# Clouds Usage for Lattice Optimization at Advanced Light Source

Cloud Summer Institute 2011
Krishna Muriki
High Performance Computing Services
IT Division
June 22nd 2011.

Work presented at 2011 Particle Accelerator Conference.
Changchun Sun, Hiroshi Nishimura, Susan James, Kai Song, Krishna Muriki, Yong Qin

# Overview

Overview:
- Case study
- Evaluate a different technology from user's perspective
- Feasibility for Cloud HPC Computing

Outline:
- Advanced Light Source (ALS)
- Lattice Optimization problem
- Genetic Algorithm
- Use of Local resources
- Use of Amazon Cloud resources
- Comparisons & Conclusions

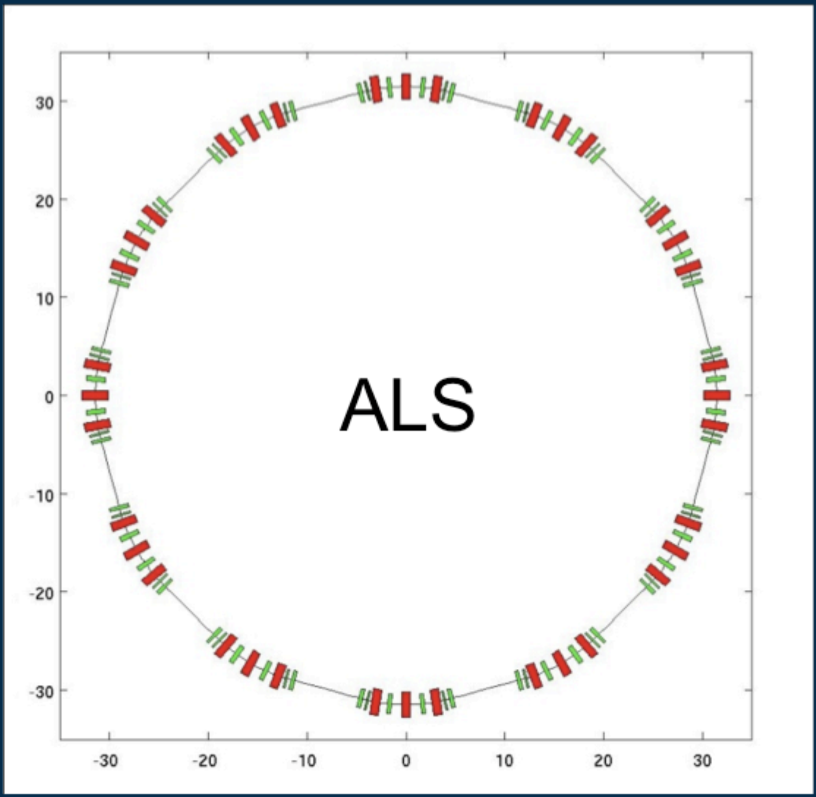
# Advanced Light Source

- First 3rd generation Synchrotron Light Source.
- National user facility commissioned at LBNL in 1993.
- X-ray light produced is used for research in materials science, biology, chemistry, physics and the environmental sciences.

ALS

BERKELEY LAB

# ALS Ring and Lattice




- About 200m circumference storage ring
- Electrons orbit in the ring at nearly the speed of light.
- Magnet sets called Lattice keep electrons circulating.
- Lot of Magnets Dipole, Quadrupole, Sextupole, etc..
- The orbiting electrons emit synchrotron radiations, which are used for scientific research.

# Lattice Optimization

- To keep ALS competitive in the future, storage ring upgrades are necessary to improve the brightness and beam lifetime.
- A challenging aspect of the upgrade is lattice optimization.
- Evaluate multiple objectives by optimizing multiple Lattice variables at the same time.

| Variables | Objectives |
|---|---|
| Magnet strengths | High Beam life time |
| Quadrupole, Sextupole, etc.. | High Brightness |
| | Small Emittance |

- A multi-objective and multi-variable optimization problem.

# Computational Techniques

- Brute force scan
  - Works for upgrades with small number (4) of magnet sets

But potential upgrades have 12 or more magnet sets.

- Genetic Algorithm
  - Best option for large number of input variables
  - Run over large number of populations and generation sizes for optimal solutions.
  - Needs a lot of computational capability

# Genetic Algorithm

- *"GeneticTracy"* code implemented
  - using Message Passing Interface (MPI).
  - Master-Slave model to achieve parallelization.
    - Master performs genetic operations and distribute tasks to slaves.
    - Slaves evaluate objectives and send results back to the master.
  - Master Pseudo code

1: Randomly generate the first generation
2: Evaluate the first generation → Distribute to Slaves
3: Sort the first generation
4: Repeat
5: select parent to generate child (cross over)
6: mutate child
7: evaluate child → Distribute to Slaves
8: merge the parent and child
9: sort the mixed population
10: select the first half of the mixed populations
11: Until reach maximum generation

(Embarrassingly parallel)

ALS

BERKELEY LAB

# Use of Local Clusters

Local cluster resources used for GeneticTracy computations

Computations are embarrassingly parallel

- Shared resources
  - Number of nodes are limited
  - Queue wait times may vary and can be long
- Fast turn around time critical for research & development
- Is it worthwhile to purchase own cluster?

# Amazon EC2 Cloud

Amazon EC2 HPC Cloud offering:

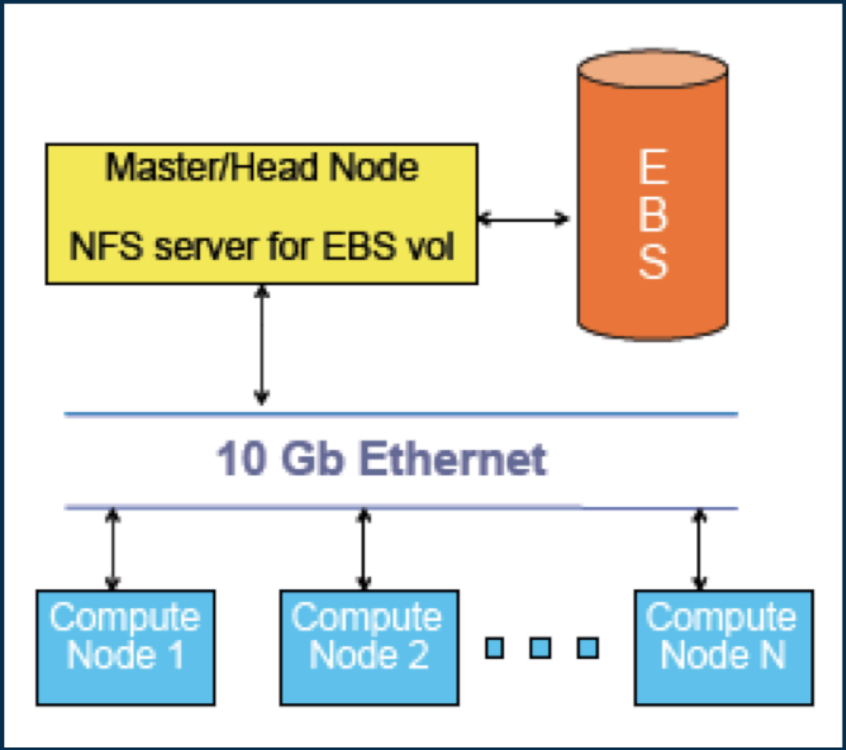
Cluster Computer Instance (CCI)

- Recently introduced by Amazon EC2
- Available only from US-EAST region today
- Pre-defined architecture & Hardware specification
- Instance specifications:
  - Dual Quad core Intel Nehelam processors
  - 23 Gb memory
  - 10Gb Ethernet
- HVM (Hardware Virtual Machine) Virtualization
- Hardware not shared with other EC2 instances
- On demand $0.20/core-hour

# Virtual Cluster in Cloud

Cluster Configuration

- Managed by AWS Console and command line API
- NFS mounted EBS volume
- node launch/terminate script.
- Same placement group will make sure nodes are close by. (same rack)
- Typical small/medium HPC cluster



# Cluster Configurations

- Comparison (specifications)

## Cluster Configurations

| | EC2 | LRC | Mako | LR2 |
|---|---|---|---|---|
| CPU Arch | X5570 | E5430 | E5530 | X5650 |
| CPU Freq (GHz) | 2.93 | 2.66 | 2.40 | 2.66 |
| Cache (MB) | 8 | 12 | 8 | 12 |
| HT | On | Off | Off | Off |
| Interconnect[1] | 10 | 20 | 40 | 40 |
| Virtualization | On | Off | Off | Off |
| Cores/Node | 16 | 8 | 8 | 12 |
| Memory/Node | 23 GB | 16 GB | 24 GB | 24 GB |

[1]. Gb/s

# Performance Comparisons

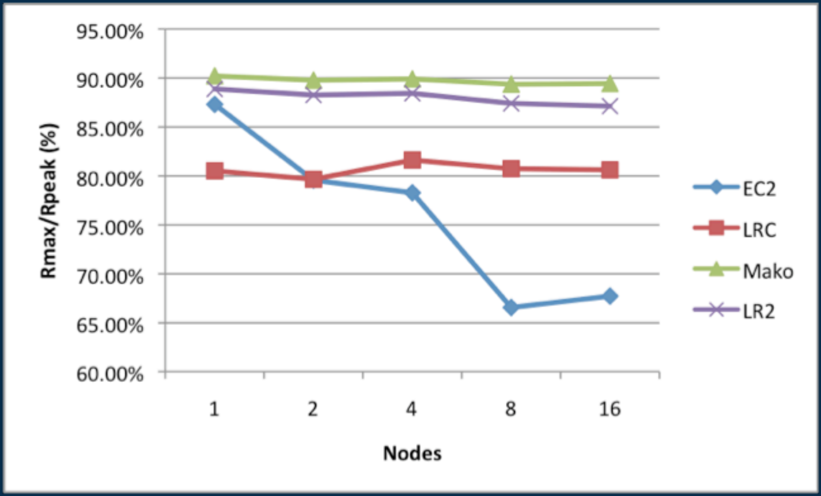
Comparison (performance)

**Runtime on Clusters**

| | EC2 | LRC | Mako | LR2 |
|---|---|---|---|---|
| Time (secs) | 679 | 857 | 724 | 566 |

General Benchmark

- HPL benchmark
- node communication causes Performance limitation



# Cost Comparisons

Comparison (cost)

- Apple to Apple comparison is hard
- Facility cost varies greatly
- Electricity and Cooling depend on local rates and data center efficiencies
- Effective cost per core/hour depends on local cluster hardware and utilization

| | EC2 | Local Cluster |
|---|---|---|
| Hardware | X | X |
| Facilities | X | X |
| Electricity & Cooling | X | X |
| HW effort | X | X |
| SW effort | Not provided | X |

$0.20 $0.01??

# Conclusions

- Cloud HPC is feasible, but costly
- Requires some expertise in system administration to setup quickly
- Suitable for small scale needs without access to local shared resources.
- Use only CCI instances for HPC workloads