RUTGERS

THE STATE UNIVERSITY
OF NEW JERSEY

Exploring Autonomics for Federated Clouds

Moustafa AbdelBaky, Javier Diaz-Montes, Mengsong Zou and **Manish Parashar**

The NSF Cloud and Autonomic Computing Center
Rutgers, The State University of New Jersey, USA
http://cometcloud.org

In collaboration with Omer Rana, Tom Beach, Ioan Petri, Cardiff University, UK

CARDIFF
UNIVERSITY
PRIFYSGOL
CAERDYDD

Moving towards the Cloud

- Cloud services provide an attractive platform for supporting the computational and data needs of academic and business application workflows

- Cloud paradigm:
  - Rent resources as cloud services on-demand and pay for what you use
  - Potential for scaling-up, scaling-down and scaling-out, as well as for IT outsourcing and automation

- Hybrid cloud services landscape spanning private clouds, public clouds, HEC centers, etc.
  - Heterogeneous offering with different QoS, pricing models, availability, capabilities, and capacities

Cloud Federations – Motivations

- Application workflow exhibit heterogeneous and dynamic workloads, and highly dynamic demands for resources
  - Various and dynamic QoS requirements
    - Throughput, budget, time
  - Often involve large amounts of data
    - Large size, heterogeneous nature, and geographic location

- Such workloads are hard to be efficiently supported using classic federation models

- Implications of the cloud paradigm
  - Rent required resources as cloud services on-demand and pay for what you use
  - Heterogeneous offering with different QoS and costs

- Provisioning and federating an appropriate mix of resources on-the-fly is essential and non-trivial

AUTONOMICS FOR CLOUD FEDERATIONS

Integrating Biology and Information Technology: The Autonomic Computing Metaphor (~2004)

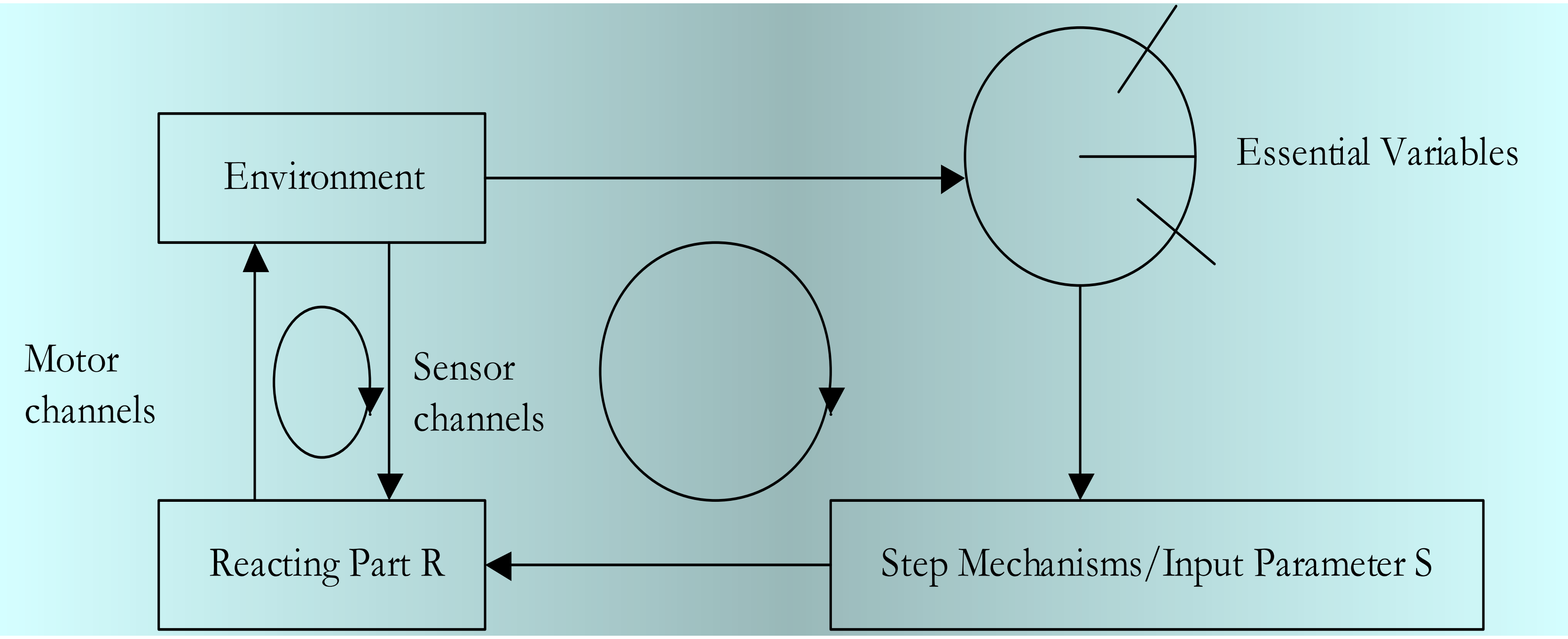
- Current paradigms, mechanisms, management tools are inadequate to handle the scale, complexity, dynamism and heterogeneity of emerging systems and applications

- Nature has evolved to cope with scale, complexity, heterogeneity, dynamism and unpredictability, lack of guarantees
  - self configuring, self adapting, self optimizing, self healing, self protecting, highly decentralized, heterogeneous architectures that work !!!

- Goal of autonomic computing is to enable self-managing systems/ applications that addresses these challenges using high level guidance
  - Separation of policy and mechanisms; Holistic; Automation

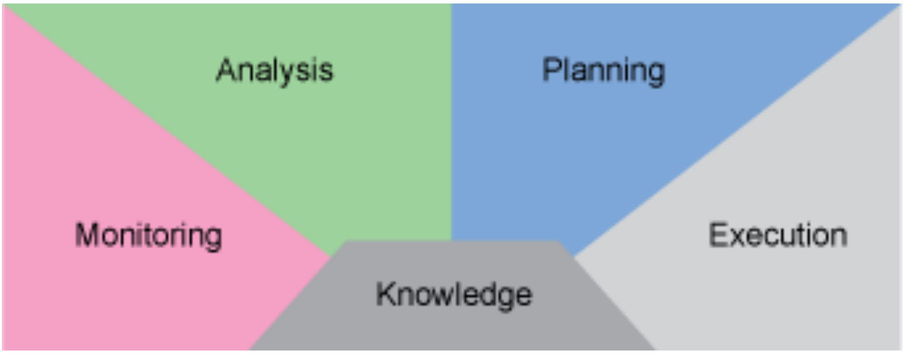
"Autonomic Computing: An Overview," M. Parashar, and S. Hariri, Hot Topics, Lecture Notes in Computer Science, Springer Verlag, Vol. 3566, pp. 247-259, 2005.

Ashby’s Ultrastable System (1920s)



Integrating Biology and Information Technology: The Autonomic Computing Metaphor (~2004)

- Rich body of work on using autonomics for cloud/data-center management
  - Provisioning
  - Workload management
  - Power/energy management
  - Etc…
  - Using control theoretic approaches



"Autonomic Computing: An Overview," M. Parashar, and S. Hariri, Hot Topics, Lecture Notes in Computer Science, Springer Verlag, Vol. 3566, pp. 247-259, 2005.

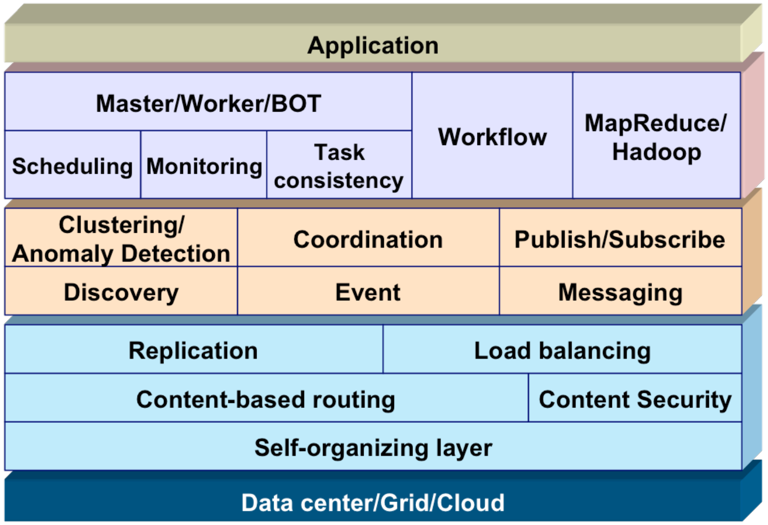
Autonomic Cloud/ACI Federation

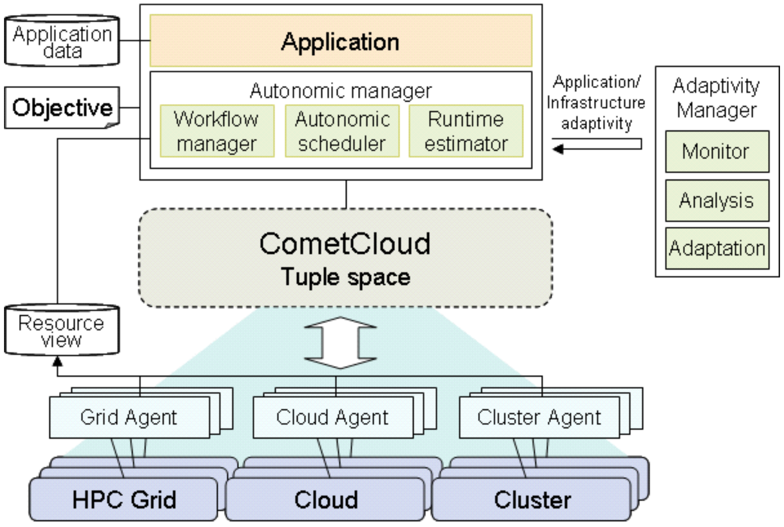
- Assemble a federated cloud/ACI on-the-fly integrating clouds, grids and HPC
  - Cloud-bursting: dynamic application scale-out/up to address dynamic workloads, spikes in demand, and other extreme requirements
  - Cloud-bridging: on-the-fly integration of different resource classes
- Provide policy-driven autonomic resource provisioning, scheduling and runtime adaptations
  - What and where to provision?
  - Policies encapsulate user's requirements (deadline, budget, etc.), resource constraints (failure, network, availability, etc.)
- Provide programming abstractions to support application workflows

CometCloud – Federated Clouds for Science

- Enable applications on dynamically federated, hybrid infrastructure exposed using Cloud abstractions
  - **Services:** discovery, associative object store, messaging, coordination
  - **Cloud-bursting:** dynamic application scale-out/ up to address dynamic workloads, spikes in demand, and extreme requirements
  - **Cloud-bridging:** on-the-fly integration of different resource classes (public & private clouds, data-centers and HPC Grids)
- High-level programming abstractions & autonomic mechanisms
  - Cross-layer Autonomics: Application layer; Service layer; Infrastructure layer
- Diverse applications
  - Business intelligence, financial analytics, oil reservoir simulations, medical informatics, document management, etc.

http://cometcloud.org

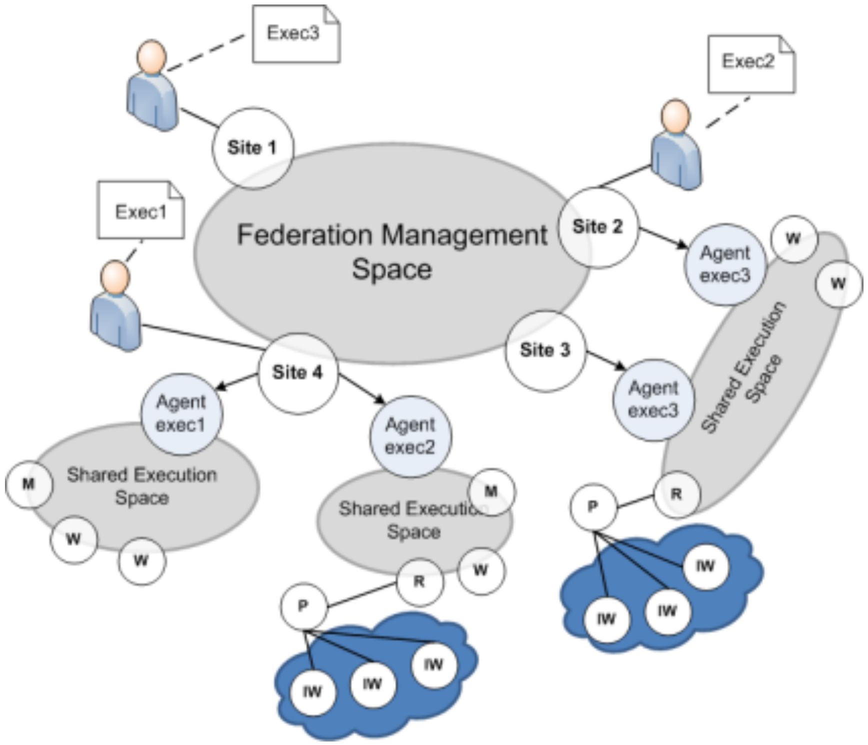




Federated (hybrid) computing infrastructure

On-Demand Elastic Federation using CometCloud

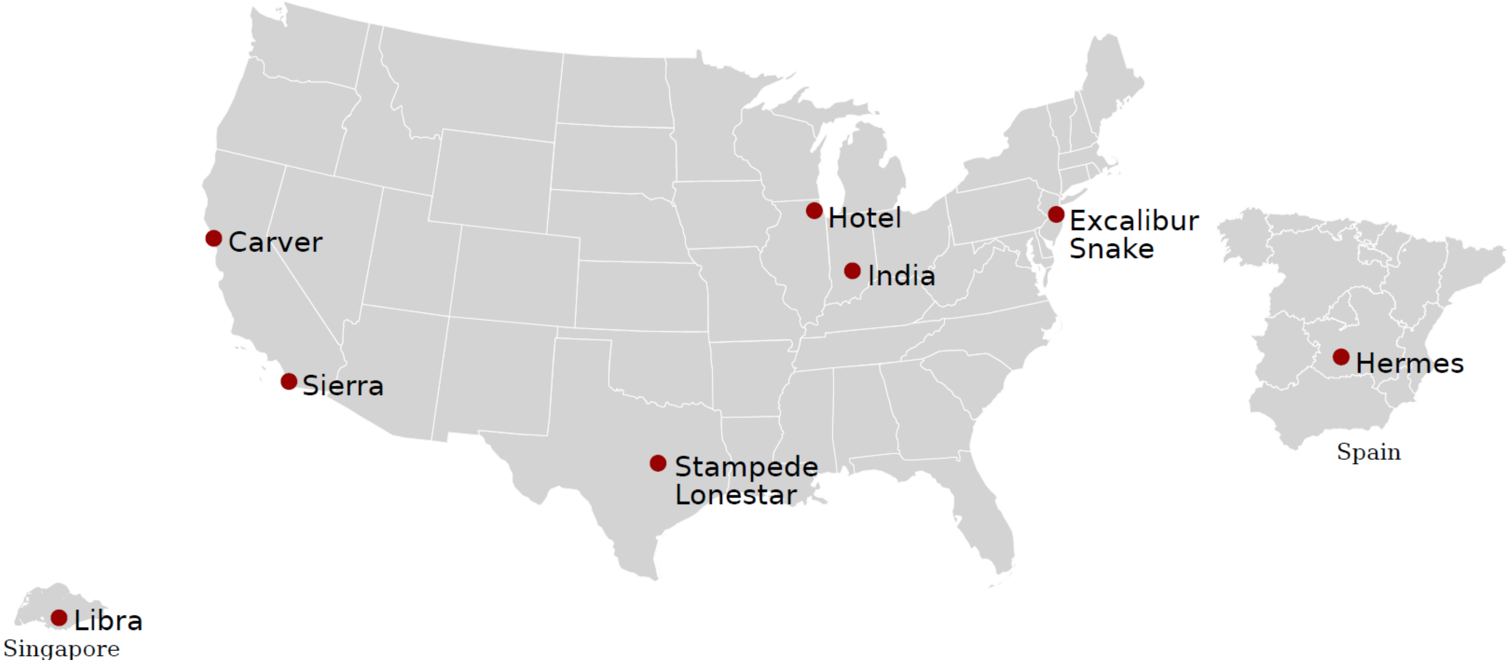
- Software defined ACI federations exposed using elastic on-demand Cloud abstractions
- Autonomic cross-layer federation management using user and provider policies and constraints
  - Separately defined; dynamically evolving
    - Specified based on availability, cost/performance constraints, etc.
    - Assimilated (or removed) dynamically
    - Sites discover/coordinate with each others to:
      - Identify themselves / Verify identity (x.509, public/private key,…)
      - Advertise their own resources capabilities, availabilities, constraints
      - Discover available resources



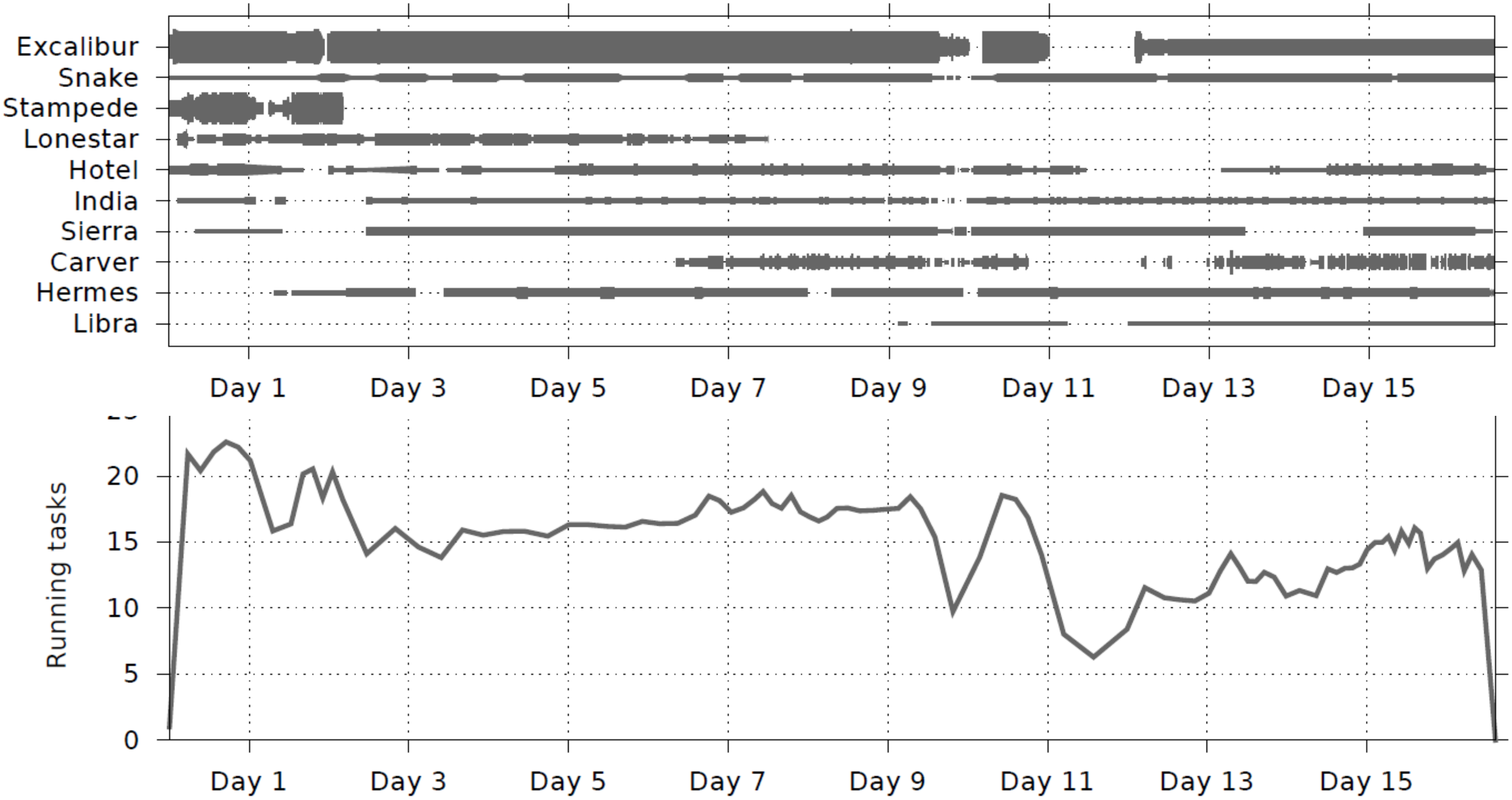
- Federated ACI testbed

UberCloud Experiment

- 10 different resources from 3 countries federated using CometCloud
- 16 days, 12 hours, 59 minutes and 28 seconds of continuous execution
- 12,845 tasks processed, 2,897,390 CPU-hours consumed, 400 GB of data generated



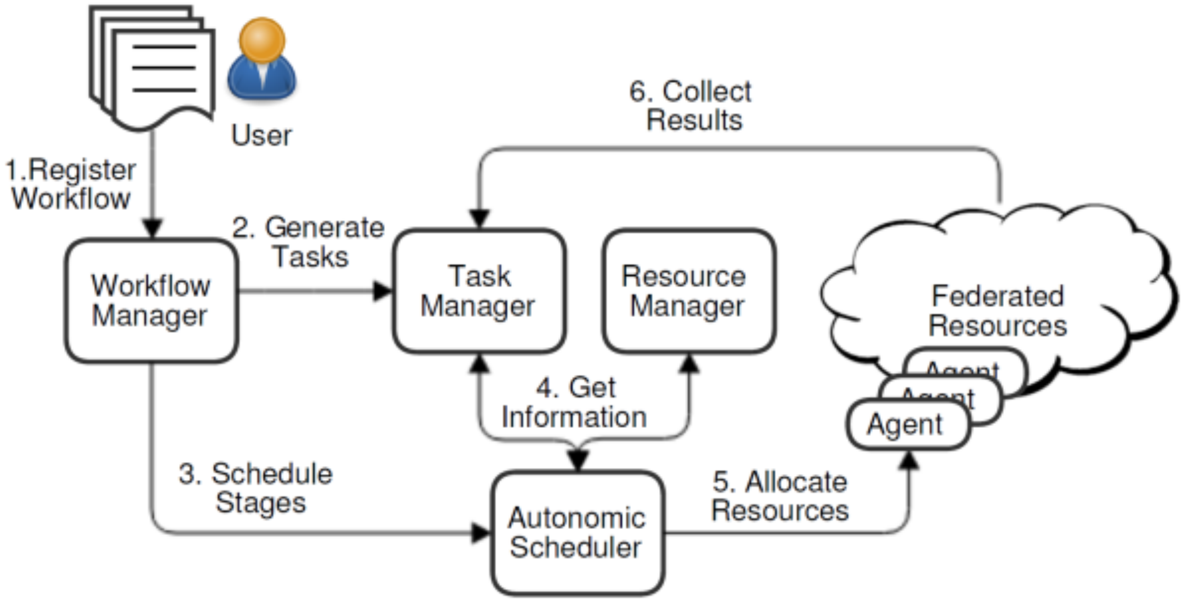
Summary of the experiment



DATA-DRIVEN WORKFLOWS [CLOUD’14] (WITH IBM)

Enabling Data-Driven Workflows

- Enable the autonomic execution of complex workflows in software-defined multi-cloud environments
- Elastically compose appropriate cloud services and capabilities to ensure that the user's objectives are met

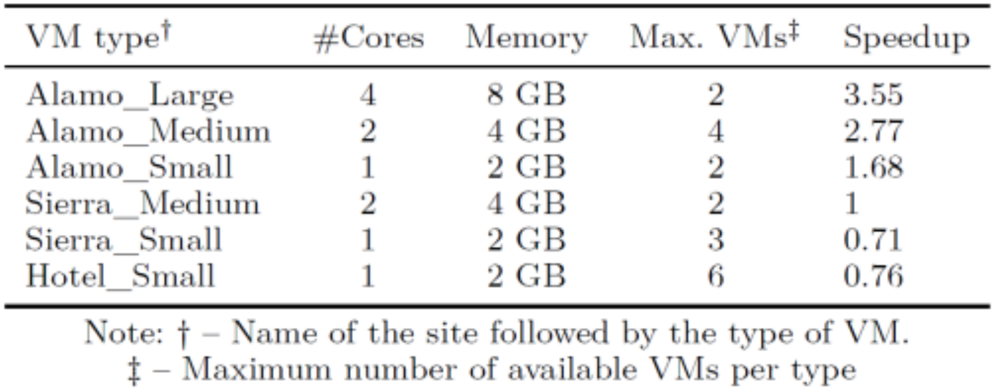


Optimizing Resource Usage in Multi-Clouds

- Execute a data-driven workflow in a multi-cloud environment
- Different scheduling policies and objectives
  - Minimum Completion Time
    - Centralized storage vs Distributed storage
  - Deadline-based Policy
    - Performance optimization (Proc)
    - Data locality optimization (Data)
    - Performance and data optimization (ProcData)
    - Cost optimization (Cost)

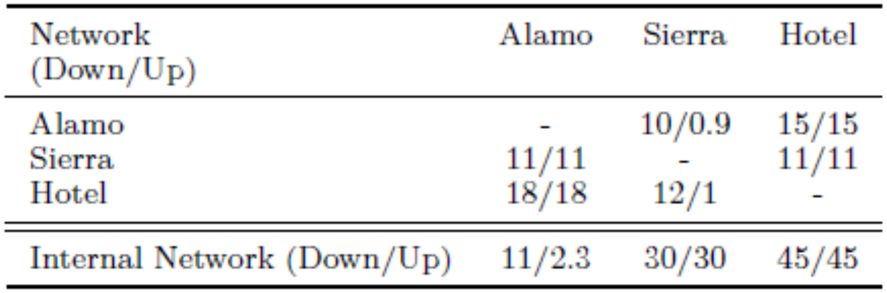
Experiment Setup

- Montage workflow
- Three heterogeneous and geographically distributed clouds



VM type[†]	#Cores	Memory	Max. VMs[‡]	Speedup
Alamo_Large	4	8 GB	2	3.55
Alamo_Medium	2	4 GB	4	2.77
Alamo_Small	1	2 GB	2	1.68
Sierra_Medium	2	4 GB	2	1
Sierra_Small	1	2 GB	3	0.71
Hotel_Small	1	2 GB	6	0.76

Note: † – Name of the site followed by the type of VM.
‡ – Maximum number of available VMs per type



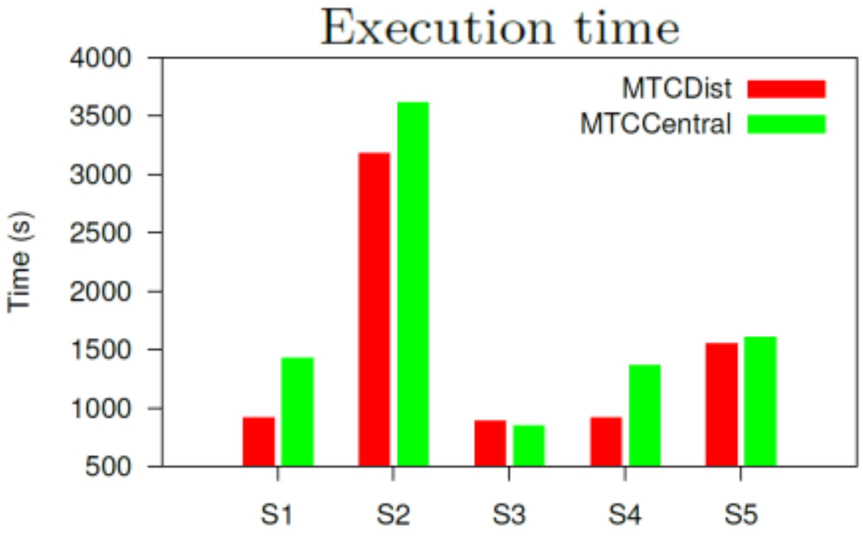
Network (Down/Up)	Alamo	Sierra	Hotel
Alamo	-	10/0.9	15/15
Sierra	11/11	-	11/11
Hotel	18/18	12/1	-
Internal Network (Down/Up)	11/2.3	30/30	45/45

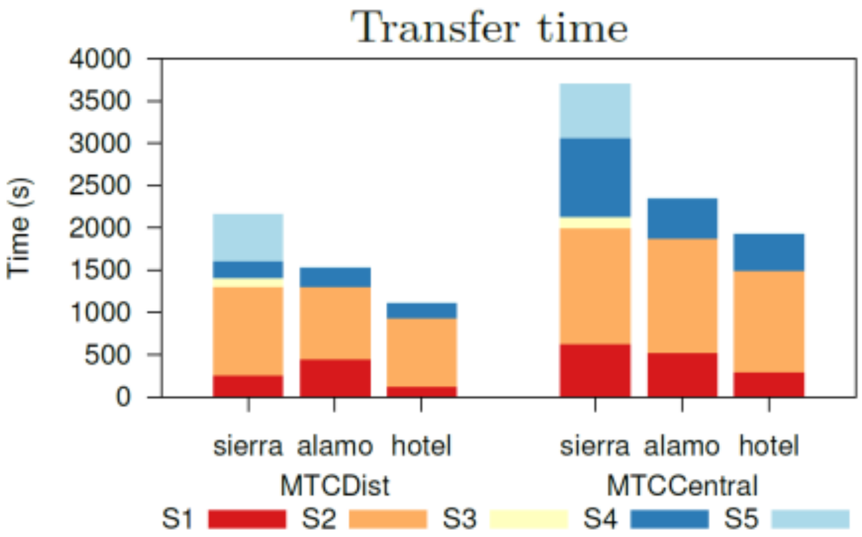
Hotel
NIMBUS
openstack
Sierra
openstack
Alamo

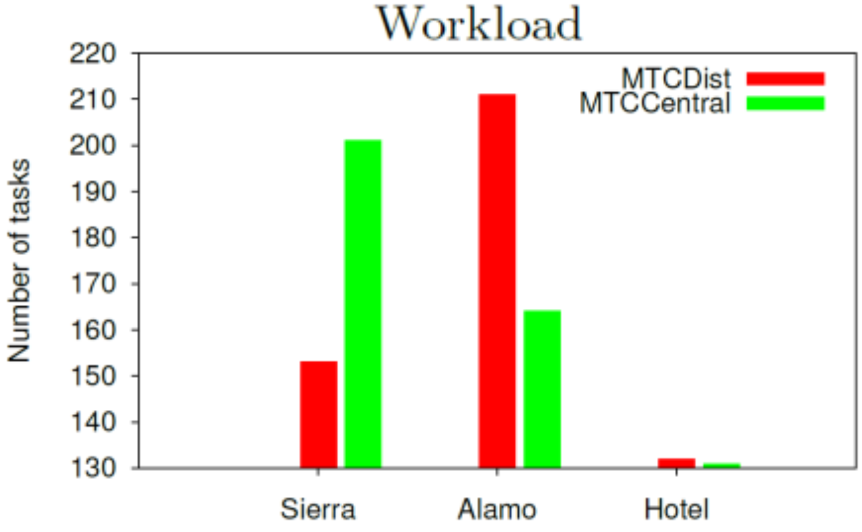
FutureGrid Resources

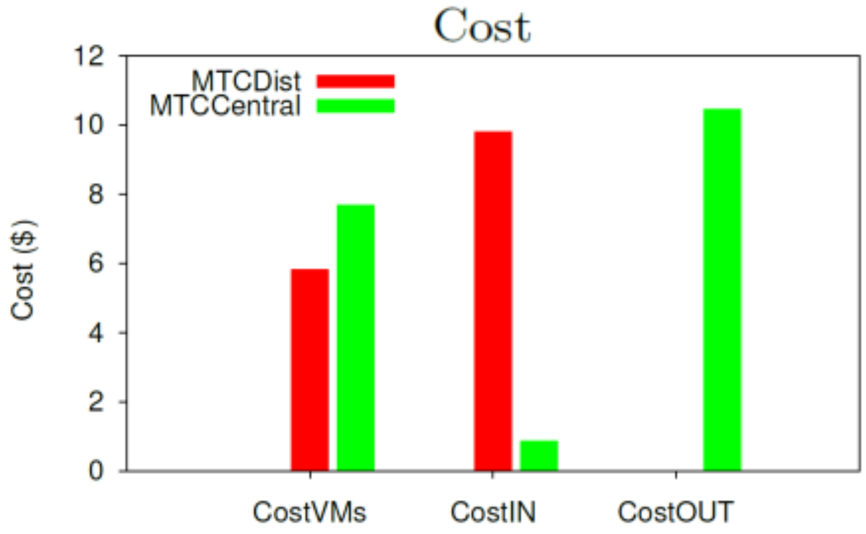
- Sierra – SDSC
- Alamo – TACC
- Hotel – U. Chicago

Minimum Completion Time

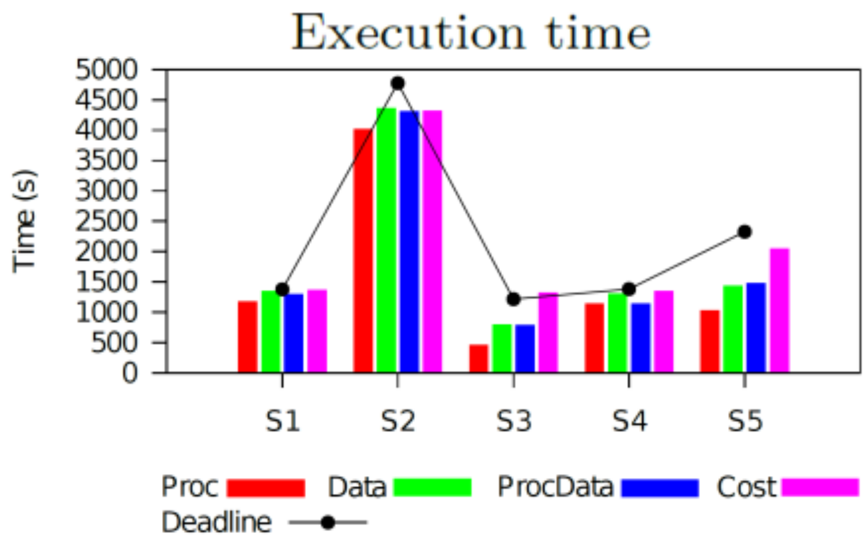


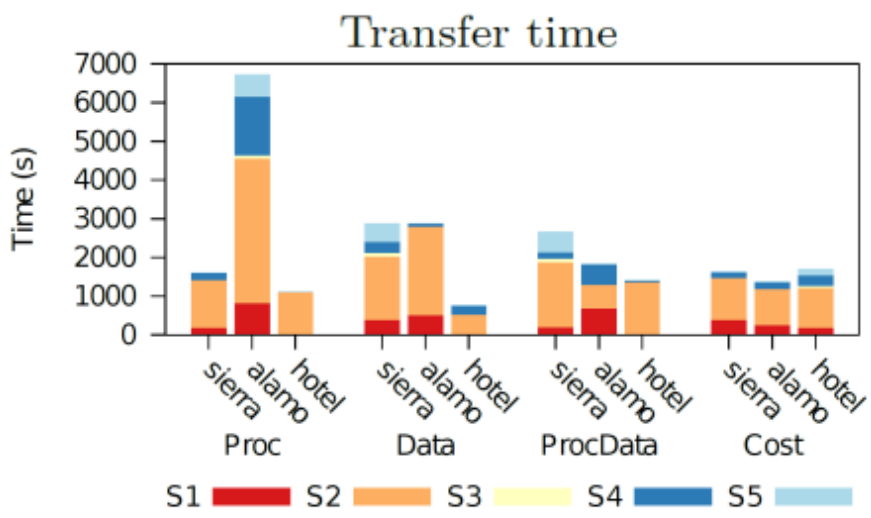


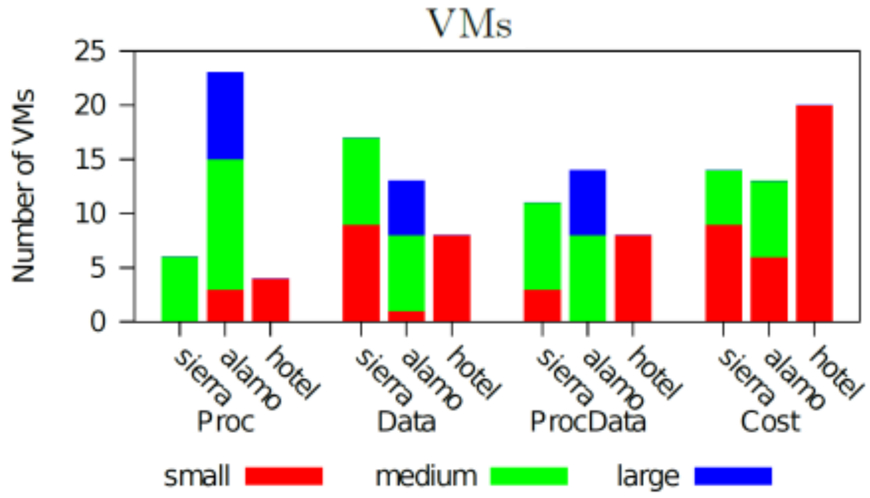




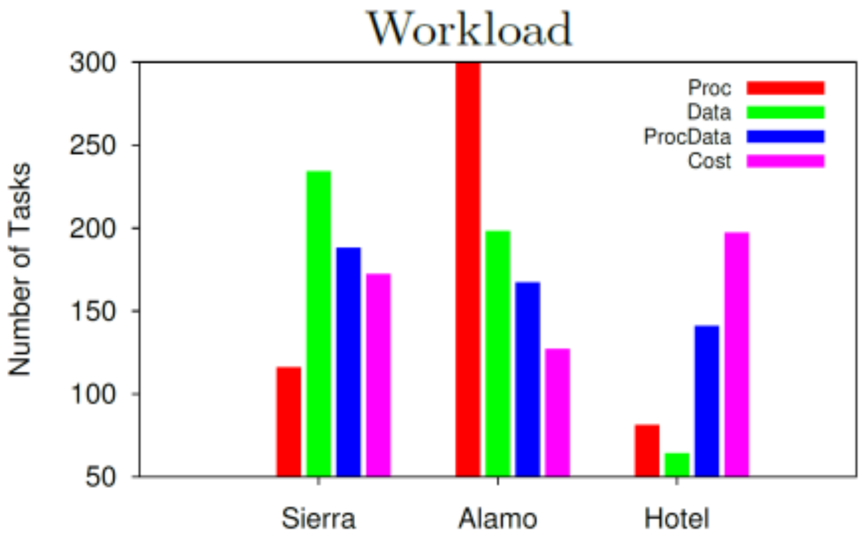
Deadline-based Policies

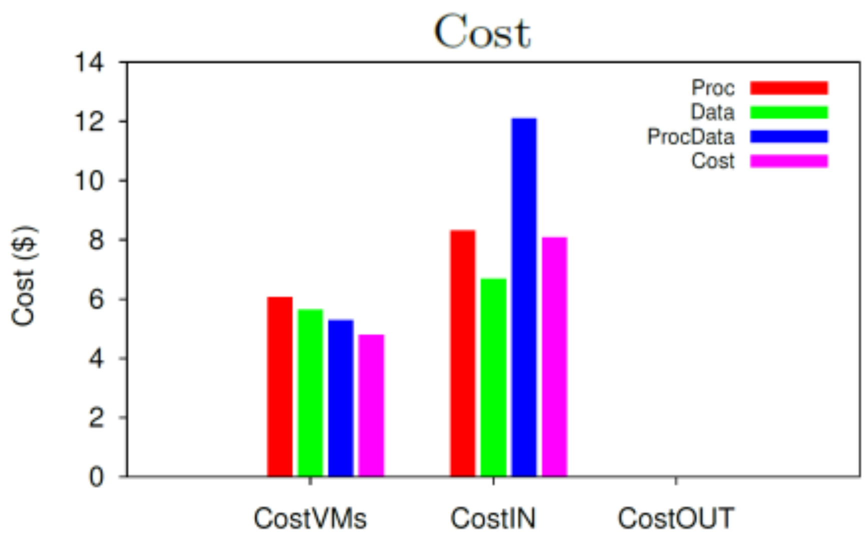






Deadline-based Policies (Cont.)





FEDERATING RESOURCES USING SOCIAL MODELS [IC2E'14]

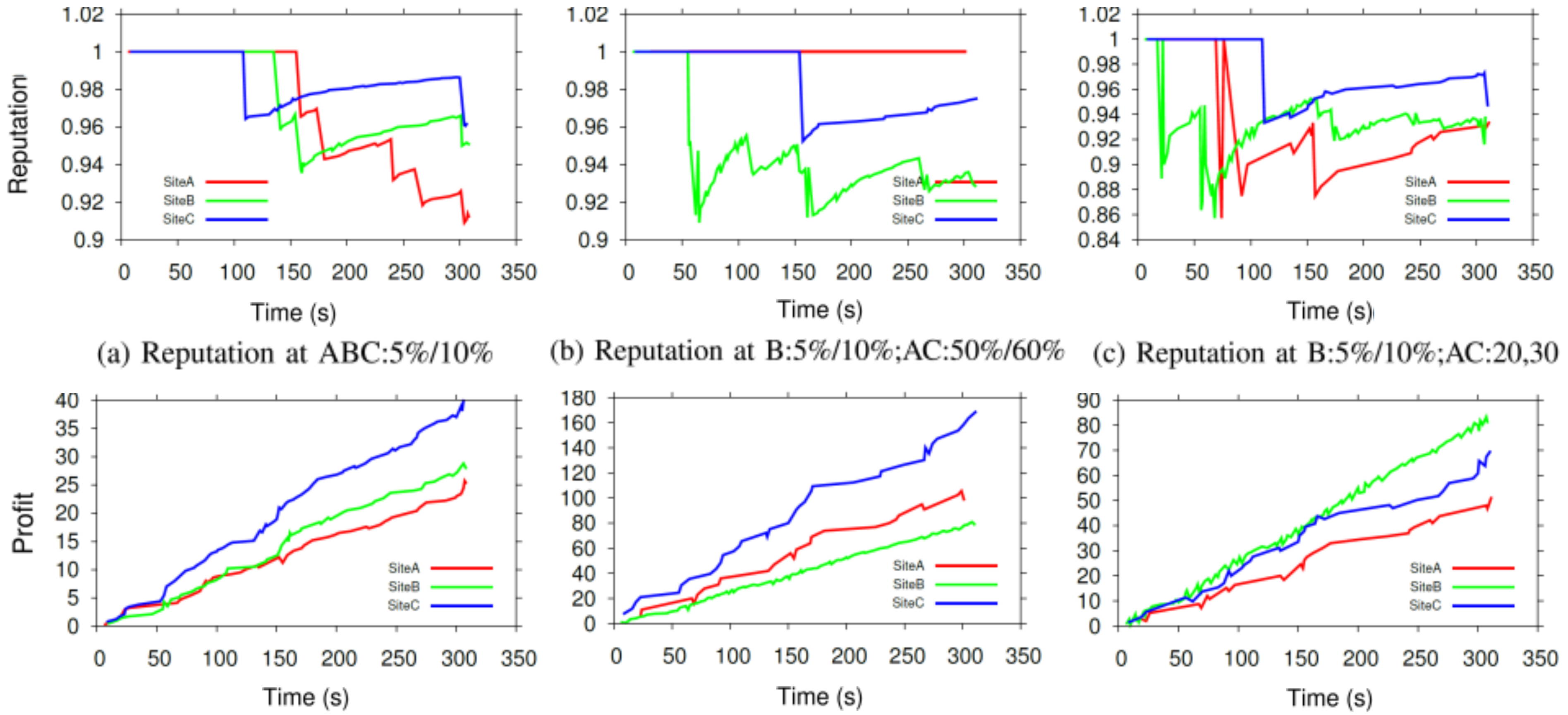
Exchanging Resources in a Federated Cloud

- Consider federation policies and determine their impact on the overall status of each site
- Market model for resource sharing
  - External task vs Local task
  - Heterogeneous tasks - different deadlines and costs
  - Each site decides how much benefit per task (% cost)
  - Federation policy = Auction criteria
- Federation infrastructure between Cardiff (UK) and Rutgers (USA)

Profit and Reputation of Each Site

- Auction Criteria based on Price

Task/Metric	Cost	TTC	Deadline
Red	10	9	12
Black	8	7	10
Blue	6	5	8



(a) Reputation at ABC:5%/10%

(b) Reputation at B:5%/10%;AC:50%/60%

(c) Reputation at B:5%/10%;AC:20,30

(d) Profit at ABC:5%/10%

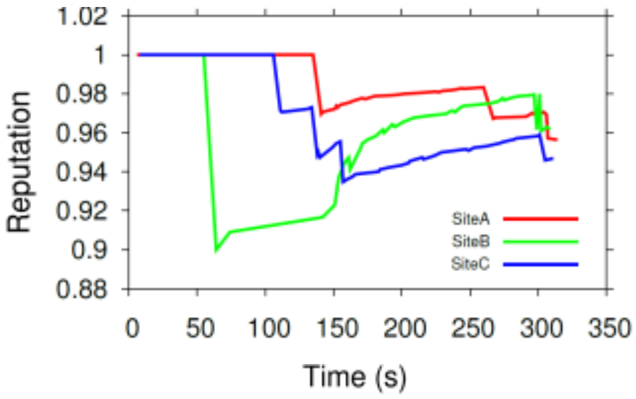
(e) Profit at B:5%/10%;AC:50%/60%

(f) Profit at B:5%/10%;AC:20%/30%

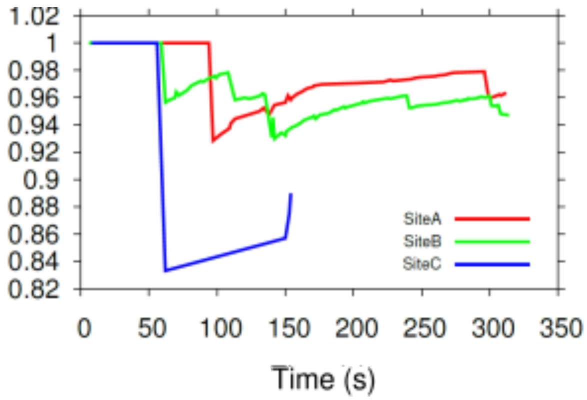
Profit and Reputation of Each Site II

Task/Metric	Cost	TTC	Deadline
Red	10	9	12
Black	8	7	10
Blue	6	5	8

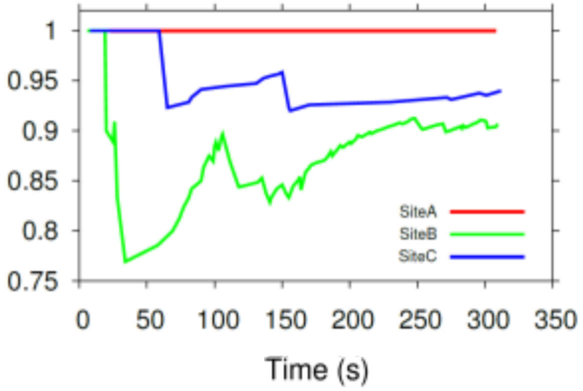
- Auction Criteria based on Price and Reputation



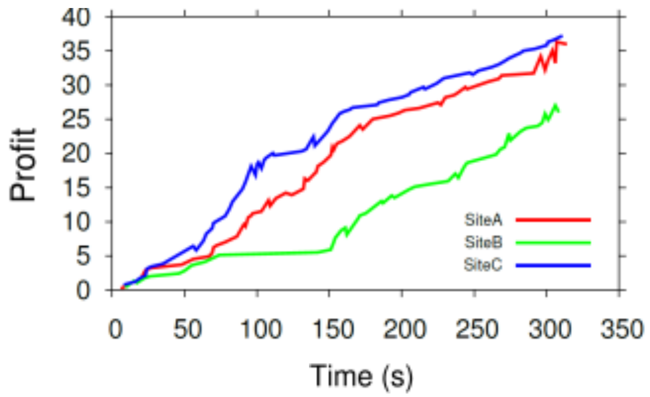
(a) Reputation at ABC:5%/10%



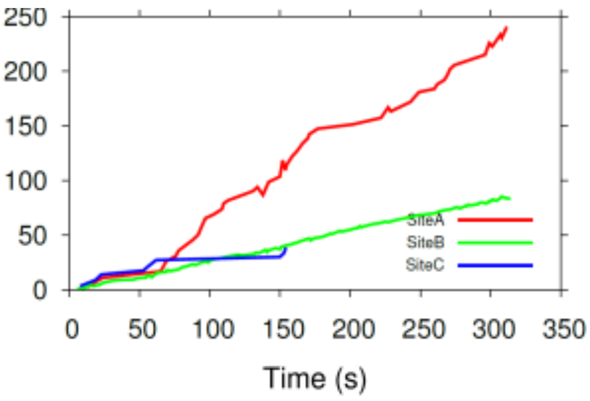
(b) Reputation at B:5%/10%;AC:50%/60%



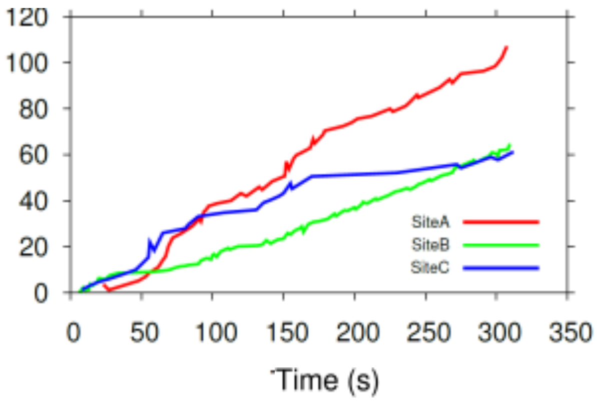
(c) Reputation at B:5%/10%;AC:20,30



(d) Profit at ABC:5%/10%



(e) Profit at B:5%/10%;AC:50%/60%



(f) Profit at B:5%/10%;AC:20%/30%

HPC PLUS CLOUD FEDERATIONS [E-SCIENCE'10]

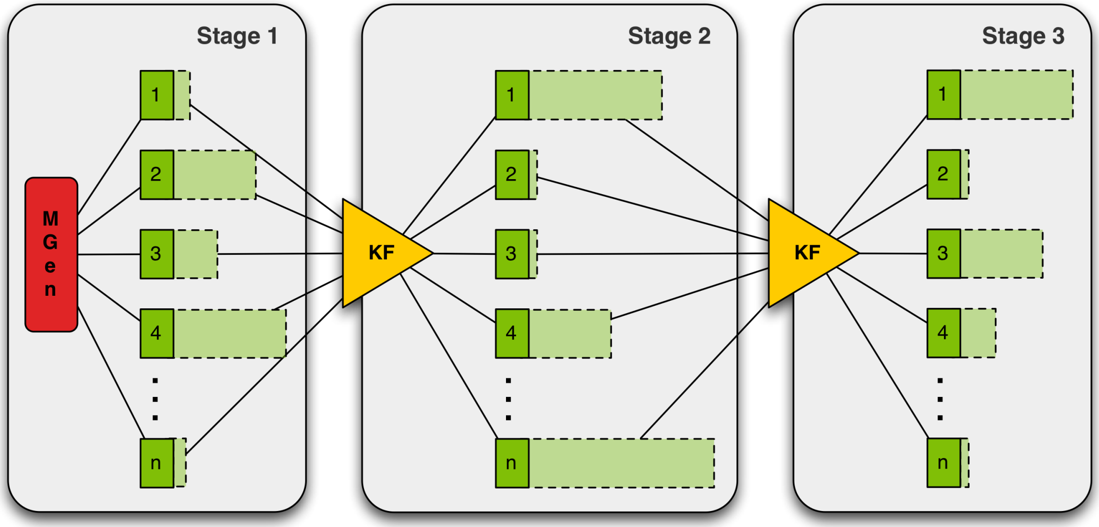
Exploring Hybrid HPC-Grid/Cloud Usage Modes (eScience’09, ScienceCloud’10)

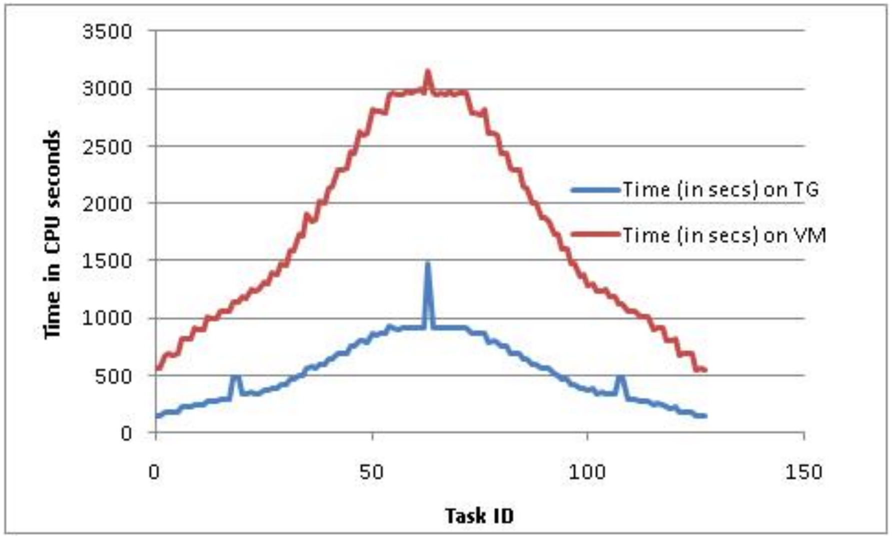
What are appropriate usage modes for hybrid infrastructure?

- Acceleration -- How can Clouds 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





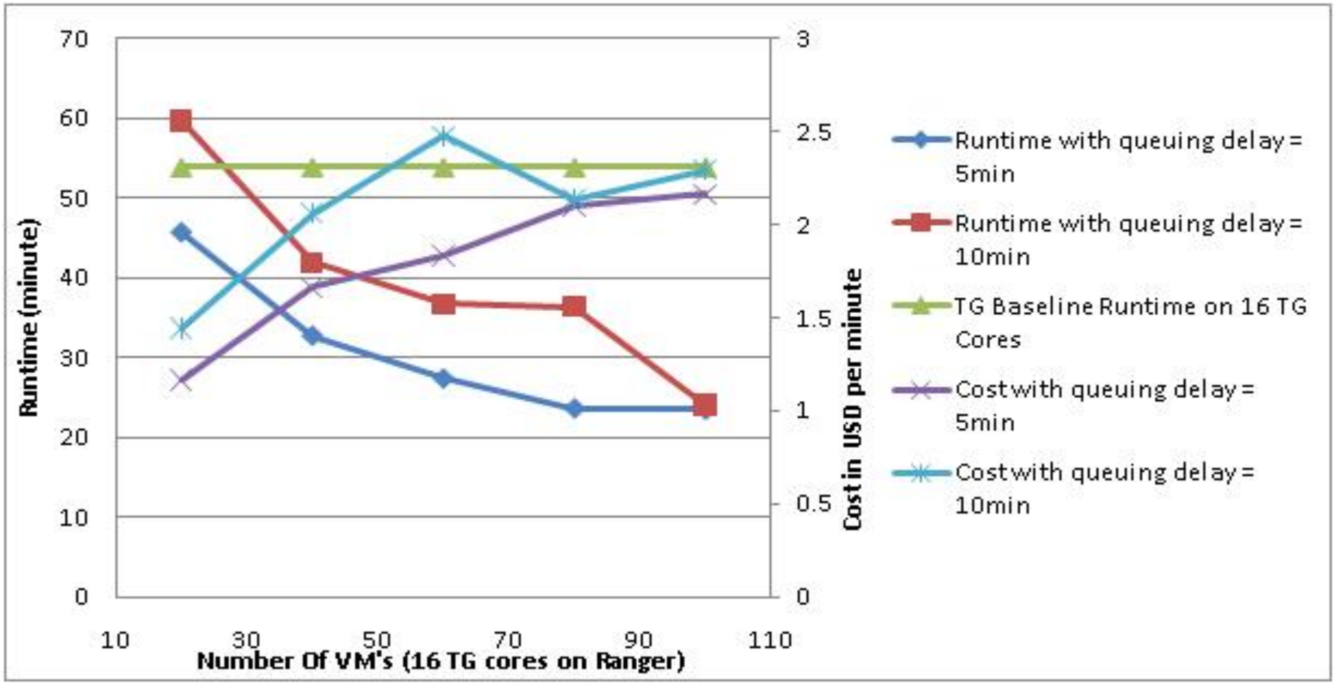
Using Clouds as Accelerators for HPC Grids

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

RUTGERS

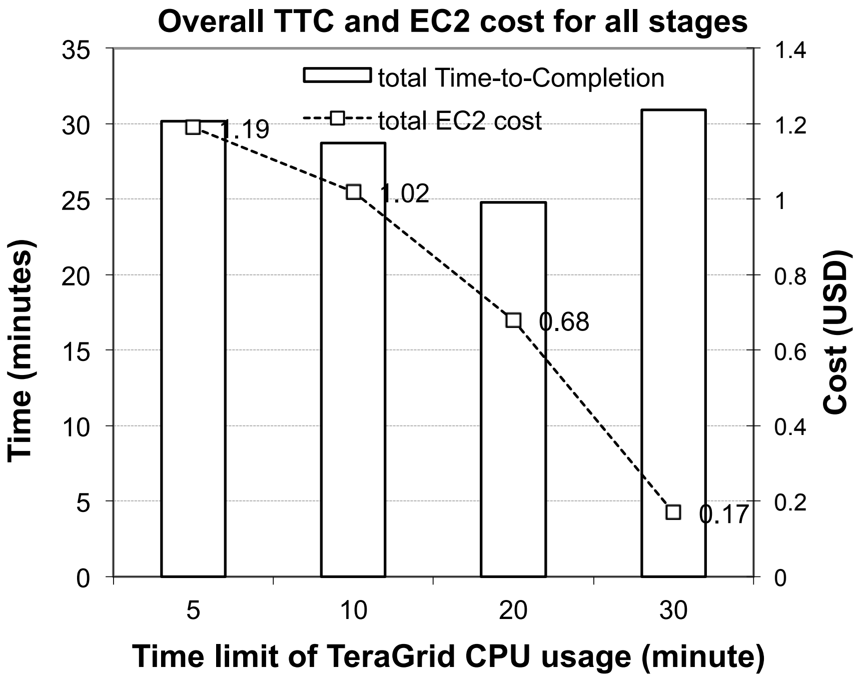
Using Clouds as Accelerators for HPC Grids I

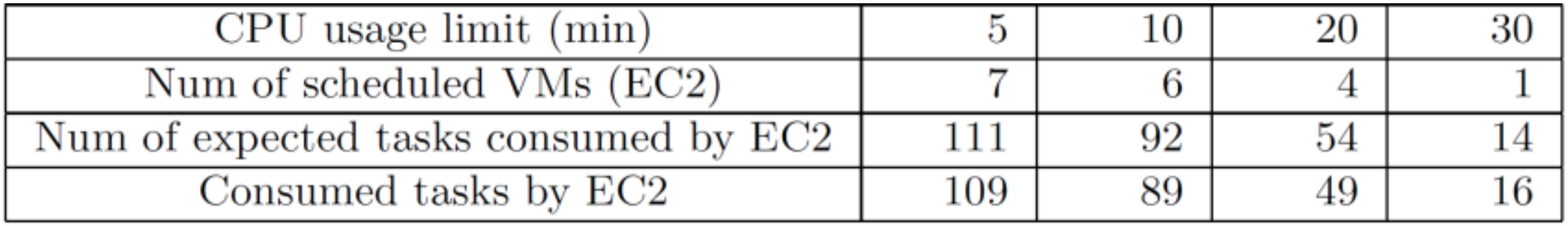
- Acceleration is more notable with more VMs - lower the TTC
- The reduction in TTC is roughly linear
  - Affected by complex interplay between the tasks in the workload and resource availability



Exploring Conservation

- Application deadline 33 minutes (time using only TeraGrid)
- What if we have limited resources on TeraGrid? But we need to keep the same deadline
- Use Cloud to save HPC resources

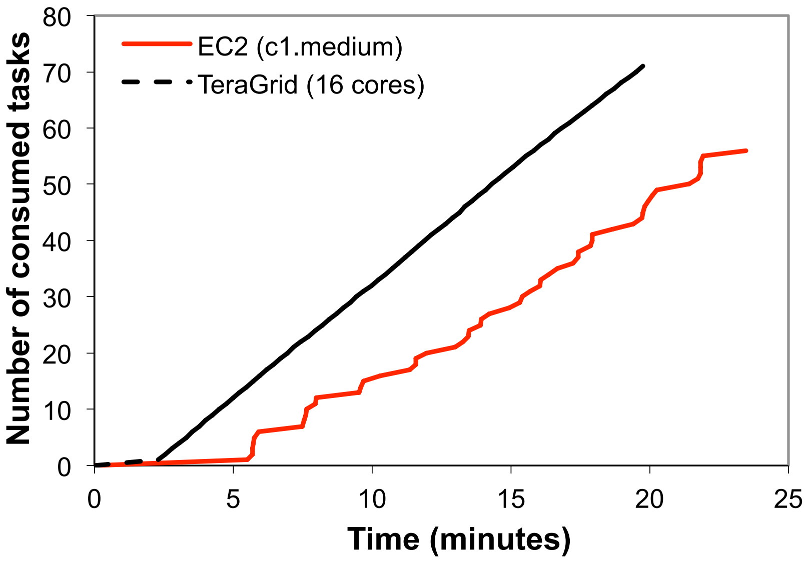




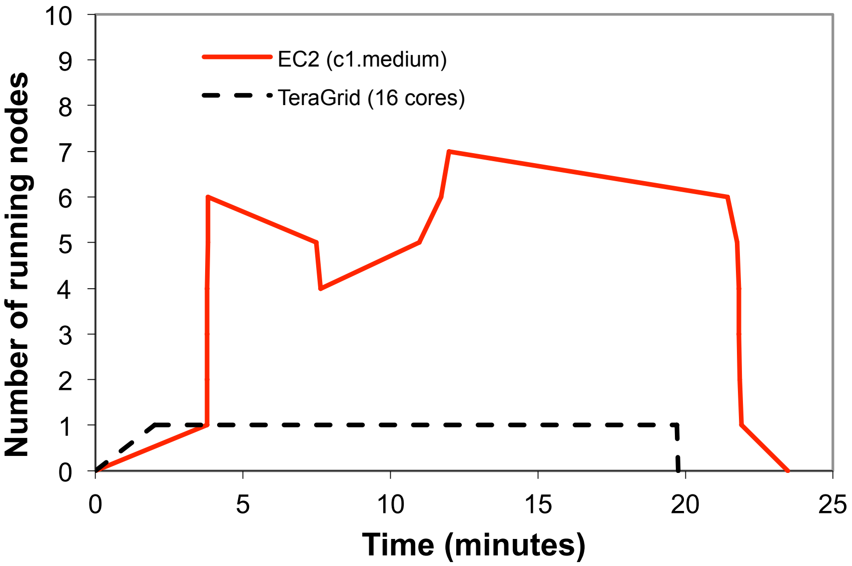
CPU usage limit (min)	5	10	20	30
Num of scheduled VMs (EC2)	7	6	4	1
Num of expected tasks consumed by EC2	111	92	54	14
Consumed tasks by EC2	109	89	49	16

Exploring Resilience

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



(a) Number of consumed tasks



(b) Number of nodes

Conclusions

- Complex application workflows necessiciate software defined federated platforms that integrated heterogeneous cloud services
- Provisioning and federating an appropriate mix of resources on-the-fly is essential and non-trivial
- Autonomics can provide the abstractions and mechanism to manage complexity
  - Separation + Integration + Automation
- However, there are implications
  - Added uncertainty
  - Correctness, predictability, repeatability
  - Validation
  - New formulations necessary….

The CometCloud Team

Moustafa AbdelBaky

PhD student, Dept. of Electrical & Computer Engr.

Rutgers University

Email: moustafa@cac.rutgers.edu

Mengsong Zou

PhD Student, Dept. of Computer Science

Rutgers University

Email: mz228@cac.rutgers.edu

Javier Diaz-Montes, Ph.D.

Assistant Research Professor, Dept. of Electrical & Computer Engr.

Rutgers University

Email: javidiaz@rdi2.rutgers.edu

Manish Parashar, Ph.D.

Prof., Dept. of Electrical & Computer Engr.

Director, Center for Autonomic Computing

Rutgers University

Email: parashar@rutgers.edu

And many collaborators….

CometCloud: http://cometcloud.org