make use of services in HPCynergy. Even if the client has security clearance, if the Audit service detects that the client has failed to pay for service use earlier, the Audit service blocks the client's access.

11.5.1.4 HPCynergy Client Services

In this group, the services offered are intended primarily for client usage and in turn make use of services in HPCynergy. As such the services in this category will vary from services provided by HPCynergy itself to services created and later offered by clients (often called reselling in business worlds).

In general, services in this collection are intended to provide high level abstraction for otherwise complex specialized tasks. For example, to ease clients in their discovery of a required cluster (or reconfiguration of an existing cluster), infrastructure-like services are placed in this category so that clients only focus on what they require in a cluster and all other resources and services are arranged for them.

Infrastructure: Infrastructure services offer functionality to clients so that systems from standalone virtual machines to complete cluster arrays (without software) can be created. For example, if the client requires a 20 node cluster, each with a single core, a service can exist to find physical servers on the cloud to host 20 virtual machines and run an appropriate hypervisor.

Platform: Like Infrastructure services, platform services offer complete hardware and software environments. For example, a client may require a 20 node Linux cluster but does not care how it is built. Thus a platform service can be presented that offers prebuilt clusters and/or stand-alone (virtual) machines.

Software: Software services are focused on offering required HPC applications and other support services to clients. For example, if a client only needs access to mpiBLAST and the delivery is irrelevant, mpiBLAST can be exposed as a service and offered in this area of HPCynergy.

Another example is a software compiler: instead of burdening clients with the task of installing and configuring all required software libraries, the task can be encapsulated in a service instead. All that is required of the client is the source files and the compiler input parameters.

Data: Data services are provided so that data sources in HPCynergy can be seen by clients as huge, homogeneous databases. For example, Client A will see the HPCynergy will all data as a relational SQL Database while Client B will see the data as XML files. Furthermore, if a version of CloudMiner [254] existed for HPCynergy, this would be the most likely location to place it.

Cluster: Cluster services are provided to ease the discovery, selection, and use of clusters. If a cluster cannot be found, the cluster services can attempt to make a required cluster using the Management Services (Section 11.5.1.3).

Workflow: Similar to the software category, if a research team finds an effective means of using existing services to quickly solve a problem, the workflow

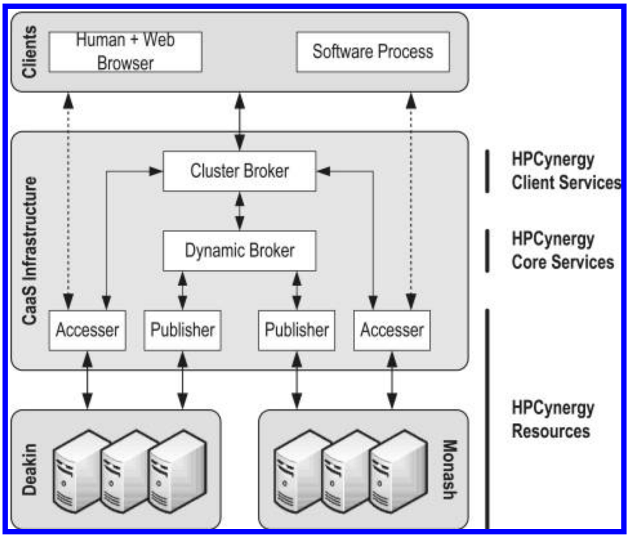


FIGURE 11.2
HPCynergy Implementation Stack.

used itself is a valuable contribution thus should be made accessible as a service.

11.5.2 HPCynergy Implementation

HPCynergy is implemented using two clusters and new Web service-based middleware called Cluster as a Service (CaaS) Infrastructure. Figure 11.2 shows the implementation stack of HPCynergy in relation to clients and the two clusters. Annotations on the right indicate where the current services relate to the HPCynergy design in Section 11.5.1.

As this is the first implementation of HPCynergy, the focus was mostly on easing the publication and access to existing clusters. The reason for this is that if clients cannot learn of existing resources (in this case, a cluster) then it does not matter how well managed or secure or reliable the resources are as the client will not know of then thus will not make use of them.

With respect to the features shown in [733], the current implementation aimed to have the features of virtualization support, providing services and resources on demand, dynamic allocation and reservation/negotiation.

The Deakin cluster is comprised of both physical and virtualized nodes. In terms of hardware, the cluster has been constructed using 20 servers, each with two quad-core Xeon processors. Of these 20 servers, 10 have VMware vSphere [730] installed thus allowing them to run virtual machines instead of a single operating system. Subsequently, 20 virtual machines (each with 6 virtual cores) have been created thus (theoretically) forming a 30 node cluster. All nodes (virtual and physical) run the CentOS operating system and have Sun Grid Engine and Ganglia installed.

The Monash cluster is a physical cluster of 36 nodes — 20 nodes are the same as the Deakin Cluster in hardware and software, and the remaining 16 nodes have the same software but instead have two dual core processors each.

In terms of virtualization support [733], the Deakin cluster supports this feature. However, the focus is on using virtualization to gain more efficiency from the cluster — virtual machines can be migrated/rearranged so that busy virtual machines can be dedicated whole physical systems, while idle virtual machines can be consolidated to single physical nodes. There was no focus on clients being able to create and maintain their own virtual machines. Also, while there is mention of supporting multiple hypervisors, that was not considered this in the HPCynergy implementation. First, a single type of hypervisor to simplify testing was kept. Second, there is yet to be the invention of a cloud that uses more than one form of hypervisor.

In terms of dynamic resource allocation, the best that is in the current implementation is that offered by individual cluster schedulers — cluster jobs start off in a queue and are executed when enough nodes become available to be allocated to them.

To make the clusters accessible to clients, the Cluster as a Service (CaaS) Infrastructure solution is used. An evolution of the previous CaaS Technology [146], the CaaS Infrastructure provides an environment where numerous clusters can be easily published by their providers and later easily discovered and used by clients.

Clusters offered via the CaaS Infrastructure are coupled with a pair of services: a Publisher Service that makes current and detailed information about clusters known and an Accesser Service that allows clients to use the related cluster in an abstract manner. In relation to the HPCynergy design, the clusters along with their Publisher and Accesser Services belong to the Resources group (Section 11.5.1.1).

To make information about the cluster known to clients, the Publisher Services forward all information to a discovery system called the Dynamic Broker [145]. In relation to the HPCynergy design, the Dynamic Broker is an implementation of both the Registry and Broker services (Core Services, See Section 11.5.1.2).

Finally, clients discover and select required clusters through a high level service called the Cluster Broker service — a simplified form of the Dynamic Broker that specializes in taking client requests, encoding them into attributes for use with the Dynamic Broker and then displaying result information in an easy to read manner. After learning and selecting a required cluster via the Cluster Broker service, clients approach the related Accesser service and make use of the cluster. In terms of the HPCynergy design, the Cluster Broker service belongs to the Client Services group (Section 11.5.1.4).

In relation to [733], the Cluster Broker service exhibits the Reservation and

Negotiation Mechanism — albeit manual. In terms of negotiation, clients are presented with search results and decide for themselves which cluster is suitable for their needs. If a required cluster cannot be found, the only negotiation possible is where clients update their requirements and run the search again. In future work, it is planned to create additional services to complete the HPCynergy Core Services group and implement the Management Services group. The Client Services group will evolve as more applications and ideas to HPCynergy are applied.

11.6 Cloud Benchmark of HPC Applications

As seen in Sections 11.3 and 11.4, cloud computing is an emerging technology and there are many concerns about the security, performance, reliability and usability. Despite this, the advantage of on-demand computing is beneficial when solving research problems that are data heavy and/or computationally intensive. It is this adoption of cloud computing by the scientific community that has paved the way for HPC clouds (such as HPCynergy described in Section 11.5). Performance of HPC clouds is important due to the nature of scientific problems. This section uses benchmarking to compare the usability and performance of cloud computers and dedicated HPC machines.

11.6.1 Tested Applications

To investigate the feasibility of running HPC applications on HPCynergy, and other clouds, two practical applications were tested. The first application is an embarrassingly parallel system biology pipeline developed at Deakin University (Section 11.6.1.1). The second application is called GADGET; a communication bound N-body physics simulation (Section 11.6.1.2).

11.6.1.1 System Biology Pipeline

Bioinformatics studies complex biological systems by applying statistics and computing to genomic data. While the first human genome was sequenced at a cost of $2.7 billion dollars improvements to sequencing chemistry has reduced the price to $10,000 [256]. As a result bioinformatics has seen a large increase in available data. It has been estimated that in 2 years the cost of sequencing will be $1000 per human genome, thus making it economically feasible to sequence large amounts of human and other complex organisms.

Collecting large amounts of genomic data has a number of important implications in cancer treatment and medicine, in particular through personalized cancer treatments. These treatments rely on first identifying cancer subtypes which can be found or diagnosed by building system models [255], which show

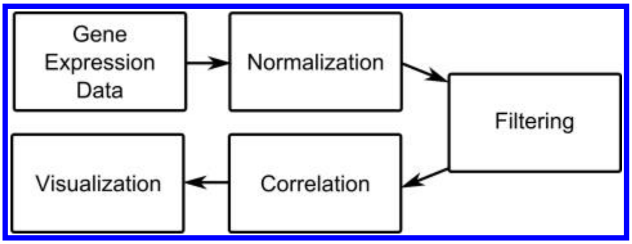
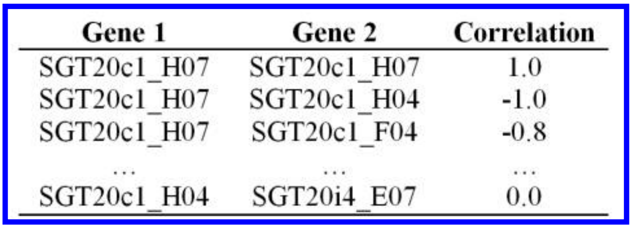


FIGURE 11.3
A Common System Network Workflow.



Gene 1	Gene 2	Correlation
SGT20c1_H07	SGT20c1_H07	1.0
SGT20c1_H07	SGT20c1_H04	-1.0
SGT20c1_H07	SGT20c1_F04	-0.8
...	...	...
SGT20c1_H04	SGT20i4_E07	0.0

FIGURE 11.4
An Example of the Simple Interaction Format.

the interaction of genes in a biological system. Building system models is a multi-step process (see Figure 11.3), consisting of; normalization and filtering of data, statistically correlating genes, and then visualizing results. Of these steps, correlating genes can be the most time consuming; a list of N genes requiring N correlations to be made for each gene.

This system biology workflow has been implemented using a combination of R [372] and C++. In this implementation cDNA array gene expression data is first normalized using a cross channel, quantile-quantile approach. Filtering is then used to remove noise and find significant genes. After filtering, a distributed implementation of the Pearson's correlation algorithm [351] is used to find relationships between genes. Data is outputted in Simple Interaction Format (SIF) [170], a tab delimited text format used to store network data (see Figure 11.4). Correlated gene data is then visualized as a network using Cytoscape [257].

The following benchmark utilized gene expression data collected before and during the Tammar Wallaby lactation period. Eight observations taken over this lactation period were first normalized and then filtered. Remaining genes were then correlated using Pearson's R. Genes which were strongly correlated were visualized (Figure 11.5).

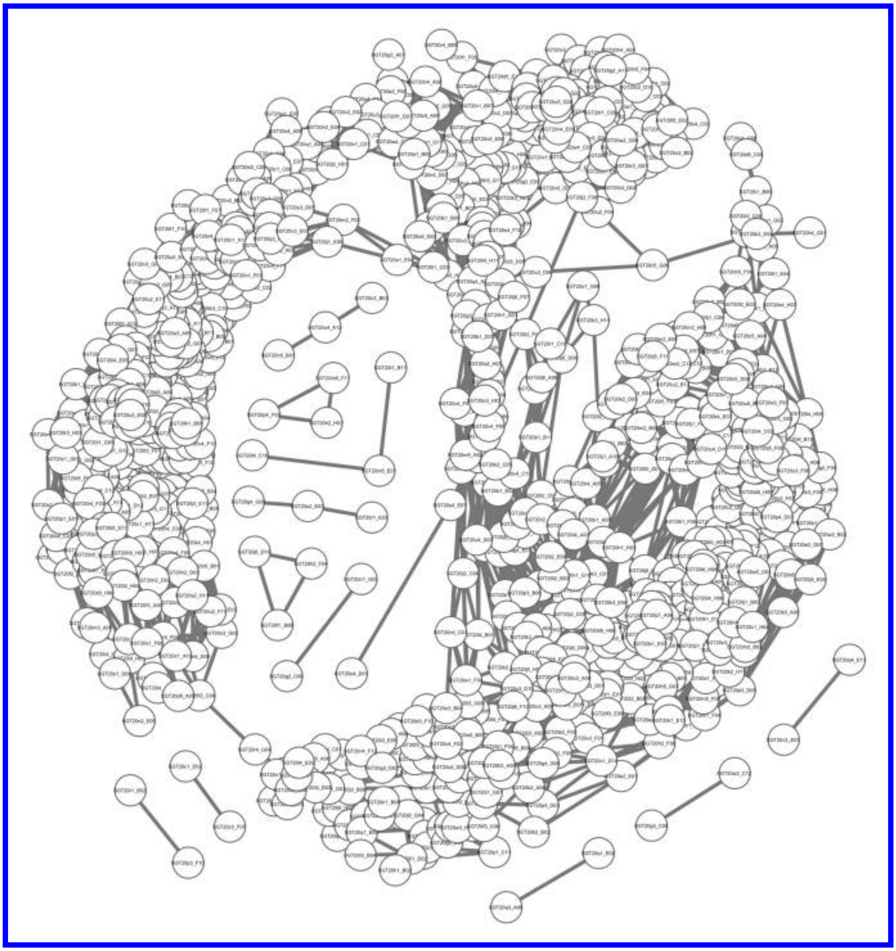


FIGURE 11.5
Network of Genes Expressed during Tammar Wallaby Lactation.

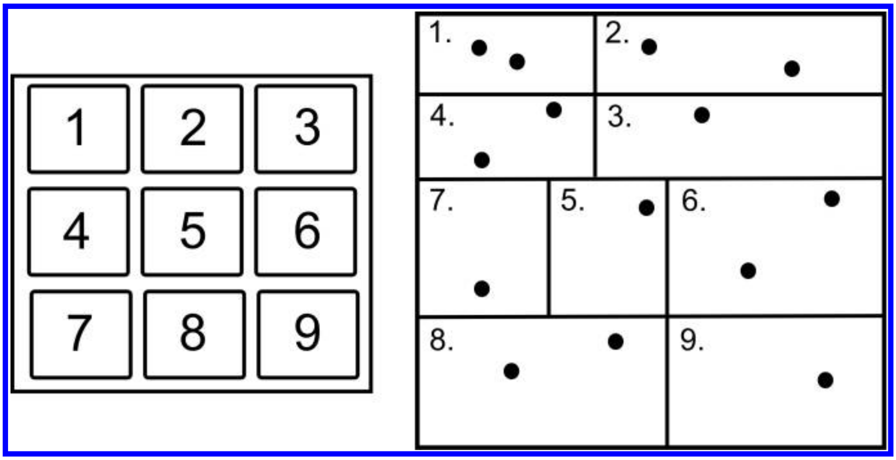


FIGURE 11.6
Example of Space to Processor Mapping during an N-Body Simulation.

11.6.1.2 GADGET: Communication Bound N-body Simulation

Like bioinformatics, physics also contains many computational intensive problems. Particle accelerators such as synchrotrons and the Large Hadron Collider (LHC) generate large amounts of data. The LHC smashes bundles of Quarks (the smallest known particles at this time) together close to the speed of light in the hopes of finding evidence for the existence of several theoretical particles, such as the Higgs boson and those predicted by Super Symmetry [576]. During these experiments terabytes of data are collected using multiple sensors which then must be verified by comparisons to particle simulations based on current theories [652].

The implementation of particle simulations, also known as N-body simulations, is diverse. A common implementation method, called physical mesh, maps compute nodes to physical space [110]. As particles move and collide in simulated space, particle data is also distributed to the node which simulates that space. This transfer of particles between nodes is facilitated by message passing. Because communication is a key requirement it is common for logical space to be ordered in a similar way to physical space (Figure 11.6). By ensuring communication is between physically close nodes, communication speed is made as fast as possible. How this mapping is accomplished depends on the hardware and N-body algorithm used.

GADGET [672] is a well-known astrophysics application designed to simulate collision-less simulations and smoothed particle hydrodynamics on massively parallel computers. This program was used to run a simulation of two colliding disk galaxies for a total of 10 simulation time-steps. To ensure a standard level of accuracy, particle positions were calculated at 0.025 current Hubble time

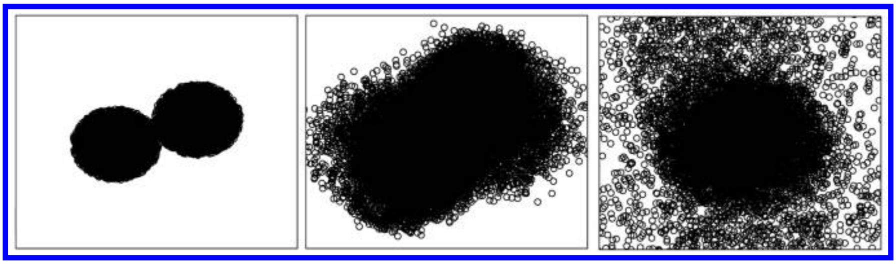


FIGURE 11.7
Visualization of Two Disk Galaxies Colliding.

intervals. Snapshots of particles were written out as binary files every 0.1 simulation time-steps, some of which have been visualized using R (Figure 11.7).

11.6.2 Benchmarked Platforms

Two physical clusters and three clouds were used during the benchmark study. Naming conventions of the machines are as follows: each cluster is referred to by network interface (InfiniBand Cluster, Ethernet Cluster) and each cloud is referred to by the cloud management interface (vSphere [730], Amazon, HPCynergy). In terms of hardware, these computer platforms were chosen to be as similar as possible to each other; even when possible utilizing the same hardware. Of the five systems described below, vSphere and the InfiniBand cluster run on subsets of the Deakin Cluster[4](described in Section 11.5.2), while the Ethernet and Amazon machines use their own dedicated hardware. Despite the large effort taken to minimize hardware differences, the Ethernet and some Amazon instances differ in the amount of cores per processor. Because of this variation, each process was mapped to a single core and when possible a single node. To validate the mapping process, CPU usage was monitored during data collection, for example a dual core system with a single process would be using 50% capacity. This methodology was chosen because it is similar to that used by the clouds, in that virtual machines are mapped to physical hardware.

The InfiniBand cluster used in this benchmark consists of the physical nodes on the Deakin Cluster. This cluster is a bare-metal system consisting of 10 nodes, each with an Intel Quad Core Duo processor running at 2.33 GHz. Each node utilizes 8 GB of RAM and runs a 64-bit version of CentOS. As a cluster dedicated to HPC, nodes are connected using 10 GB InfiniBand, and a

---

[4] Testing has been on the Deakin cluster directly and (later) through HPCynergy. This is to assess the claim that using middleware with high levels of abstraction results in clusters being unusable due to poor performance.

mounted network drive allows users to easily setup MPI applications. In terms of CPU speed and RAM, this cluster is equivalent to the documented specification of the large Amazon Instance but differs by having a faster network interconnect.

The Ethernet cluster used in this benchmark is also devoid of virtualization and is equivalent to the documented specification of the small Amazon instance. This four node cluster was constructed using Intel Dual Core computers running at 1.6 GHz each with 2 GB of RAM. A 32-bit version of Ubuntu 9.10 was used as the OS. Compute nodes were connected by a low I/O Ethernet network (1 Gb/sec).

Three Amazon instance types were tested; small, large and cluster. It has been documented that Amazon uses a modified version of Xen as the hypervisor [59]. In each case the Amazon Elastic Block Store (an Amazon service which provides persistent storage of virtual hard-drives) was used to store the state of the deployed virtual machines. Each instance type differed in CPU, RAM and network I/O. Amazon measures the performance of CPUs in Amazon Compute Units (ACUs), this is equivalent to an Intel Xeon chip.

Each Amazon small compute instance contains 1 ACU and 1.7 GB ram. Connection between Amazon small instances is documented as low I/O [61]. The large instances contain a dual core CPU (each with 2 ACU of power) and 7.5 GB of RAM. Connection between Amazon large instances is documented as high I/O [61]. The Amazon Cluster Compute Instances is the best defined, these machines contain two Intel "Nehalem" quad-core CPU running at 2.98 GHz and 26 GB of RAM [61]. Connection between cluster instances uses a 10 GB Ethernet connection. The small and large instance types were used to setup a 17 node cluster; however, the allocation of the Cluster Compute Instance was capped at 8 nodes.

The second cloud used in this benchmark utilized VMware vSphere. This private cloud runs on the physical nodes of the Deakin Cluster. A ten node virtual cluster was deployed through this VMware cloud, each with dual core processors running at 2.33Ghz. A 10 GB InfiniBand network was used to provide inter-node communication.

The final cloud used in this benchmark is HPCynergy (see Section 11.5). This cloud platform exposed the whole Deakin Cluster through the underlying CaaS Infrastructure. A total of seventeen compute nodes were utilized through HPCynergy, each node containing a hexa-core processor running at 2.33 GHz[5]. A 10 GB InfiniBand network provided inter-node communication.

Specifications of all platforms used in the benchmark tests are summarized in Table 11.1.

---

[5] Only virtual nodes were utilized during the benchmark to enable comparisons between HPCynergy and the vSphere cloud.

TABLE 11.1
List of Computer Platforms Broken Down by Specifications.

Name	Nodes	Hyper-visor	Platform	Hard Disk	CPU	RAM	Network
Amazon Cluster	8	Modified Xen: HVM	64-bit CentOS	Elastic Block Store	2 x Intel quad-core Nehalem (2.93GHz)	23GB	10Gb Ethernet
Amazon Large	17	Modified Xen: Para-virtual	64-bit Ubuntu 9.10	Elastic Block Store	2 x 2007 Xeon equivalent (2.2GHz)	7.5GB	High I/O
Amazon Small	17	Modified Xen: Para-virtual	64-bit Ubuntu 9.10	Elastic Block Store	2007 Xeon equivalent (1.1Ghz to 1.6GHz)	1.7GB	Low I/O
vSphere Cloud	10	VMware	64-bit Ubuntu 9.10	Separate Drives	2.33Ghz Intel Dual Core	2GB	10Gb InfiniBand
Infini-Band Cluster	10	None	64-bit CentOS	Shared Drive	2.33GHz Intel Quad Core Duo	8GB	10Gb InfiniBand
Ethernet Cluster	4	None	64-bit Ubuntu 9.10	Separate Drives	1.6GHz Intel Dual Core	2GB	1Gb Ethernet
HPC-ynergy	30: 10 physical 20 virtual	VMware	64-bit CentOS	Shared Drive	Virtual-Hexa-cores (2.33GHz) Physical-Dual Quad Cores	8GB	10Gb InfiniBand

11.6.3 Setting up the Clouds: Methodology

Setting up computer resources for HPC is a time consuming task and often serves to interrupt research. While the Ethernet and InfiniBand clusters used in these benchmarks could be used once code had been compiled, the Amazon and vSphere clouds required modification to enable HPC. The HPCynergy cloud solution aims to reduce setup time by exposing systems which have middleware already setup. The modification scope has been defined by both the application and cloud infrastructure.

11.6.3.1 General Setup

Before benchmarking could occur, each cloud required a number of steps including: (i) transferring data and source code, (ii) configuring the dynamic linker, (iii) compiling the source code and any dependencies, (iv) configuring the sshd client, (v) generating public and private keys, (vi) passing public keys to all nodes and (vii) creating a machineFile for MPI.

In addition to the above steps, each cloud had limitations which required additional setup time. The vSphere system did not contain any VM templates thus installation of the Ubuntu OS was required before operation. While all Amazon E2C instances used in these benchmarks did not have common utilities such as the g++ compiler, the g77 compiler, vim or zip, setting up these applications required additional time. Once each system was setup, transfer of benchmark specific data and compilation of necessary software was required.

11.6.3.2 System Biology Pipeline

A number of programs were used during the bioinformatics benchmark; filtering and normalization was performed using C++ and correlation utilized the R runtime environment (see Section 11.6.1.1). The data and source code was zipped and transferred through the scp utility; once on the target machines, the source code was then configured and compiled. The InifiniBand cluster used a shared drive and therefore compilation was only required once. Set up of both the Amazon and vSphere cloud systems was simplified through use of virtualization; a template containing the necessary software and data was created and then cloned. The Ethernet cluster made use of separate drives but because each node was homogeneous to each other (in hardware architecture and software versions) each could use the same compiled binary.

Figure 11.8 shows the necessary setup times for each platform; these times are based on best case scenarios and do not take into account problems occurring during compilation. It should be noted that this is not usually the case. Common problems include missing configuration augments, missing library dependencies, and compiler specific code. If any of these problems occur compilation has to start over.

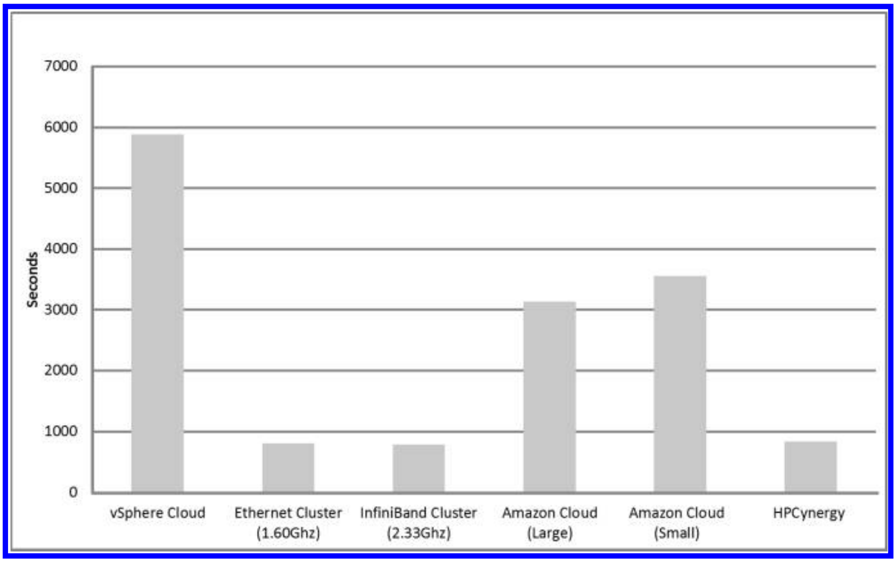


FIGURE 11.8
Total Setup Time of the 4-Node System Biology Benchmark.

11.6.3.3 GADGET: Communication Bound N-body Simulation

The physics application GADGET (Section 11.6.1.2) required a large number of dependencies. These dependencies are as follows:

- OpenMPI for distribution of GADGET across nodes,
- GSL - a general scientific library for C++,
- FFTW - an MPI implementation of the Fourier transformation, and
- HD5F - a binary output format.

As in the bioinformatics setup, data and source code was zipped and transferred through the SecureCopy (scp) utility from a local machine. Dependencies were compiled from source code on each target machine and GADGET was setup for HD5F output and FFTW double floating point support. Reported setup time (see Figure 11.9) is based on best case scenario and was further minimized by use of cloning or shared drives. Despite these advantages each cloud system required on average 7 hours to setup, while the cluster setup time was less than an hour.

11.6.3.4 HPCynergy Features and Setup

When setting up HPCynergy for benchmarking, setup time was minimized due to its unique interface. Like other clouds, HPCynergy monitors and acts as a broker to linked (physical and virtual) hardware. However, instead of hiding

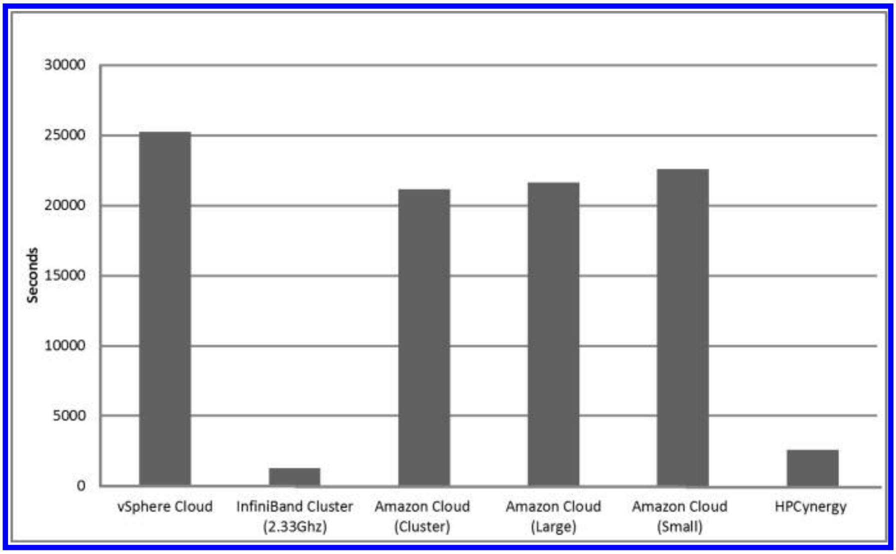


FIGURE 11.9
Total Setup Time of the 17-Node Physics Benchmark.

the state and specification of hardware from the users, the opposite approach is taken. Users are informed of the software and underling (virtual) hardware specifications of each machine. This allows jobs to be optimized to the CPU architecture as well as minimizing the need to install specific libraries.

When compared to other clouds (see Figure 11.10) the HPCynergy solution has a reduced setup time. This is due to a Web interface which allows users to search for a cluster that contains enough resources to support their job. Once a user has selected a cluster, minimal user requirements are used to configure an underlying scheduler — such as the number of required nodes. On submission of a job, a user provides only a bash script which oversees the whole cluster job (workflow) and any data files they wish to upload. While compilation of application level software is sometimes necessary, cluster middleware is already setup (for example, schedulers and libraries such as OpenMPI). The advantage of this method is that a user has only to configure and start the applications.

11.6.4 Benchmarking

Comparisons made between collected results highlight the advantages and weaknesses of utilizing specific cloud platforms for high performance scientific computing. HPCynergy addresses many of these weaknesses (see Section 11.4) through a combination of Web services and easy to use Web forms, but benchmarking is necessary to prove that HPCynergy is feasible with regard to performance. It is often claimed that providing ease of use or higher levels of

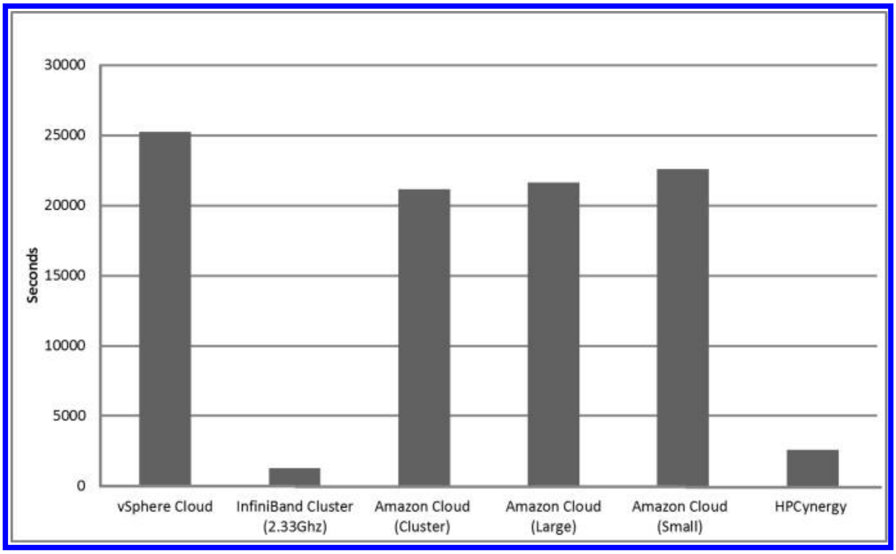


FIGURE 11.10
Total Setup Time of the HPCynergy System.

abstraction results in poor performance — despite the fact that performance and ease of use are two unrelated fields. To test performance, the system biology pipeline (Section 11.6.4.1) and GADGET application (Section 11.6.4.2) were run on a number of commercial cloud solutions, dedicated clusters, as well as (virtual) nodes discovered and used via HPCynergy.

11.6.4.1 Bioinformatics Benchmark

Performance of the system biology pipeline (described in Section 11.6.1.1) was recorded from five machines, the small and large Amazon virtual clusters, the private vSphere cloud, the Ethernet cluster and the InfiniBand cluster. As stated in Section 11.6.2, the Ethernet cluster is equivalent to the Amazon small instance and the InfiniBand cluster comparable to the Amazon large cluster and vSphere cloud.
Once set up according to the methodology specified in Section 11.6.3, the state of the virtual machine was saved as an Amazon Machine Image (AMI). Creation of this AMI allowed easy scaling of the bioinformatics application through virtual machine cloning. Access to each machine used in this benchmark was achieved remotely through an SSH client. Only one process was run on each node in order to minimize overhead of any other operating system processes. Performance results for each machine were measured up to four nodes; each test was run three times in order to ensure the validity of results. As seen in Figure 11.11, the results show a nearly linear increase of perfor-

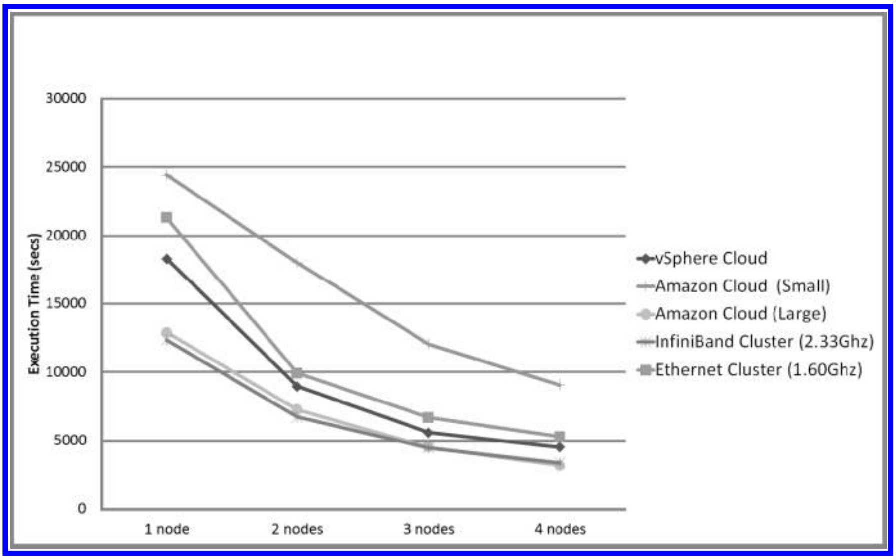


FIGURE 11.11
Building a System Network on Different Computer Architectures.

mance to available resources; this is expected, as most of the system network workflow is embarrassingly parallel. When compared to physical hardware, the vSphere cloud shows a noticeable increase in required computational time. It is likely that this increase is due to virtualization overhead, in which part of the CPU is constantly being delegated to simulate the specified environment. This virtualization overhead leads to an interesting relationship where the smaller a job is, the closer cloud performance will match physical hardware performance. These results also highlight that different hypervisors and cloud service implementations also affect performance. Performance of Amazon which uses a modified Xen hypervisor is very close to physical hardware, while the vSphere cloud which makes use of VMware virtualization suffered the most overhead.

From a user view, setting up a cloud for HPC is time consuming. If non-persistent storage is utilized, setup must occur every time a job is run. In embarrassingly parallel applications, data and code must first be transferred over the internet to the cloud. Depending on the size of problem or result data, a considerable delay is required before a job can be started. In the case of this system biology workflow, the delay of retrieving results from the cloud is an even bigger problem, the correlated data becoming many times larger than the original gene set.

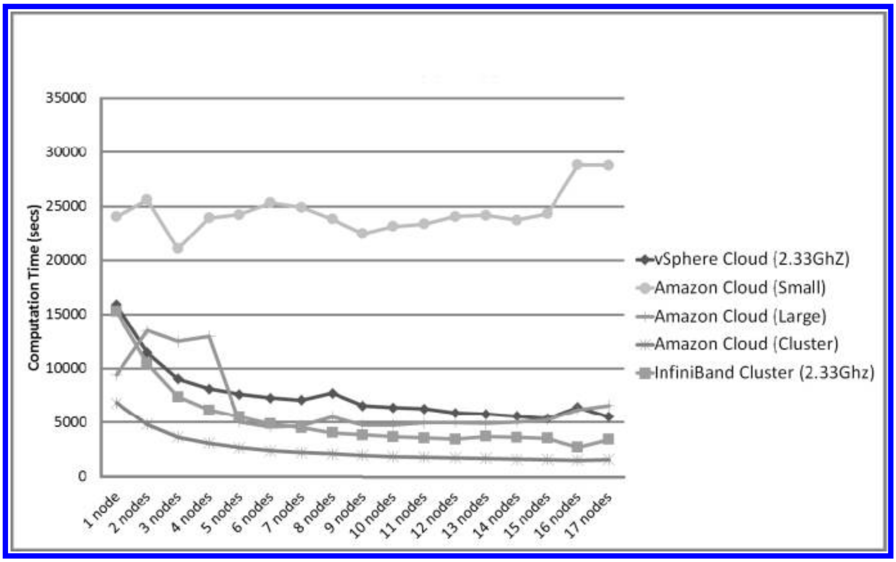


FIGURE 11.12
Simulating Particles over a Range of Computer Architectures.

11.6.4.2 Physics Benchmark

As in the bioinformatics example, many cloud architectures were compared, each running similar virtualized hardware. The platforms utilized in this application study were the small, large and cluster Amazon E2C clouds, the private vSphere cloud and an InifiniBand cluster. Benchmarking made use of full machine capacity, tests running on up to 17 nodes. Each Amazon machine was based on template AMI created during the setup process and started via the Elastic Block Store. Access to each machine used in this benchmark was achieved through an SSH client.

To view the effect of network speed, only one GADGET process was run on each node. Each point was run three times in order to ensure the validity of results. The results from this GADGET benchmark study can be seen in Figure 11.12.

The physical hardware results represent the ideal performance of this study, a near constant computational decrease as more compute nodes are added. The vSphere cloud, which runs on the same hardware, shows this relationship but with a similar offset as seen in bioinformatics study (Section 11.6.1.1).

Performance of the Amazon EC2 cloud varies depending on the instance type chosen. The small instance shows a sharp computational increase at 2 nodes before performance becomes optimal at 3 nodes. The large instance with higher I/O shows a similar early computational spike before optimizing at 5 nodes. Both the small and large EC2 cloud instances show an increase in computation time as more nodes are added past this optimal performance threshold. This relationship is an indication of a communication bottleneck, where each node is spending more time communicating than processing.

Amazon recently added a Cluster Compute Instance which has been optimized for running computation heavy applications. The performance of this instance shows a decrease in execution time mirroring other high speed clusters. This optimal performance is dependent on allocating cluster instances at the same time. Because of this requirement the user loses one of the biggest draws to the cloud, the ability to elastically scale their applications.

Unlike the system biology problem presented in Section 11.6.1.1, this N-body algorithm requires communication between nodes. Collected results from Amazon show that performance is not necessarily linked to the amount of machines used. When running communication based applications, it is important that load is balanced between nodes and that communication is minimized. If each node is communicating more than it is processing, the computation time increases as resources are added. Cloud computing resources are highly distributed and performance of communication heavy applications can vary depending on the network architecture and the location of machines that have been allocated to the user.

11.6.4.3 Bioinformatics-Based HPCynergy Benchmark

Performance assessment of the execution of bioinformatics application on HPCynergy was carried out when building the lactation system network visualized in Figure 11.5. In this test jobs were submitted to the virtual nodes of the Deakin Cluster (as described in Section 11.6.2). Each job submitted to this system utilized only a single core of each node. Through this methodology it was hoped that overhead of the operating system and CPU memory caching would be minimized. As in the previous tests, the bioinformatics benchmark was run on up to four virtual nodes.

Results shown in Figure 11.13 indicate that HPCynergy scales nearly linearly like all other computers tested in this benchmark. While not performing as well as physical hardware or the Amazon cloud, HPCynergy incurs a smaller overhead than the vSphere interface. On average HPCynergy performance results were 5% faster than vSphere. This saving can be significant when running large-scale distributed applications. In this scenario, utilizing the HPCynergy solution simplifies submission of HPC applications while reducing performance overhead (when compared to other clouds).

11.6.4.4 Physics-Based HPCynergy Benchmark

Results were also taken while running the N-body application on HPCynergy. Like the previous benchmark, jobs were submitted to the virtual nodes of the Deakin Cluster (as described in Section 11.6.2). Up to seventeen nodes were utilized during this physics benchmark

Collected results (see Figure 11.14) show a similar decreasing trend to the

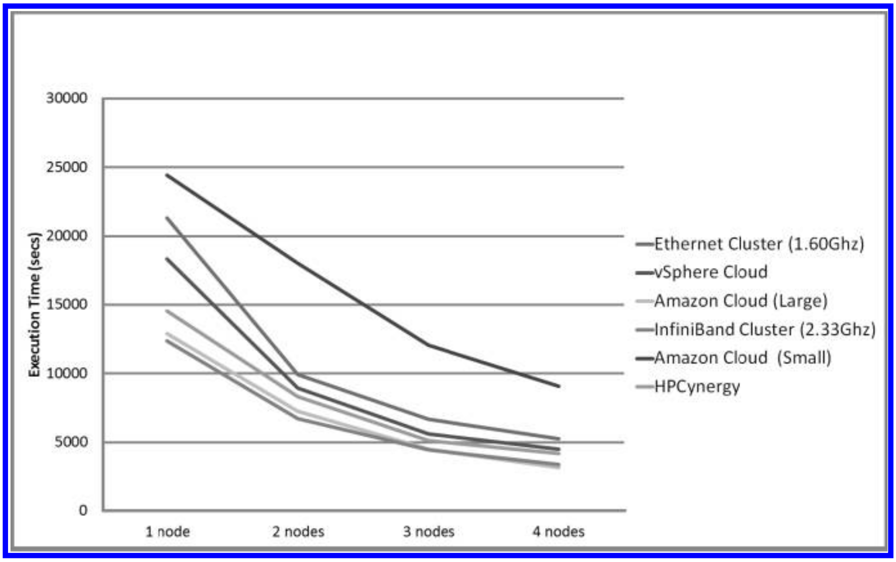


FIGURE 11.13
Performance of System Biology Workflows on HPCynergy.

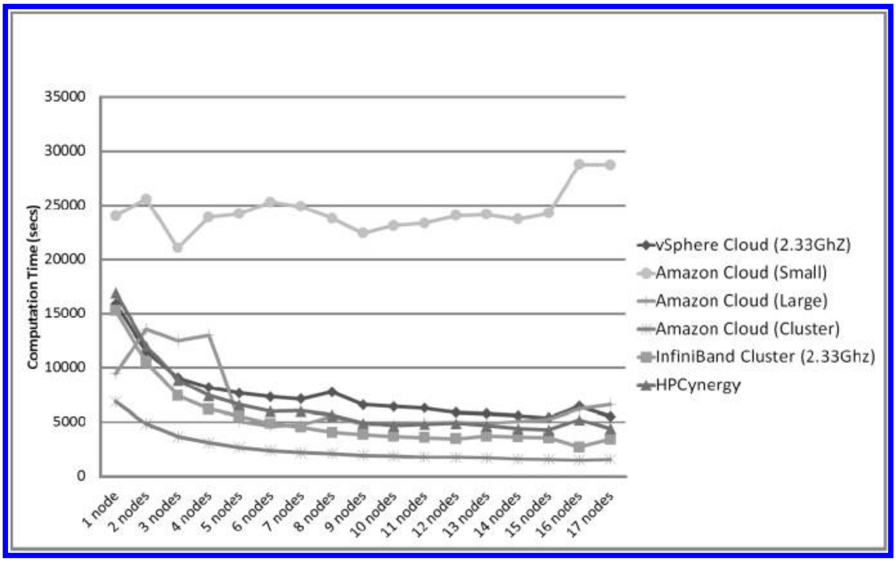


FIGURE 11.14
Performance of N-Body Simulations on HPCynergy.

other cluster utilizing high speed networks. When compared to the vSphere cloud, an average performance improvement of 16% is observed. The simple interface of HPCynergy allows for this improved performance, but it is not streamlined enough match the performance of the physical hardware. In conclusion, HPCynergy is a viable alternative to other cloud platforms providing both improved usability and performance

11.6.4.5 Conclusion

The results in this section show that even standard clouds can achieve performance similar to that of dedicated HPC clusters, depending on the class of problem. When running embarrassingly parallel applications a near linear speed up is achievable. Collected results show clearly that the effects of virtualization vary with the type of hypervisor used. The use of Xen seems to have minimal performance effect on computation while VMware is noticeable. This overhead can be reduced through the addition of multiple nodes, becoming insignificant once computation is reduced to 3 hours. When running communication bound applications performance results vary. On the clouds with slow network speeds the n-body application achieved maximum performance early on, past this point the required compute time steadily increased due to communication overhead. The three clouds with HPC hardware (Amazon Cluster Compute Instance, HPCynergy and VMware vSphere) showed the same decreasing performance trend as the InfiniBand cluster. These performance results indicate it is feasible to run communication bound applications only when cloud providers make use of HPC dedicated hardware.

The usability and setup time of cloud computers is another major issue. Setup time can be greatly reduced through use of pre-existing templates. Templates can be difficult to utilize as they are often missing common dependencies (compilers, text editors, etc.) and may have a range of security access setups. Amazon makes use of a Machine Image repository sortable only by operating system, architecture and image name. Using this search interface it is difficult to know what software each image contains until an instance of the template is launched. Ideally when launching an Amazon cluster instance a message passing interface such as MPI and common compilers should be pre-compiled; when using the recommended cluster compute image this was not the case.

HPCynergy addresses issues in cloud performance and usability, providing an easy to use job submission interface and resource monitor. In both benchmarks the HPCynergy cloud platform showed a scaling trend similar to that of dedicated HPC clusters. When compared to the vSphere VMware cloud, HPCynergy showed an average performance improvement of 10% but did not match the physical hardware. It is hypothesized that the remaining performance lag is due to the hypervisor, and further improvement could be made if a Xen hypervisor was utilized instead. In conclusion, the HPCynergy platform addresses a number of HPC cloud challenges while improving computational performance through reduction of management overhead.

Along with the benchmark results, the HPCynergy design is a significant contribution: it is the first (and possibly, only) existing design that encompasses many issues faced when using cloud computing to supplement or fully conduct HPC research. All issues from required resources to publication and discovery to management to (finally) high level simplifications have been presented and full details given on how all elements relate to each other.

The current HPCynergy implementation is the first step to realizing a promising cloud environment for HPC: where clients can quickly learn of and make use of existing clusters with minimal effort and do not have to spend significant amounts of time installing additional software. It has even been found through benchmarking that the CaaS Infrastructure, employed by the HPCynergy, has little to no performance penalty when running HPC applications. In effect, HPC clients are able to get the turnaround time of application results without the installation and setup overhead.

11.7 Conclusions and Future Trends

The main advantage of utilizing cloud computing is that it allows users to scale resources to their problem without investment in hardware. A key example of this is using clouds to handle increased traffic of web sites and web applications during peak times. When running high performance applications on clouds, hardware requirements are different (high speed network interconnect, RAID drive for parallel read and writes, etc.).

Furthermore, the use of virtualization incurs an overhead and makes it difficult to optimize code at the hardware level. With these weaknesses, running HPC applications on clouds has thought to have been unfeasible [345].

This chapter demonstrates that HPC clouds have a bright and promising feature. The combination of cloud computing elasticity, pay-as-you-go payment, and flexibility through virtualization offer significant opportunities in accessing HPC infrastructure and furthering collaboration with other research teams/projects.

This chapter focused on what challenges exist when using cloud computing to support HPC and in response proposed and demonstrated via HPCynergy the possibility of accessing HPC clusters as on demand utilities. Despite what many experts have claimed, no significant overhead when using cloud for HPC was found. Through the tests with Amazon, various configurations of cluster, and HPCynergy, it was found that (given a high speed network interconnect) the performance of HPC application in clouds is comparable to that of physical HPC clusters.

There are, however, other significant issues that still restrict the use of cloud for HPC. First, performance - there is no means to measure the quality of service (or whole clusters) within clouds nor can such information be made known.

There is also the issue of security - just as how some organizations may have obligations to the privacy of result data, so too do research project. Besides obligations, security is vital so that malicious users cannot access and destroy research results. While cloud computing can make HPC more reachable to teams that cannot afford the costs of clusters, it also makes it more accessible to malicious users who have access to the same cloud.

The last problem of interest is interoperability between HPC clouds. With the volume of result data generated by research projects, there is a strong need to provide data in a uniform manner so that clients can access and contribute data irrespective of the storage format. Using cloud computing for HPC data storage will have a significant impact as research teams will no longer need to wait excessively long periods of time to transfer huge volumes of data between locations. Result data itself will become an on demand utility, thus allowing research teams to focus more time on data processing.