Towards Scalable and Efficient Scientific Cloud Computing

Overview
Commercial clouds bring a great opportunity to the scientific computing area. Scientific applications usually need huge resources to run on. However not all of the scientists have access to significant high-end computing systems, such as those found in the Top500. Cloud has gained the attention of the scientist as a competitive resource to run HPC applications at a lower cost. But it is not clear that having a different infrastructure, if they are capable of doing scientific computing. Recently, some of the cloud provider companies have tried to provide infrastructure capable of running scientific applications.

The goal of this research is to investigate the ability of clouds to support the characteristics of scientific applications and bring the ideas to optimize the infrastructure for scientific applications. These applications have grown accustomed to a particular software stack, namely one that supports batch scheduling, parallel and distributed POSIX-compliant file systems, and fast and low latency networks such as 10Gb/s Ethernet or InfiniBand. This work will explore how overhead virtualization techniques (e.g. Palacios VMM), investigate network performance and how it might affect network bound applications, and explore a wide range of parallel and distributed file systems for their suitability of running in a cloud infrastructure.

Methodology
Our method evaluates the capability of different instance types of Amazon EC2 cloud for scientific computing and analysis the cost of cloud computing. The method is divided into three parts:

- First: run the micro benchmarks to measure the actual performance and compare with the theoretical peak that we expect to get:
  - also include a non-virtualized system, to understand virtualization effect.
- Second: evaluate the performance of a virtual cluster of multiple instances, running real applications.
- Third: analyse the cost of the cloud based on the actual performance results.

Performance Metrics
The performance metrics for the experiments are based on the critical requirements of the scientific applications. We divide our metrics into different categories:

- **CPU:**
  - Giga flops (Gflops).
- **Memory:**
  - Capacity: Giga Bytes (GB)
  - Bandwidth: GB per second (GB/s)
- **Network:**
  - Bandwidth: Gigabits per sec (Gb/s)
  - Latency: milliseconds (ms)

Performance Results
**Compute performance**
- Low efficiency on compute performance
- High overhead of virtualization on processors

**Network performance**
- Predictable/stable network performance on single client/server case
- Poor scalability on multiple client case, Not scalable/predictable

**Memory Performance**
- The memory bandwidth scales perfectly on some instance types. HPC instances beat not-virtualized nodes at large scale
- Unstable write performance, Stable read performance

Virtual Cluster Performance
- Poor efficiency at larger scales
- Reason: poor network performance, virtualization effect

Performance Variance
**Small Instance Memory Bandwidth**
- High memory bandwidth variance
- Reason: an entire CPU core is not allocated to each instance, CPU cores are time shared across the instances

**Micro Instance Compute Performance**
- High compute performance variance

Cost Analysis
- High-memory instances are the most cost-effective instance in memory capacity. But for the memory bandwidth, smaller instances are more cost-effective

Conclusion
- Measured compute performance lower than expected
- Relatively poor and inconsistent interconnect performance inside datacenters
- Need for dedicated resources rather than virtualized, shared resources; need for better isolation of virtualized resources
- Cost effective instances are the smaller instances, poor choice for HPC
- Need for new lightweight hypervisors that focus on HPC (e.g. Palacios)

Future work
- Apply new techniques on the VMM level and evaluate the performance:
  - Pass-through I/O technique
  - Customized Memory Paging technologies
- Optimize network virtualization by adding ‘driver’ domain on VMM level and redefining virtual network interface

References
- Amazon EC2: http://aws.amazon.com/ec2/