

# Exploring the role of Clouds in Computational Science and Engineering

Manish Parashar\*

(Hyunjoo Kim, Yaakoub el-Khamra and Shantenu Jha)

Center for Autonomic Computing

Rutgers, The State University of New Jersey

(\*Also at OCI, NSF)



# A Cloudy Weather Forecast



Web

## □ A Cloudy Outlook

- About 3.2% of U.S. small businesses, or about 230,000 businesses, use cloud services.
- Another 3.6%, or 260,000, plan to add cloud services in the next 12 months.
- Small-business spending on cloud services will increase by 36.2% in 2010 over a year ago, to \$2.4 billion from \$1.7 billion.

□ Source: IDC, 2010

computers?

# The Lure ....

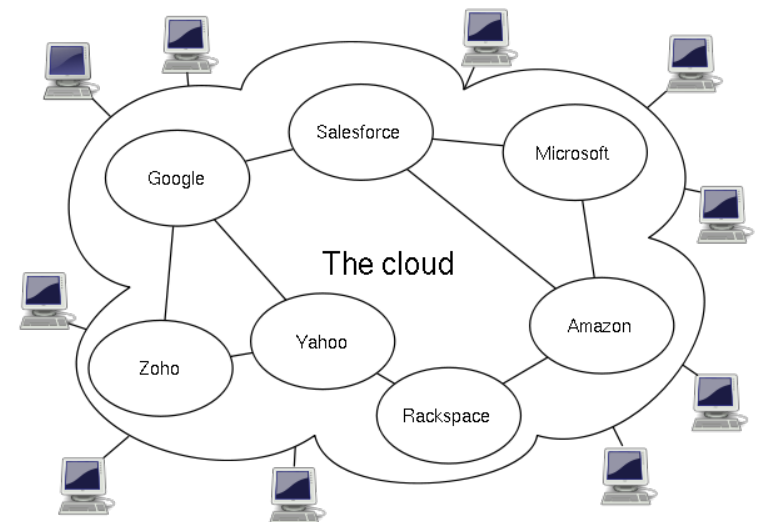
---

- A seductive abstraction – unlimited resources, always on, always accessible!
- Economies of scale
- Multiple entry points
  - \*aaS: SaaS, PaaS, IaaS, HaaS
- IT- outsourcing
  - Transform IT from being a capital investment to a utility
  - TCO, capital costs, operation costs
- Potential for on-demand scale-up, scale-down, scale-out
- Pay as you go, for what you use...
- .....

# Defining Cloud Computing

- Wikipedia – Cloud computing is Internet-based computing, whereby shared resources, software and information are provided to computers and other devices on-demand like a public utility.
- NIST – A cloud is a computing capability that provides an abstraction between the computing resource and its underlying technical architecture (e.g., servers, storage, networks), enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction

- SLAs
- Web Services
- Virtualization



# Cloud Computing Challenges: Complexity, Complexity, Complexity ...

---

## □ Development

- E.g., changes the way software is developed
  - Hardware provisioning, Deployment and Scaling now part of developer lifecycle as a program / script as compared to a Purchase Order

## □ Execution, Runtime Management

- E.g., unique provisioning challenges
  - Multiple entry point distributed, dynamically interleaved application types and workloads; Complex requirements/constraints that must balance efficiency, utilization, costs, performance, reliability, response time, throughput, etc.; Coordination/synchronization challenges; Jitter; IO; ...

## □ System/Application Operation/Management

- Economics, power/cooling, security/privacy, Green-ness, ....

## □ Societal, regulatory, legal, .....

- Need to hand over their data to a third party => big leap of faith

## □ Security, Reliability, Usability, ....

- Misbehaving clouds can have potentially disastrous consequences.....

# CS&E on the Cloud

---

- Clouds support different although complementary usage models as compared to more traditional HPC grids
  
- Some questions
  - Application types and capabilities that can be supported by clouds?
  - Can the addition of clouds enable scientific applications and usage modes that are not possible otherwise?
  - What abstractions and systems are essential to support these advanced applications on different hybrid grid-cloud platforms?

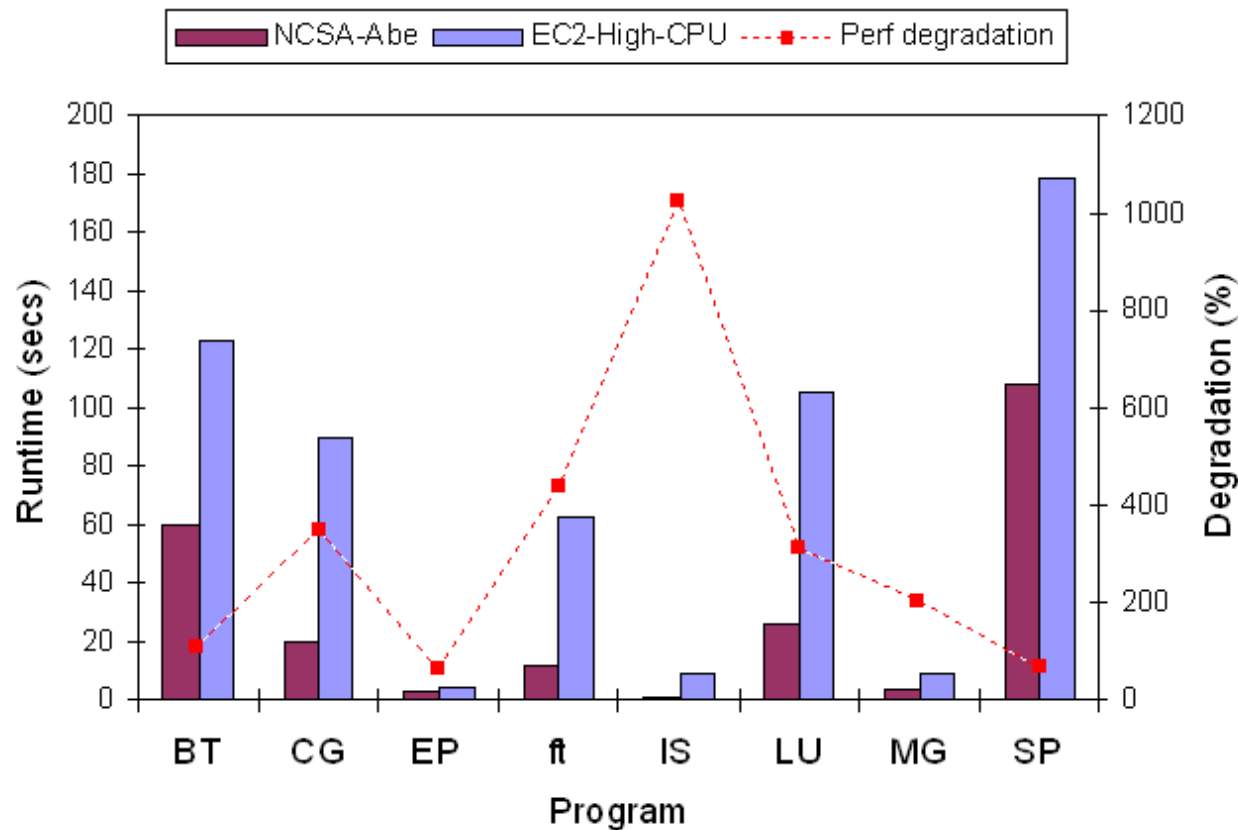
# CS&E on the Cloud - Obvious candidates

---

- Nicely parallel
- Minimal synchronization, Modest I/O requirements
- Large messages or very little communication
- Low core counts

# MPI benchmarks on Clouds

## □ NAS Parallel Benchmarks, MPI, Class B



E. Walker, "Benchmarking Amazon EC2 for High-Performance Scientific Computing," *login*, 2008.



# CS&E on the Cloud – Moving beyond the obvious candidates

---

- New application formulations
  - Asynchronous, resilient
  - E.g., Asynchronous Replica Exchange Molecular Dynamics, Asynchronous Iterations
  
- New usage modes
  - Client + Cloud accelerators
  - E.g., Excel + EC2
  
- New hybrid usage modes
  - Cloud + HPC + Grid

# CometCloud (cometcloud.org)

- Framework for enabling applications on dynamically federated, hybrid infrastructure
  - Integrate (public & private) clouds, data-centers and HPC grids
  - On-demand scale up, down, out

- High-level **programming abstractions** and autonomic mechanisms

- Coordination/interaction through virtual **shared spaces**

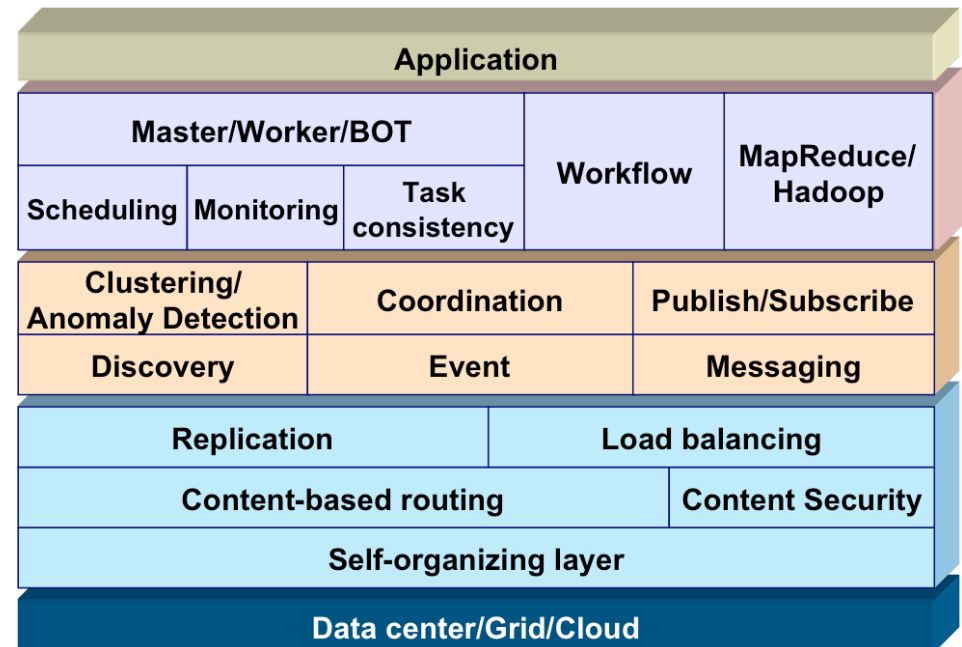
- **Autonomic (macro/micro) provisioning**

- **Runtime self-management**, push/pull scheduling, dynamic load-balancing, self-organization, fault-tolerance

- **Diverse applications**: business intelligence, financial analytics, oil reservoir simulations, medical informatics, document management, etc.

- **Cross-layer Autonomics**

- **Application/Programming layer autonomics**: Dynamics workflows; Policy based component/service adaptations and compositions
- **Service layer autonomics**: Robust monitoring and proactive self-management; online provisioning, dynamic application/system/context-sensitive adaptations
- **Infrastructure layer autonomics**: On-demand scale-out; resilient to failure and data loss; handle dynamic joins/departures; support “trust” boundaries



# CometCloud – Some Applications

---

- VaR analytics engine
  - *"Online risk analytics on the cloud," International Workshop on Cloud Computing , Cloud 2009, Shanghai, China, May 2009.*
- Medical informatics
  - *"Investigating the use of cloudbursts for high-throughput medical image registration, GRID2009 , Banff, Canada, Oct. 2009.*
- Molecular dynamics & drug design
  - *"Accelerating MapReduce for Drug Design Applications: Experiments with Protein/Ligand Interactions in a Cloud," submitted for publication, 2009.*
  - *"Asynchronous Replica Exchange for Molecular Simulations, Journal of Computational Chemistry, 29(5), 2007.*
- PDEs solvers using synchronous and asynchronous iterations
  - *A decentralized computational infrastructure for grid based parallel asynchronous iterative applications," Journal of Grid Computing, 4(4), 2006.*
- Others...
  - MapReduce acceleration
  - System level acceleration
  - Workflow engine
    - parameter estimation, autonomic oil reservoir optimization

<http://www.cometcloud.org>

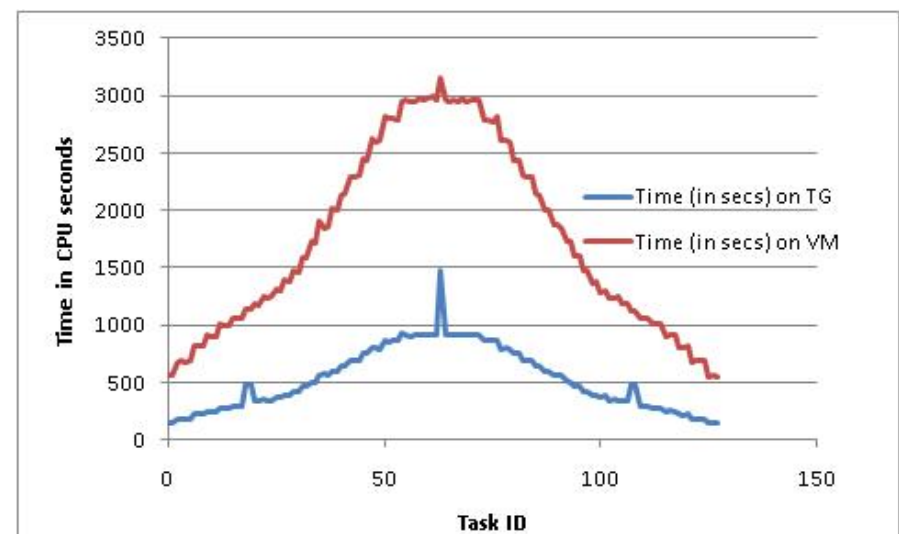
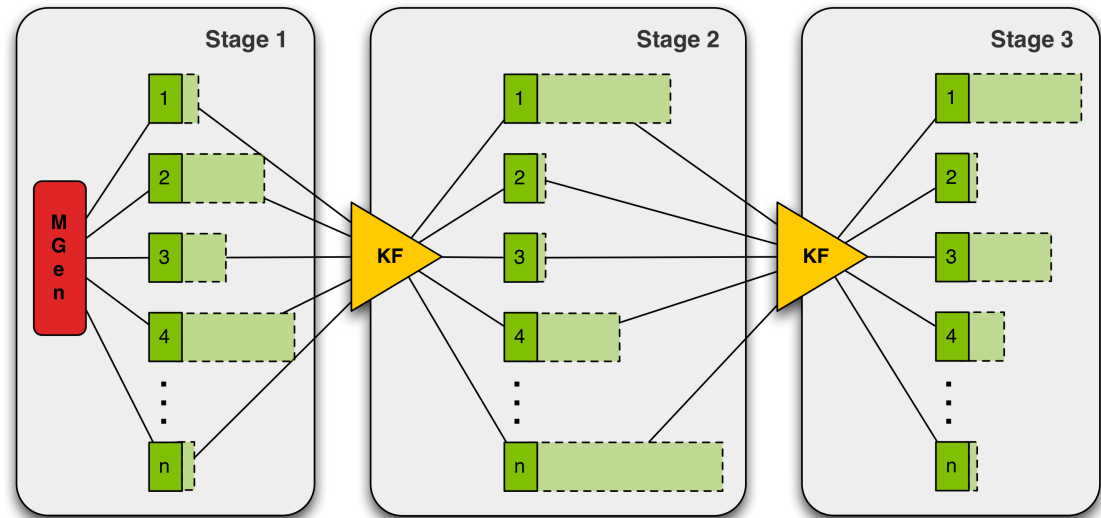
# Exploring Hybrid HPC-Grid/Cloud Usage Modes

---

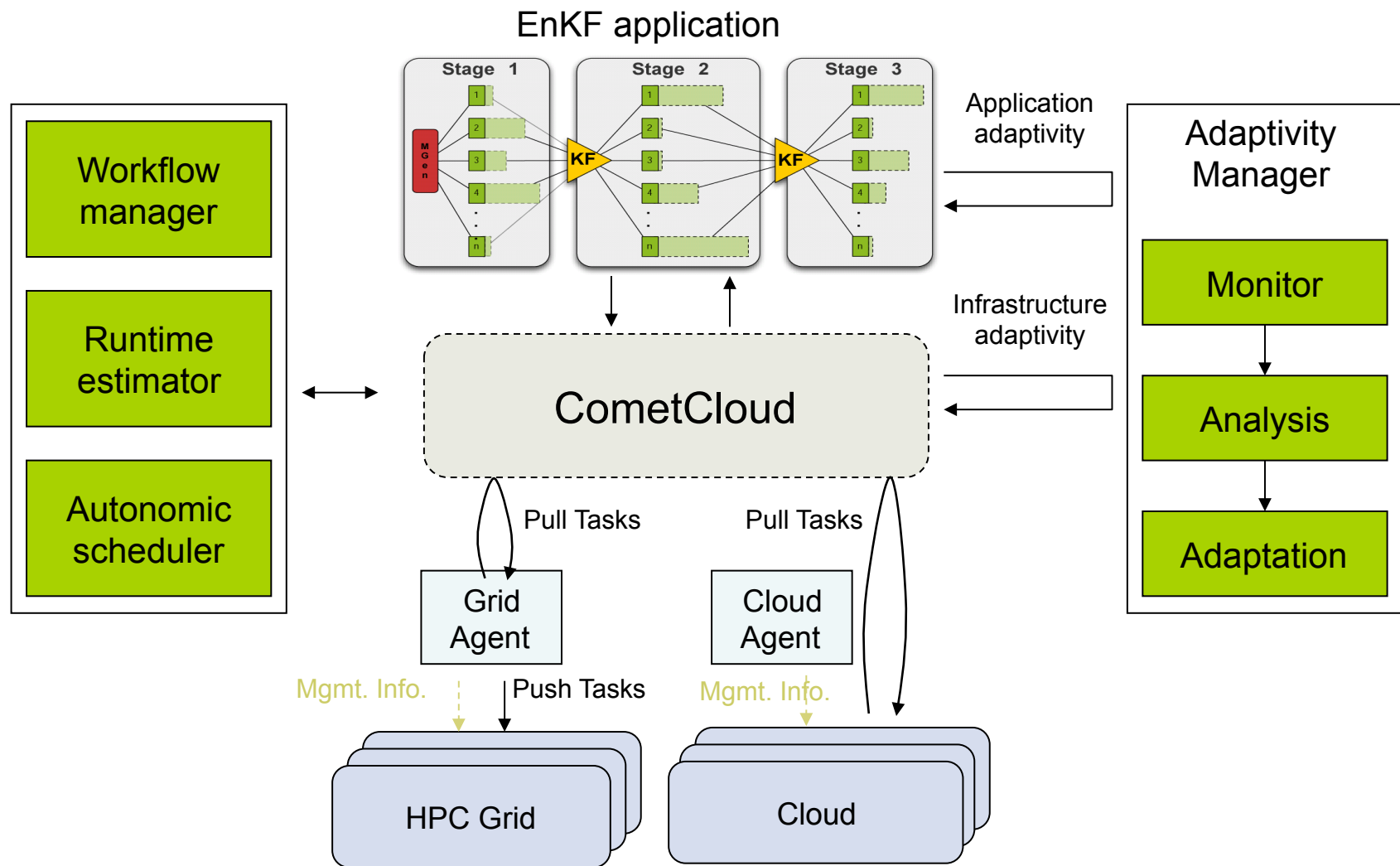
- What are appropriate usage modes for hybrid infrastructure?
  - Acceleration
    - Explore how Clouds can be used as accelerators to improve the application time to completion
      - To alleviate the impact of queue wait times
      - “Strategically Off load” appropriate tasks to Cloud resources
      - All while respecting budget constraints.
  - Conservation
    - How Clouds can be used to conserve HPC Grid allocations, given appropriate runtime and budget constraints.
  - Resilience
    - How Clouds can be used to handle:
      - General: Response to dynamic execution environments
      - Specific: Unanticipated HPC Grid downtime, inadequate allocations or unexpected Queue delays/QoS change

# Reservoir Characterization: EnKF-based History Matching

- ❑ Black Oil Reservoir Simulator
  - simulates the movement of oil and gas in subsurface formations
- ❑ Ensemble Kalman Filter
  - computes the Kalman gain matrix and updates the model parameters of the ensembles
- ❑ Heterogeneous workload, dynamic workflow
- ❑ Based on Cactus, PETSc



# Exploring Hybrid HPC-Grid/Cloud Usage Modes using CometCloud



# Experimental environments

---

- Three stages of the EnKF workflow with 20x20x20 problem size and 128 ensemble members with heterogeneous computational requirement
- Deploy EnKF on TeraGrid (16 cores) and several instance types of EC2 (MPI enabled)

Instance type	Cores	Memory(GB)	Platform(bit)	cost(\$/hour)
m1.small	1	1.7	32	0.1
m1.large	2	7.5	64	0.34
m1.xlarge	4	15	64	0.68
c1.medium	2	1.7	32	0.17
c1.xlarge	8	7	64	0.68

*TABLE I:* EC2 instance types used in experiments

# Experiment Background and Set-Up

---

## □ Key metrics

- Total Time to Completion (TTC)
- Total Cost of Completion (TCC)

## □ Basic assumptions

- TG gives the best performance but is relatively more restricted resource.
  - EC2 is a relatively more freely available but is not as capable.
- Note that the motivation of our experiments is to understand each of the usage scenarios and their feasibility, behaviors and benefits, and not to optimize the performance of any one scenario.

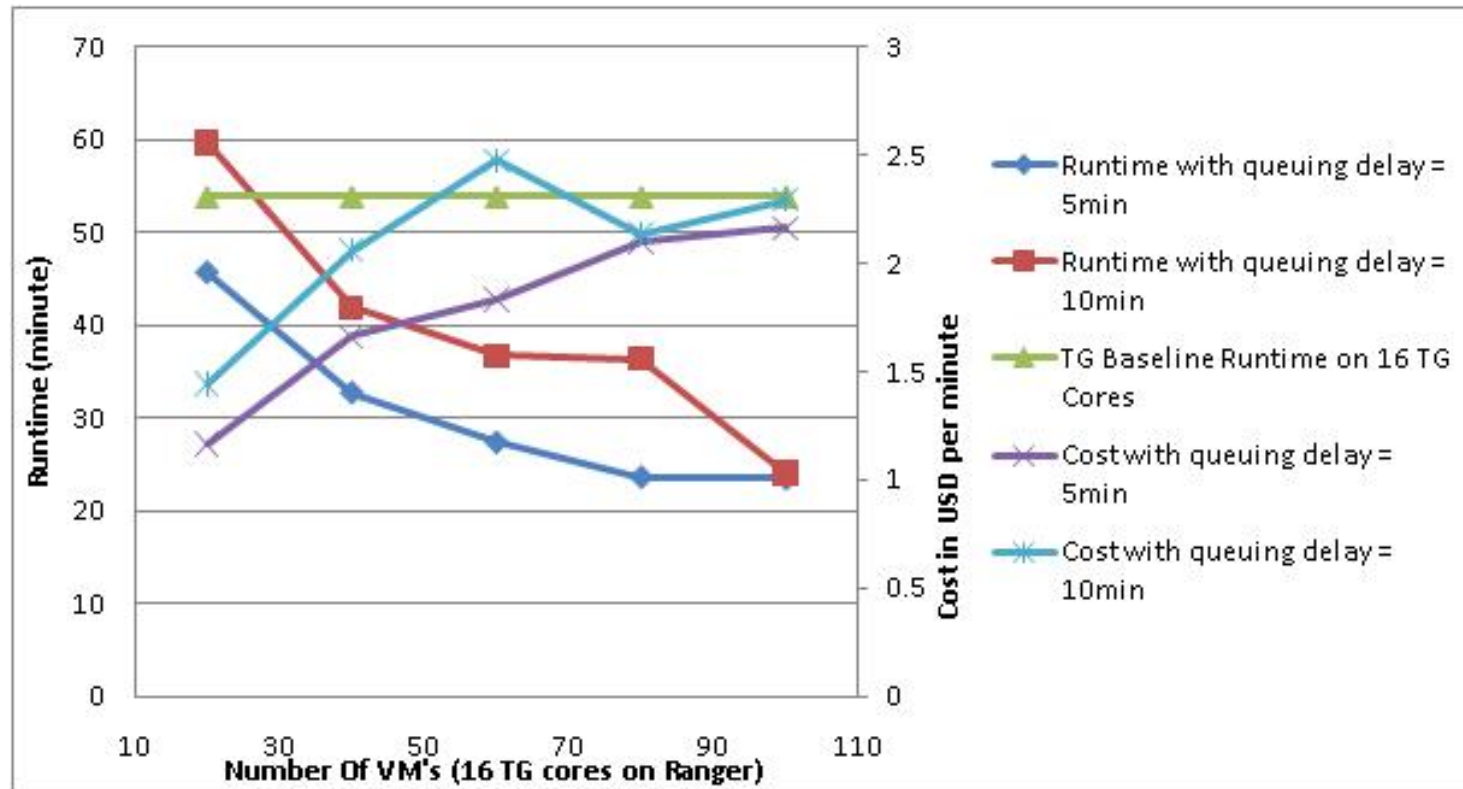


# Objective I: Using Clouds as Accelerators for HPC Grids (1/2)

---

- Explore how Clouds (EC2) can be used as accelerators for HPC Grid (TG) work-loads
  - 16 TG CPUs (Ranger)
  - average queuing time for TG was set to 5 and 10 minutes.
  - the number of EC2 VMs (m1.small) from 20 to 100 in steps of 20.
  - VM start up time was about 160 seconds

# Objective I: Using Clouds as Accelerators for HPC Grids (2/2)



The TTC and TCC for Objective I with 16 TG CPUs and queuing times set to 5 and 10 minutes. As expected, more the number of VMs that are made available, the greater the acceleration, i.e., lower the TTC. The reduction in TTC is roughly linear, but is not perfectly so, because of a complex interplay between the tasks in the work load and resource availability

# Exploring Adaptations

---

- Three approaches to adapt applications based on acceleration objective
  - Track1: selection and adaptivity of infrastructure
  - Track2: tuning of applications
  - Track3: adaptivity for both infrastructure and application
  
- Infrastructure adaptivity
  - Explores infrastructure space and selects appropriate numbers of types (e.g., the number and type of virtual machines)
  
- Application adaptivity
  - Involves adapting the structure and behavior of the applications based on application/system characteristics (e.g., the size of ensemble members, problem size and application configuration) and runtime state

# Results – Baseline

- No adaptation is applied
- Resource classes: TeraGrid, c1.medium

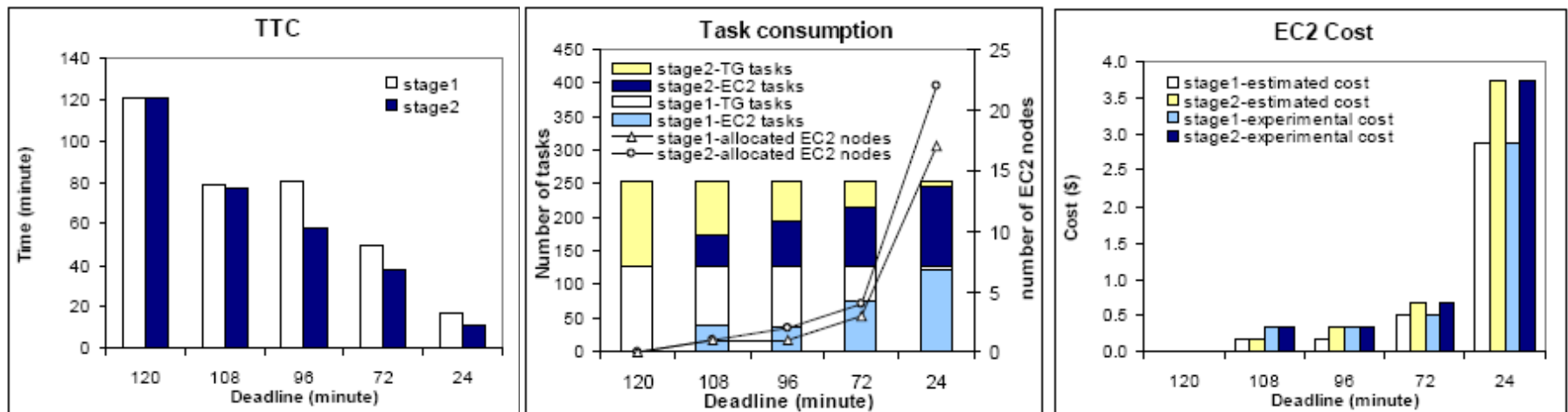


Figure 1: Baseline experiment without adaptivity with a deadline policy. Tasks are completed within a given deadline. The shorter the deadline, the more EC2 nodes are allocated.

# Results - Track 1

- Infrastructure adaptivity
- Resource classes: TeraGrid, 5 instance types of EC2

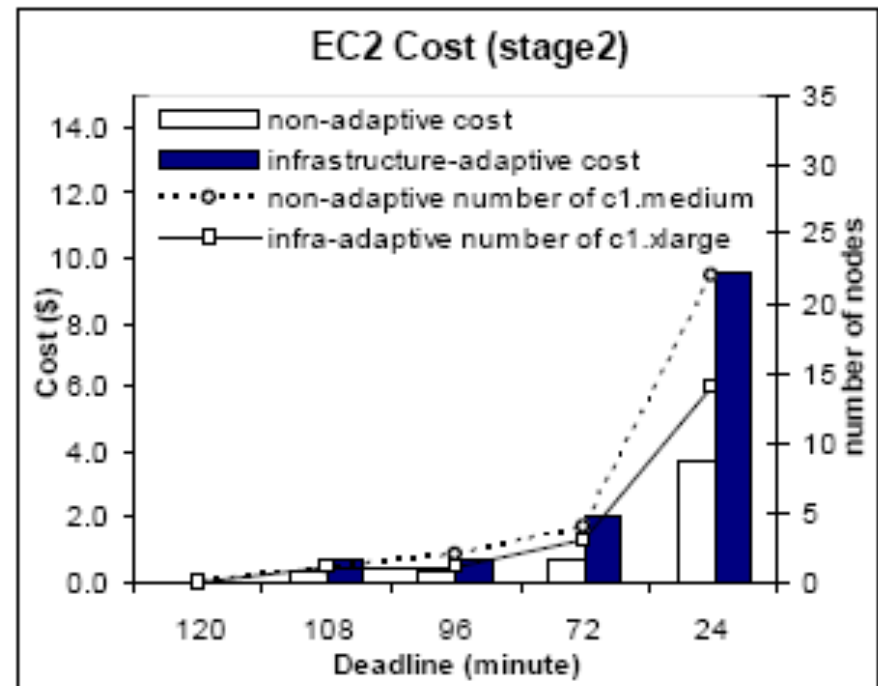
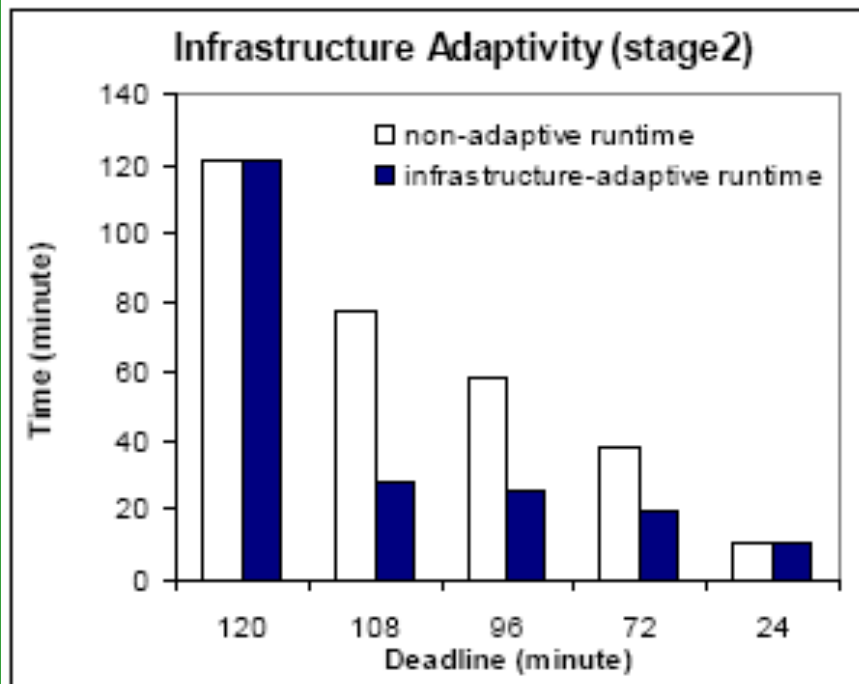


Figure 2: Experiments with infrastructure adaptivity. The TTC is reduced with infrastructure adaptivity at additional cost.

# Results – Track 2

- Application adaptivity
- Resource classes: TeraGrid, 5 instance types of EC2

Figure 3: TTC for simulations of 20x20x20 with different solvers (GMRES, CG, BiCG) and block-Jacobi preconditioner. Benchmarks ran on EC2 nodes with MPI

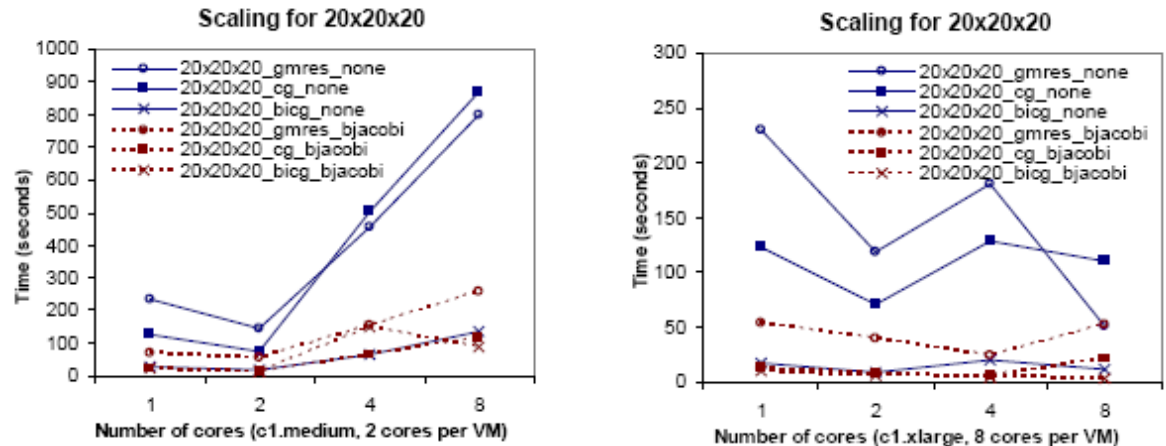
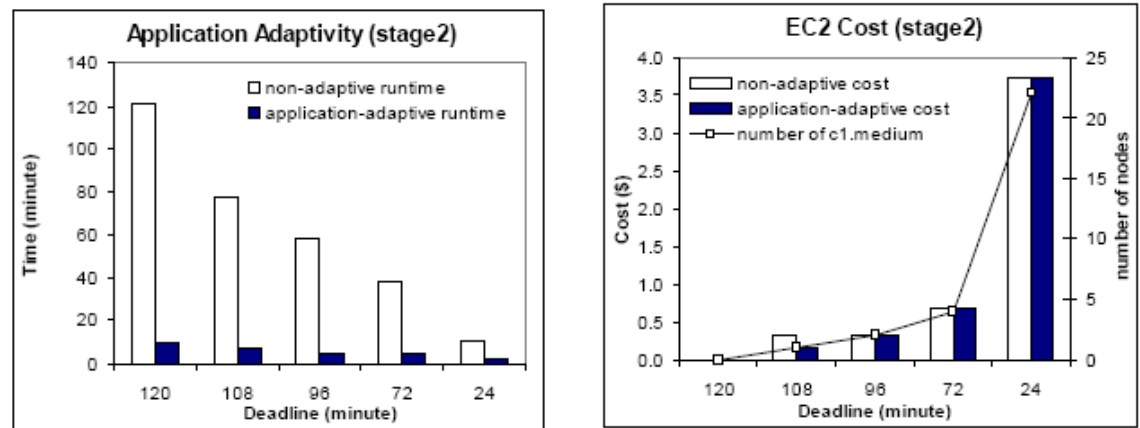


Figure 4: Experiments with application adaptivity. The TTC is reduced with application adaptivity for equivalent or slightly less cost.



# Results – Track 3

- Both infrastructure/application adaptivities
- Resource classes: TeraGrid, 5 instance types of EC2

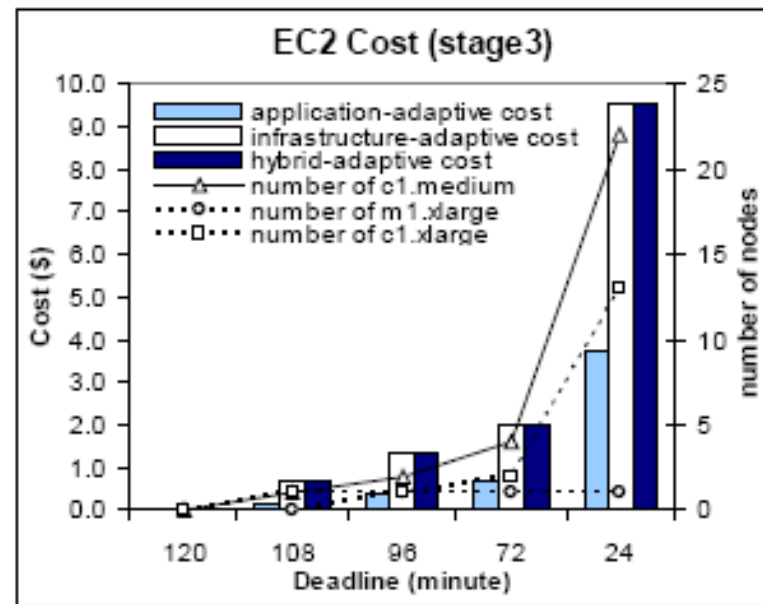
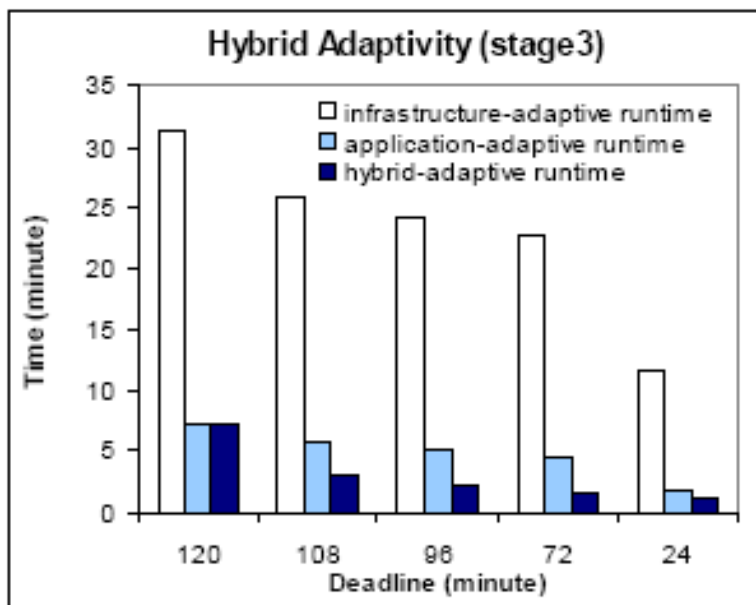
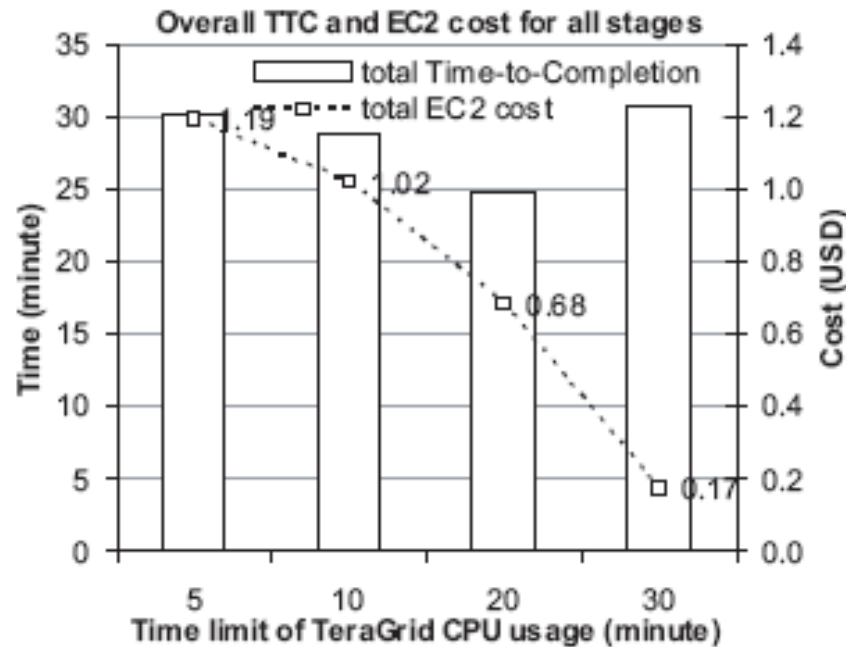


Figure 5: Experiment with adaptivity applied for both infrastructure and application. The TTC is reduced further than with application or infrastructure adaptivity on its own. The cost is similar to that in infrastructure adaptivity for durations less than one hour since EC2 usage is billed hourly with a one hour minimum.

# Exploring Conservation



(a) TTC and EC2 cost

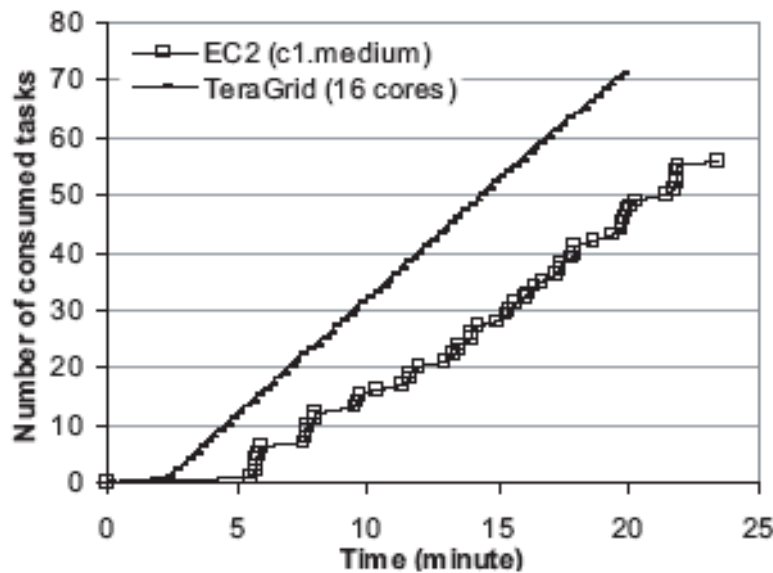
CPU usage limit (min)	5	10	20	30
Num of scheduled VMs (EC2)	7	6	4	1
Num of expected tasks consumed by EC2	111	92	54	14
Consumed tasks by EC2	109	89	49	16

(b) Schedule

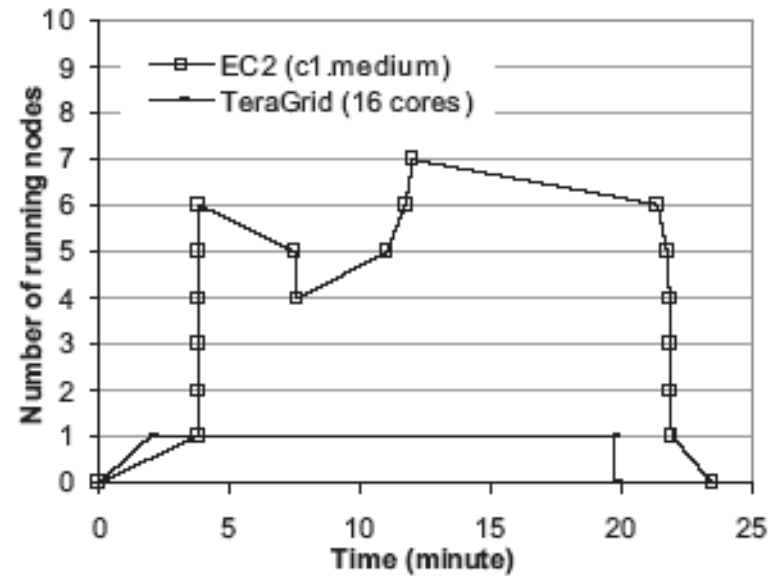


# Exploring Resilience

- Deadline 20 minutes
- Two EC2 instances are failed at around 8 minutes.



(a) Number of consumed tasks



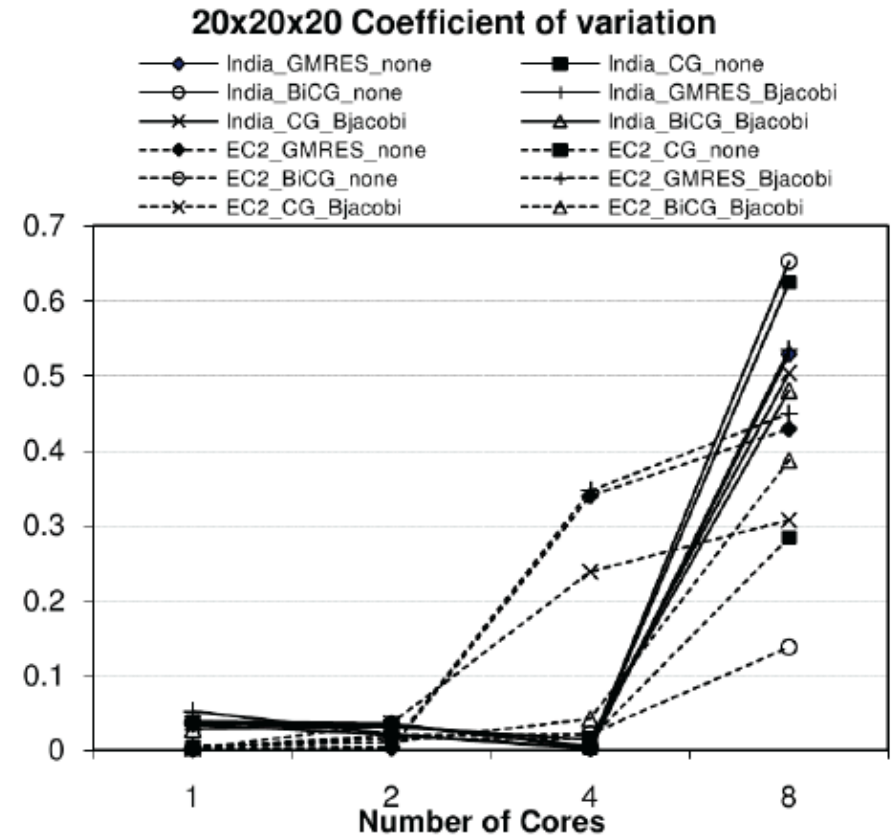
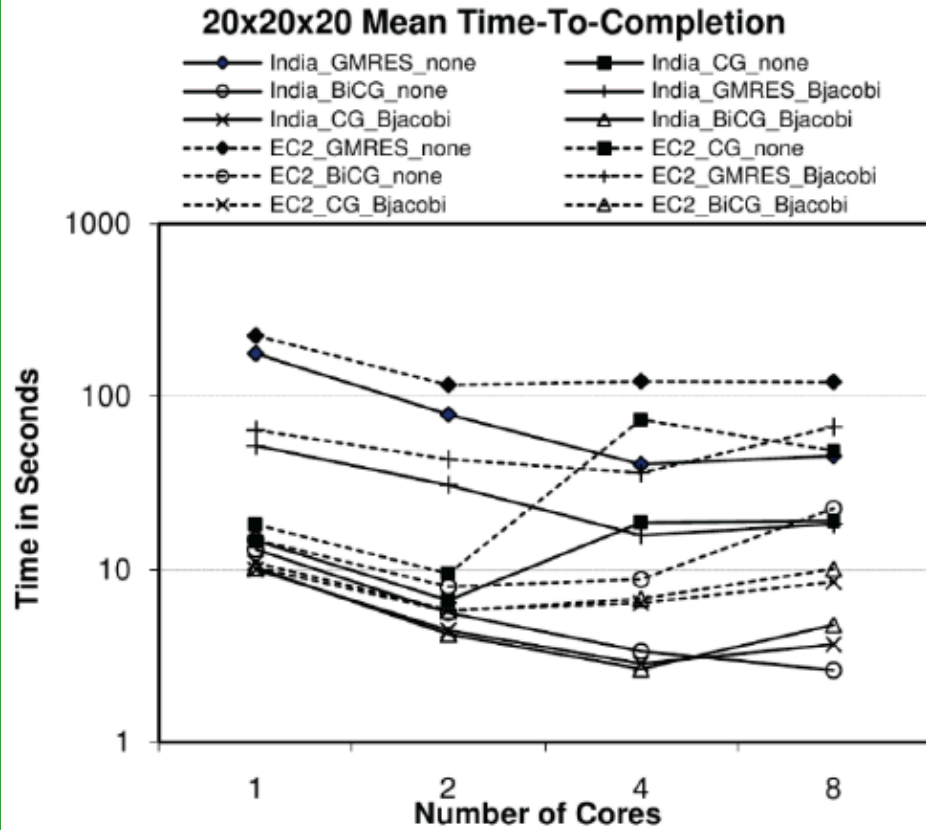
(b) Number of nodes

# Summary and Conclusion

---

- The future is Cloudy...
  - Cloud becoming a part of production computational environments
  - Clouds will play a role in Science and Engineering
  
- Some obvious application candidates
  - nicely parallel, parameter sweeps, data analysis, etc.
  
- However moving beyond obvious candidates requires some thought
  - New application formulations
    - Asynchronous, resilient
  - New usage modes
    - Client + cloud accelerators
  - New hybrid usage modes
    - Cloud + HPC + Grid
  
- Need understand what are the requirements? What are meaningful usage modes? What are the new issues?

# For Example: Performance Fluctuations for HPC Workloads



# Thank You!

---



# References

---

- **[JGC 06]** “A Decentralized Computational Infrastructure for Grid based Parallel Asynchronous Iterative Applications,” Z. Li\* and M. Parashar, Journal of Grid Computing, Special Issue on Global and Peer-to-Peer Computing, Springer-Verlag, Pages 1 – 18, May 2006.
- **[JCC 07]** “Asynchronous Replica Exchange for Molecular Simulations,” E. Gallicchio, R.M. Levy, M. Parashar, Journal of Computational Chemistry, Wiley Periodicals, Inc., Volume 29, Number 5, pp. 788 – 794, 2007.
- **[Cloud 09]** “Online Risk Analytics on the Cloud,” H. Kim, S. Chaudhari, M. Parashar and C. Marty, Proceedings of the International Workshop on Cloud Computing (Cloud 2009), in conjunction with 9th IEEE/ACM International Symposium on Cluster Computing (CCGrid 2009), Shanghai, China, IEEE Computer Society Press, May 2009.
- **[Grid 09]** “Investigating the Use of Cloudbursts for High Throughput Medical Image Registration,” H. Kim, M. Parashar, D. Foran and L. Yang, Proceedings of the 10th IEEE/ACM International Conference on Grid Computing (Grid 2009), Banff, Alberta, Canada, pp. 34 – 41, October 2009.
- **[eScience 09]** “An Autonomic Approach to Integrated HPC Grid and Cloud Usage,” H. Kim\*, Y. el-Khamra, S. Jha and M. Parashar, Proceedings of 5th IEEE International Conference on e-Science (e-Science 2009), Oxford, UK, December 2009.
- **[ScienCloud 10]** “Exploring Adaptation to Support Dynamic Applications on Hybrid Grids-Clouds Infrastructure,” *Proceedings of the 1<sup>st</sup> Workshop on Scientific Cloud Computing (ScienceCloud 2010), co-located with the 19th ACM International Symposium on High Performance Distributed Computing (HPDC 2010)*, Chicago, Illinois, USA June, 2010.