TOWARDS EFFICIENT HPC IN THE CLOUD

Abhishek Gupta, 4th year Ph.D. student, CS, UIUC. Advisor: Laxmikant V. Kale

gupta59@illinois.edu

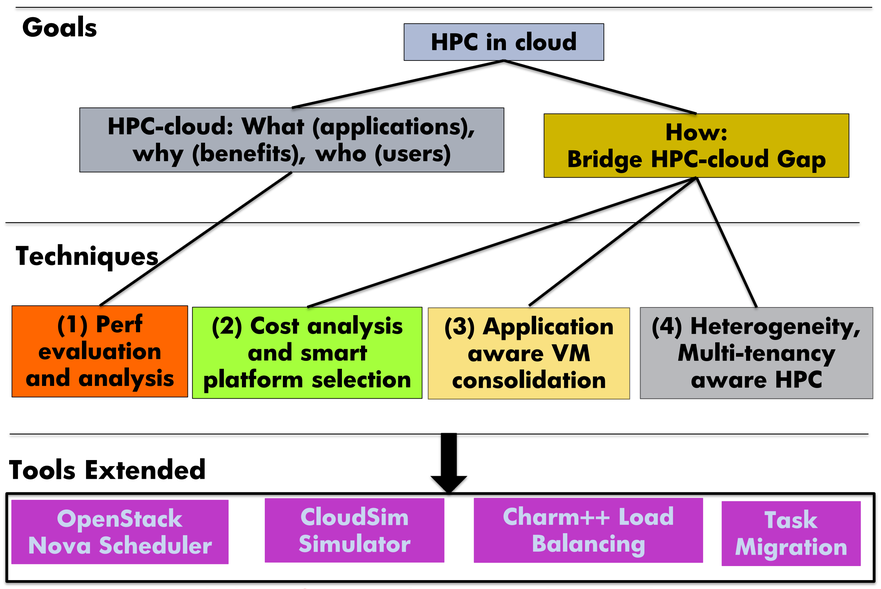
Motivation and Problem

– Why clouds for HPC
- Rent vs. own, *pay-as-you-go*
- *Elastic* resources
- Virtualization benefits – customization, isolation, migration, resource control

– HPC cloud divide
- Performance vs. resource utilization
- Dedicated execution vs. multi-tenancy
- Homogeneity vs. inherent heterogeneity
- HPC-optimized interconnects vs. commodity and virtualized networks

Mismatch: HPC requirements and cloud characteristics
- Only embarrassingly parallel or small scale HPC applications currently run in clouds

Research Goals and Contributions



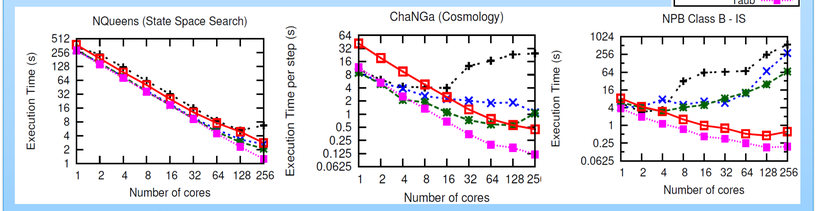
Past research has focused on just the "What" question

1. Performance Evaluation

1a. Experimental Testbed

Platform/ Resource	Ranger (TACC)	Taub (UIUC)	Open Cirrus (HP Labs)	Private Cloud (HP Labs)	Public Cloud
Network	Infiniband (10Gbps)	Voltaire QDR Infiniband	10 Gbps Ethernet internal; 1 Gbps Ethernet x-rack	Emulated network card, KVM (1Gbps physical Ethernet)	Emulated network , KVM (1Gbps physical Ethernet)

1b. Performance of standard platforms



i) Some applications are cloud-friendly
NQueens, NPB-EP, Jacobi2D

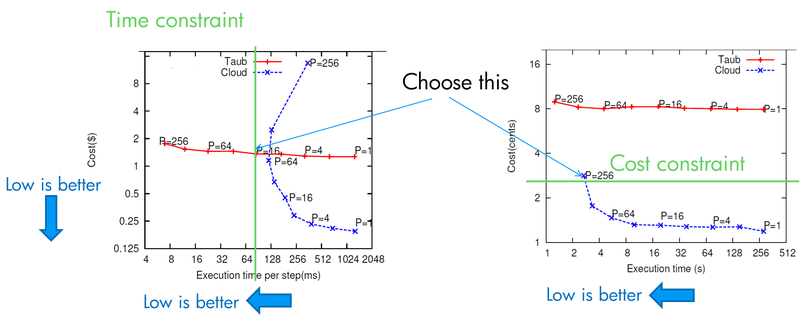
ii) Some applications scale till 16-64 cores
ChaNGa, NAMD,NPB- LU

iii) Some applications cannot survive in cloud
NPB-IS

Critical factors: cloud commodity interconnect, network virtualization overhead, heterogeneity, and multi-tenancy

2. Cost Analysis and Platform Selection

Cost = Charging rate($ per core-hour) × P × Time



Interesting cross-over points when considering cost. Best platform depends on scale, budget, and application characteristics.

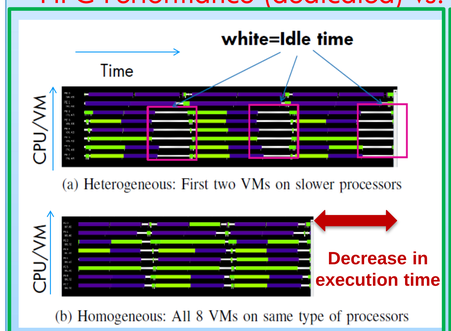
– Platform selection algorithms (meta-scheduler)
- Minimize cost meeting performance target
- Maximize performance under cost constraint
- Consider an application set as a whole
- Which application, which cloud

– Benefits: Performance, Cost, Improved resource utilization

3. HPC-aware Cloud Schedulers

– OpenStack cloud on Open Cirrus (KVM as hypervisor)

– HPC Performance (dedicated) vs. cloud utilization (shared)

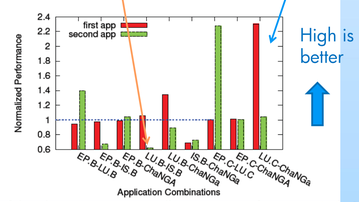


3a. Topology, Hardware aware VM placement

Shared mode (2 apps on each node – 2 cores each on 4 core node) performance normalized wrt. **dedicated mode**

Challenge: Interference

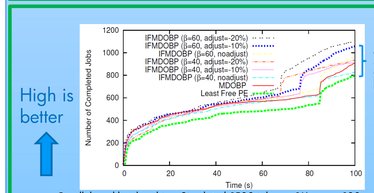
Scope



Careful co-locations can actually improve performance. Why?
Correlation : LLC misses/sec and shared mode performance.

3b. Characterize apps for shared mode execution

a) Cache intensiveness and b) Network sensitivity



- Parallel workload archive, Simulated 1500 jobs on 1K cores, 100 seconds
- Assigned each job a cache score from (0-30) using a uniform distribution random number generator
- β=Cache threshold = degree of resource packing
- Modified execution times (adjust) to account for the improvement in performance resulting from cache-awareness

3c. HPC-aware consolidation

– Dedicated execution for extremely tightly-coupled HPC applications

– For rest, Multi-dimensional Online Bin Packing (MDOBP): Memory, CPU
- Dimension aware heuristic

– Cross application interference aware
- Co-locate apps with complementary execution profile (using 3b)

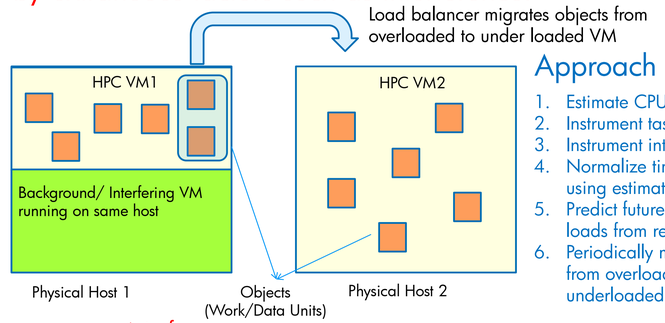
4. Cloud-aware HPC Load Balancer

– Multi-tenancy => Interference => Dynamic heterogeneity

– Random and unpredictable

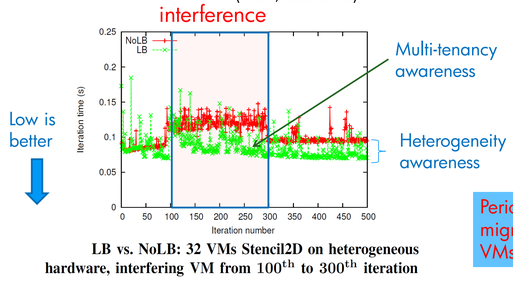
– For HPC, one slow process => all underutilized processes

– **Challenge**: Load imbalance application intrinsic or caused by extraneous factors such as interference.



Approach

1. Estimate CPU Frequencies
2. Instrument task times
3. Instrument interference
4. Normalize times to ticks using estimated frequencies
5. Predict future loads using loads from recent iterations
6. Periodically migrate tasks from overloaded to underloaded VMs



LB vs. NoLB: 32 VMs Stencil2D on heterogeneous hardware, interfering VM from 100th to 300th iteration

Results

Up to 45% benefits for different applications – Stencil2D, Waive2D, Mol3D

Periodically measuring idle time and migrating load away from time-shared VMs works well in practice.

Conclusions

– Cost effective: *some* HPC applications in cloud *not all*

– Multiple platforms + intelligent mapping promising

– Significant performance improvement with LB (40%)

– Substantial throughput improvement with application aware consolidation (32%)

Ongoing Work

– Application characterization (cloud vs. supercomputer)

– Simulate/emulate cloud environment for larger-scale results

Related Publications

- A. Gupta and D. Milojicic, "Evaluation of HPC Applications on Cloud," in Open Cirrus Summit (Best Student Paper), Atlanta, GA, Oct.
- A. Gupta et al., "Exploring the Performance and Mapping of HPC Applications to Platforms in the cloud," in HPDC '12. New York, NY, USA: ACM, 2012
- A. Gupta, D. Milojicic, and L. Kale, "Optimizing VM Placement for HPC in Cloud," in Workshop on Cloud Services, Federation and the 8th Open Cirrus Summit, San Jose, CA, 2012.
- A. Gupta et al., "HPC-Aware VM Placement in Infrastructure Clouds ," in IEEE Intl. Conf. on Cloud Engineering IC2E '13.
- A. Gupta et al., "Improving HPC Application Performance in Cloud through Dynamic Load Balancing," in IEEE/ACM CCGRID '13.