



Exploring Autonomics for Federated Clouds using CometCloud

Rutgers University: Moustafa AbdelBaky, Javier Diaz-Montes, Mengsong Zou, Manish Parashar

Cardiff University: Omer Rana, Tom Beach, Ioan Petri

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

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: AN AUTONOMIC CLOUD ENGINE

<http://cometcloud.org>

RUTGERS **CARDIFF UNIVERSITY**
PRIFYSGOL CAERDYDD

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>

RUTGERS **CARDIFF UNIVERSITY**
PRIFYSGOL CAERDYDD

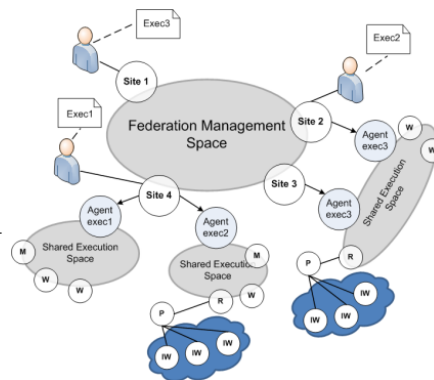
Many Applications

- Medical informatics (Master/worker, workflow)**
 - Xin Qi, Fuyong Xing, Meghana Ghadge, Ivan Rodero, Moustafa Abdelbaky, Manish Parashar, Evita Sadimin, David J. Foran, Lin Yang, "Content-based image retrieval on imaged peripheral blood smear specimens using high performance computation" 15th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Nice, France 2012.
 - Lin Yang, Hyunjoon Kim, Manish Parashar, and David J. Foran, "High throughput landmark based image registration using cloud computing," 14th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Toronto, Canada, Sep. 18-22, 2011.
 - Xin Qi, Hyunjoon Kim, Fuyong Xing, Manish Parashar, David J. Foran and Lin Yang, "The analysis of image texture feature robustness using CometCloud," 14th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Toronto, Canada, Sep. 18-22, 2011.
 - "Investigating the use of cloudbursts for high-throughput medical image registration, GRID2009, Banff, Canada, Oct. 2009.
- Molecular dynamics & drug design (MapReduce)**
 - "Accelerating MapReduce for Drug Design Applications: Experiments with Protein/Ligand Interactions in a Cloud," 2009.
 - "Asynchronous Replica Exchange for Molecular Simulations, Journal of Computational Chemistry, 29(5), 2007.
- PDEs solvers using synchronous and asynchronous iterations**
 - Hyunjoon Kim, Yaakoub el-Khamra, Shantenu Jha, and Manish Parashar, "Exploring Adaptation to Support Dynamic Applications on Hybrid Grids-Clouds Infrastructure," 1st Workshop on Scientific Cloud Computing (ScienceCloud), in conjunction with the ACM International Symposium on High Performance Distributed Computing (HPDC), Chicago, Illinois, June 20-25, 2010.
 - Hyunjoon Kim, Yaakoub el-Khamra, Shantenu Jha, and Manish Parashar, "An Autonomic Approach to Integrated HPC Grid and Cloud Usage," the 5th IEEE International Conference on e-Science, Oxford, UK, Dec. 2009.
 - A decentralized computational infrastructure for grid based parallel asynchronous iterative applications, J. of Grid Computing, 2006.
- Others...**
 - Reservoir simulator with Ensemble Kalman Filter (Workflow)
 - Analytics applications from Xerox (Workflow)
 - Asynchronous Replica Exchange (Master/worker)
 - Manish Parashar, Moustafa AbdelBaky, Ivan Rodero, and Aditya Devarakonda, "Cloud Paradigms and Practices for CDS&E", CAC Research Report, 2012

AUTONOMICS FOR CLOUD FEDERATIONS

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



RUTGERS

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYB

Managing Autonomics

- **Autonomic manager** manages workflows, benchmarks application and provision resources.
- **Adaptivity manager** monitors application performance and adjusts resource provisioning.
- **Grid/Cloud/Cluster agent** manages local cloud resources, accesses task tuples from CometCloud and gathers results from local workers so as to send them to the workflow (or application) manager.

RUTGERS

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYB

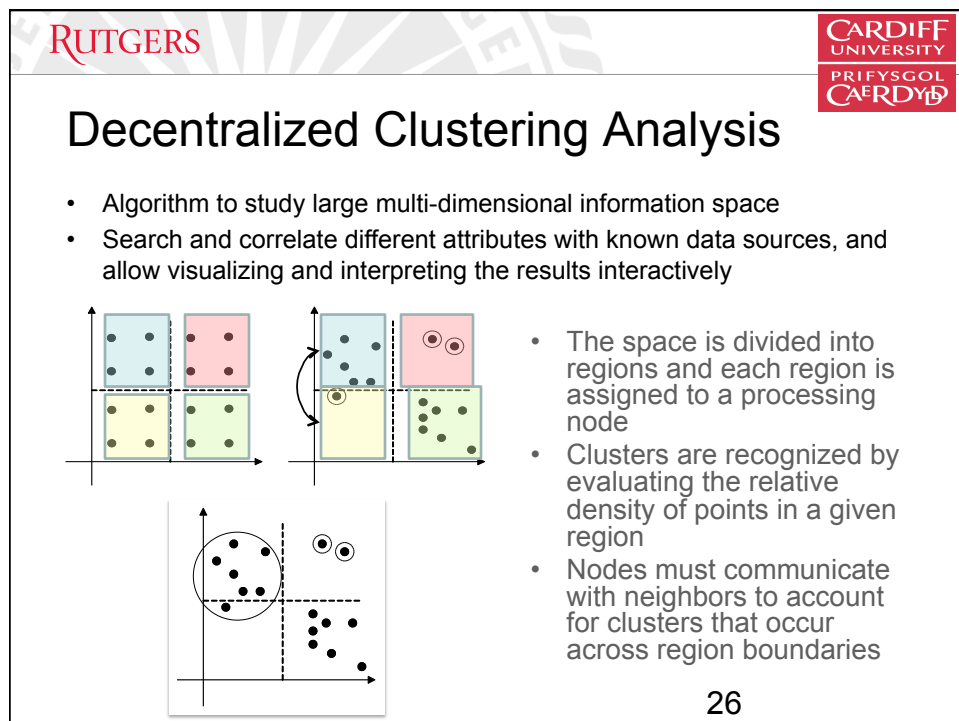
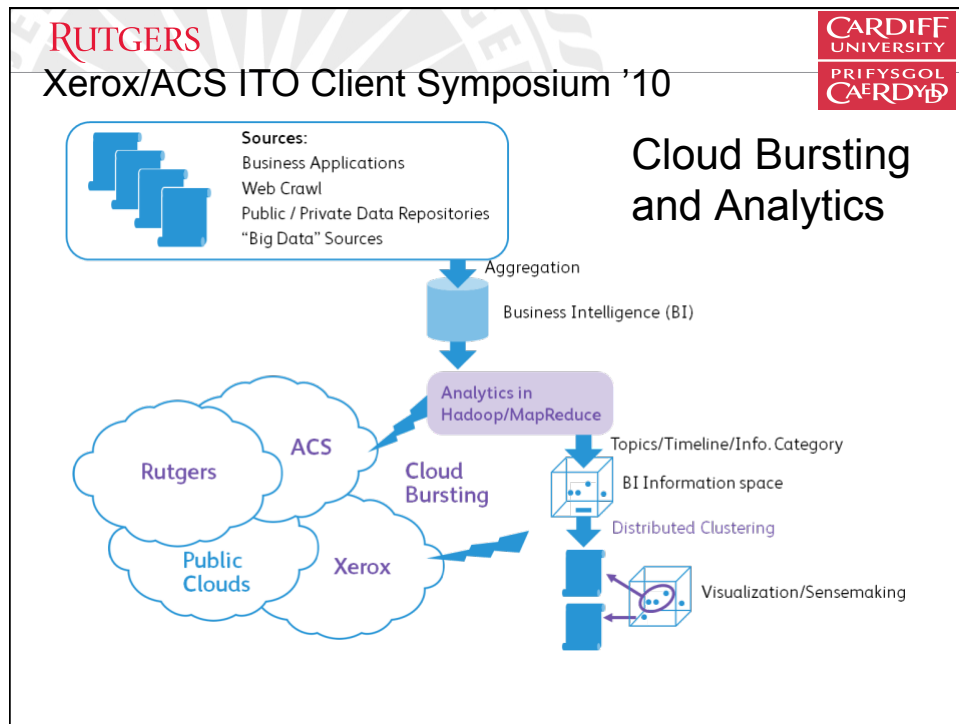
The Autonomics Loop

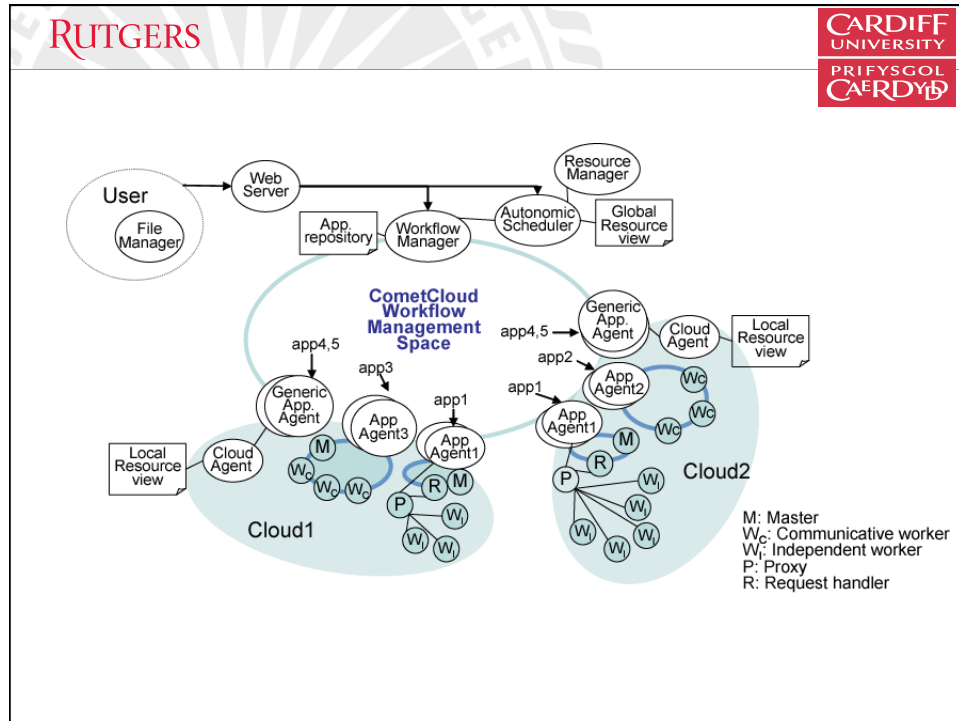
- **Sampling and estimation**
 - Estimate runtime of all tasks on all resource classes
- **Scheduling and provisioning**
 - Schedule each task to the most appropriate resource class based on policy, constraints or the objective and the number of nodes per resource class is decided
- **Monitor and adaptation**
 - The actual runtime of each task is monitored and scheduling decision is adapted if the runtime is different from the estimated runtime enough to affect objective.

Objectives & Constraints

- Deadline
 - Time constraint to complete an application
 - To select the fastest resource class for each task and to decide the number of nodes per resource class based on the deadline.
- Budget
 - Budget constraint to complete an application
 - When a budget is enforced on the application, the number of allocable nodes is restricted by the budget.
- Economics + deadline
 - Resource class can be defined as the cheaper but slower resource class that can be allocated to save cost unless the deadline is violated.
- Benefit / profit

CASE STUDY: WORKFLOW MANAGEMENT [WITH XEROX]



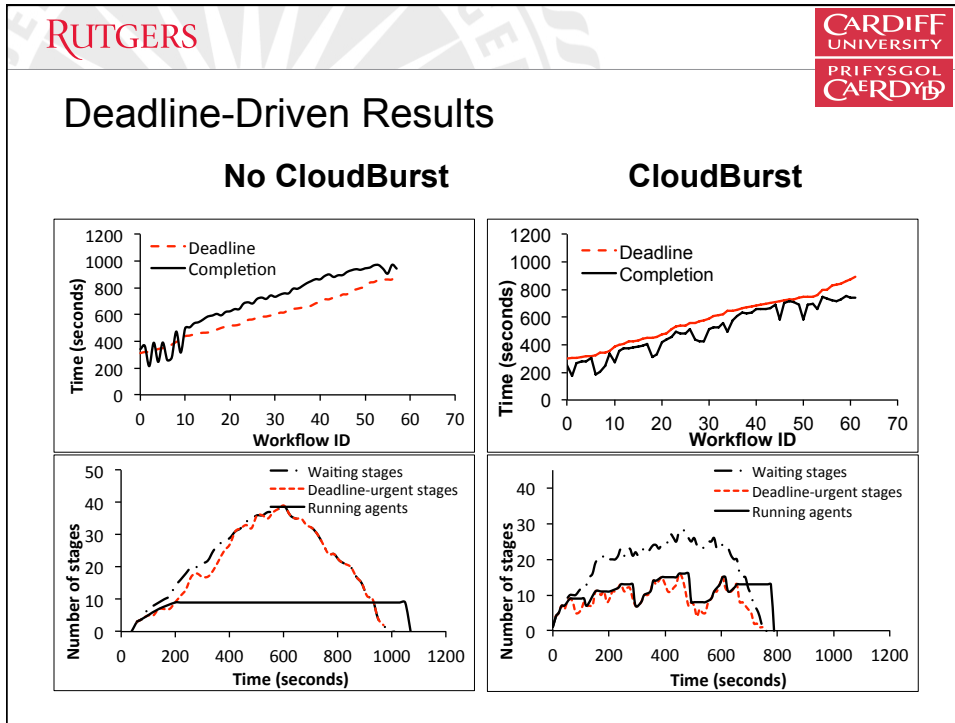


RUTGERS **CARDIFF UNIVERSITY**
PRIFYSGOL CAERDYDD

Experiment

- Deadline-driven workflows
 - Each workflow has 3 different stages of the DOC application
 - Each stage of the workflow has a different execution time
 - Each stage is a task which is completed by 1 agent and 2 workers
 - Deadline for a workflow is set to average 300 seconds (100 seconds per stage)
 - Submitting workflows every 10 seconds during initial 600 seconds of experiment
 - CloudBurst – No CloudBurst
- Resources
 - Rutgers cluster has 27 machines
 - Amazon EC2 - c1.medium instance type

32

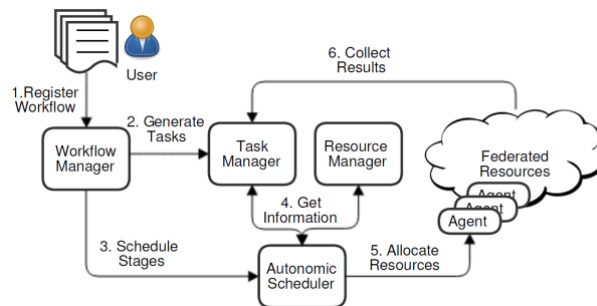


RUTGERS **CARDIFF UNIVERSITY**
PRIFYSGOL CAERDYDD

**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

FutureGrid Resources

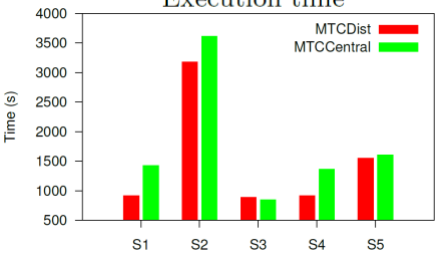
- Sierra – SDSC
- Alamo – TACC
- Hotel – U. Chicago



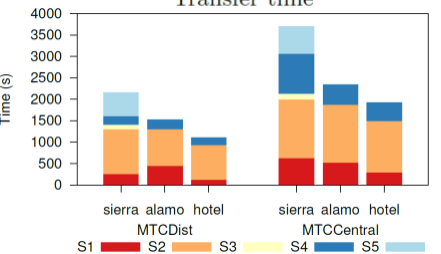


Minimum Completion Time

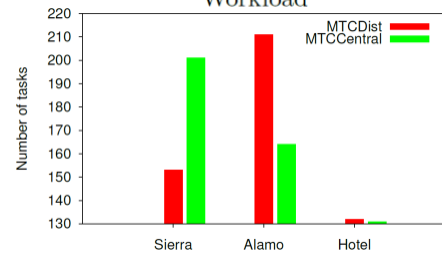
Execution time



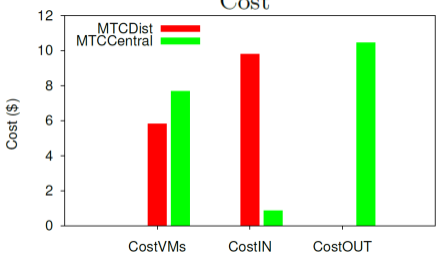
Transfer time

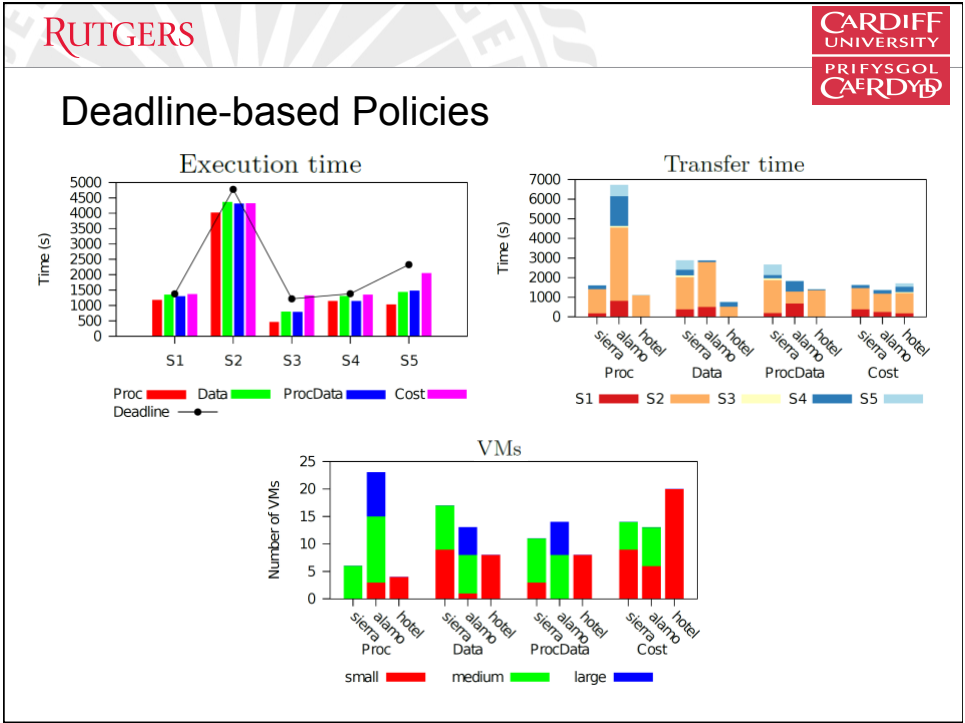


Workload



Cost





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)

Implementation

- Requirements for a site to join the federation:
 - Java support
 - Valid credentials (authorized SSH keys)
 - Configure some parameters (i.e. address, ports, number of workers)

- Resources

Resources	Cardiff	Rutgers
Machines	12	32
Core per Machine	12	8
Memory	12 GB	6 GB
Network	1 GbE	Infiniband

- Indiana site
 - Uses FutureGrid (OpenStack, Infiniband interconnect, 2 cores/machine with 4GB memory) – also supports Cloudmesh Teefaa and Rain

Outsourcing Policies

- Tasks are *discriminated* based on their origin to decide the offered price as well as resource availability
 - Local task: task request submitted by a local user
 - External task: task request submitted by a remote user
- Each site attempts to maximize revenue from external tasks while preserving QoS of local tasks
- **Provider Policy:**
 - Local tasks are always accepted
 - Remote tasks are accepted only if the $TTC < \text{Deadline}$
- **Market Policy:** All tasks go to a common marketplace looking for offers from every site interested in executing them

I. Petri, T. Beach, M. Zou, J. Diaz-Montes, O. Rana and M. Parashar, "[Exploring Models and Mechanisms for Exchanging Resources in a Federated Cloud](#)", IEEE international conference on cloud engineering (IC2E 2014), Boston, Massachusetts, March 2014.

EnergyPlus and Building Optimisation

- Real time optimisation of building energy use
 - sensors provide readings within an interval of 15-30 minutes,
 - Optimisation run over this interval
- The efficiency of the optimisation process depends of the capacity of the computing infrastructure
 - deploying multiple EnergyPlus simulations
- Closed loop optimisation
 - Set control set points
 - Monitor/acquire sensor data + perform analysis with EnergyPlus
 - Update HVAC and actuators in physical infrastructure



Sporte²
Energy Efficiency for European Sport Facilities

Instrumented Facility

CENTRO SPORTIVO FIDIA ROMA (<http://www.asfidia.it/>)







Pool (indoor) – size: 25m x 16m, depth: 1,60m to 2,10m, Capacity: 760 m³
 Learning Pool (indoor) – size: 16m x 4 m, depth: 1m, Capacity: 64 m³
 1 Gym (indoor) provided of electric equipment (electric bicycles, etc...)
 1 Fitness room (indoor) size: 18m x 9m x 3m, Volume: 486m³
 1 Volleyball court (indoor) – size: 40m x 28m x 8m, Volume: 8960 m³
 2 Tennis/Five-a-side courts (outdoor, with changing rooms) – size: 30m x 20m

INPUT FIDIA Scenario 1

Time:	13:31:02
Date:	2014-02-04
Occupancy:	25 2014-02-04T13:29:36Z
Indoor Relative Humidity(%):	88.2 2014-02-04T13:29:36Z
Current Room Temperature(deg.C):	24.05 2014-02-04T13:29:36Z
Pool Water Temperature(deg.C):	29.39 2014-02-04T13:29:37Z
Supply Air Flow Rate(m3/s):	6.69 2014-02-04T13:29:36Z
Supply Inlet Air Temperature(deg.C):	23.89 2014-02-04T11:29:37Z

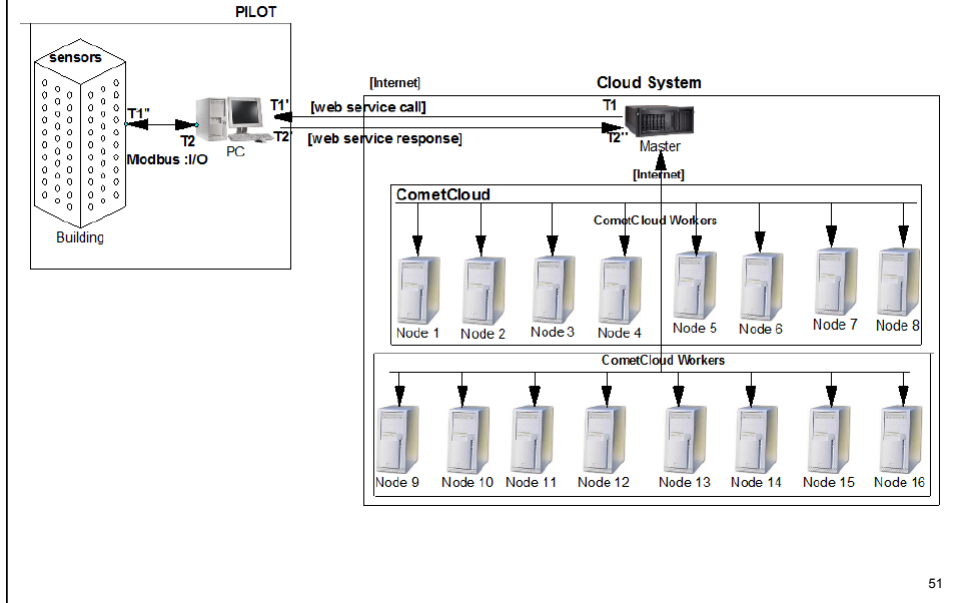
OUTPUT Optimisation results are as follow:

Type of Set Points	Supply Air Flow Rate(kg/s)	Supply Inlet Air Temperature(deg.C)
Initial Set Points	2.954	23.899
Optimized Set Points	5.784	4.827

Optimisation Results	Predicted Results(Initial Set Points)	Optimised Results(CU Solution)
Thermal Energy Consumption(Kwh)	38.333	38.242
Electricity Consumption(Kwh)	0.088	0.090
PMW	0.359	2.061

SetPoint changed to->4.827

EnergyPlus and Building Optimisation



51

Federation constraints

Two metrics:

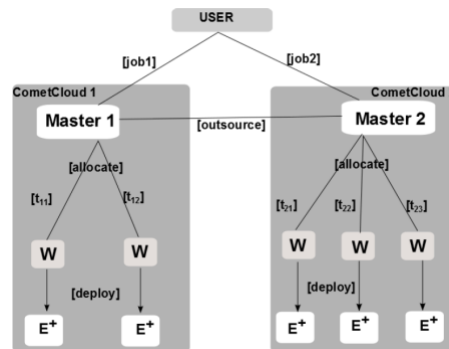
- Time to complete
- Results quality

Trading quality of results vs. overall simulation time

cost function: $f(X) : C \rightarrow R$ where C is a set of constraints (cost, deadline) and R is a set of decisions based on the existing constraints C .

•Each Master decides how to compute the received job :

- (i) where to compute the tasks: (a) Single CometCloud or (b) federated CometCloud;
- (ii) how many combinations to run giving the deadline received from the user.



Evaluation

- In our experiments we use two different configurations
 - (a) *single cloud context* where all the tasks have to be processed locally
 - (b) *federation cloud context* where the sites have the option of outsourcing tasks to remote sites.
- We use as inputs for our calculation
 - (i) *CPU time of remote site* as the amount of time spent by each worker to compute the tasks and
 - (ii) *storage time on remote site* as the amount of time needed to store data remotely.
- All the costs have been calculated in £ derived from Amazon EC2 cost.

53

Experiment 1: Job completed

Table III: Input Parameters: Experiment 1

P1	P2	P3	P4	Deadline
{16,18,20,22,24}	{0,1}	{0,1}	{0,1}	1 Hour

Table IV: Results: Experiment 1

	Single Cloud	Federated Cloud
Nodes	3	6
Cost	£ 0	£ 7.46
Tasks	38	38
Deadline	1 hour	1 hour
Tuples exchanged	-	15
CPU on remote site	-	5626.45 Sec
Storage on remote site	-	1877.10 Sec
Completed tasks	34/38	38/38 in 55min 40s

- the federation site has two options: (i) run tasks on the local infrastructure (single cloud case) or (ii) outsource some tasks to a remote site (federation cloud case)
- A corresponding deadline of 1 hour, only 34 out of 38 can be completed.
- In the federation in 55 minutes by outsourcing 15 to the remote site.
- The process of outsourcing has an associated cost of 7.46 £

54

Experiment 2: Job uncompleted:

Table V: Input Parameter: Experiment 2

P1	P2	P3	P4	Deadline
{16,17,18,19,20,21,22,23,24}	{0,1}	{0,1}	{0,1}	1 Hour

Table VI: Results: Experiment 2

	Single Cloud	Federated Cloud
Nodes	3	6
Cost	0	£ 7.90
Tasks	72	72
Deadline	1 hour	1 hour
Tuples exchanged	-	15
CPU on remote site	-	5637.27 Sec
Storage on remote site	-	1869.41 Sec
Completed tasks	37/72	58/72

- In the context of single cloud federation (3 workers) only 37 out of 72 tasks are completed within the deadline of 1 hour.
- Exchanging 15 tuples between the two federation sites, with increased cost for execution and storage.

55

Experiment 3: Job uncompleted—parameters ranges extended:

Table VII: Input Parameters: Experiment 3

P1	P2	P3	P4	Deadline
{14,15,16,17,18,19,20,21,22,23,24,25,26,27}	{0,1}	{0,1}	{0,1}	1h 30 min

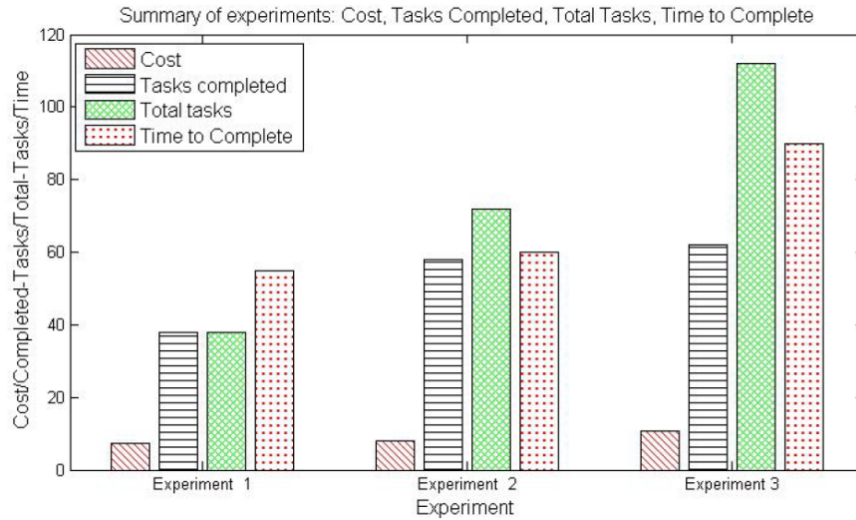
Table VIII: Results: Experiment 3

	Single Cloud	Federated Cloud
Nodes	3	6
Cost	0	£ 10.70
Tasks	112	72
Deadline	1 h 30 min	1 h 30 min
Tuples exchanged	-	22
CPU on remote site	-	7983.74 sec
Storage on remote site	-	2687.15 sec
Completed tasks	42/112	62/112

- we extend the deadline associated to 1 hour and 30 minutes
- when using the federation to outsource a percentage of tasks we observe that the number of tasks completed increases to 62

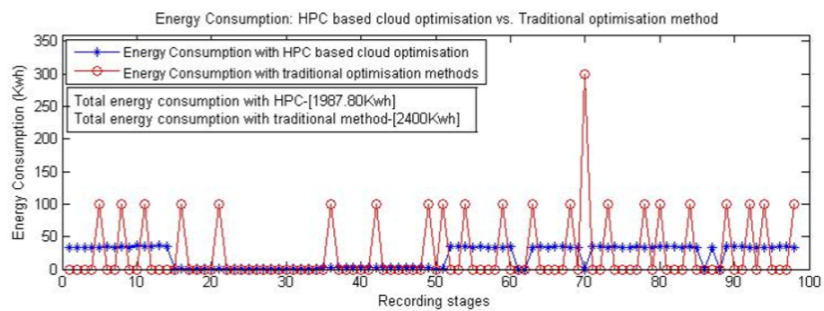
56

Summary of results



57

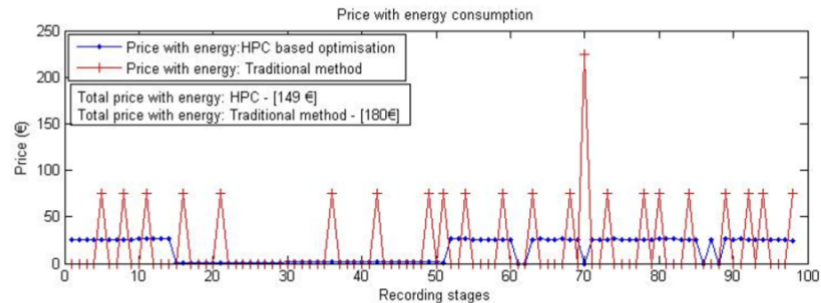
Validation – Energy in FIDIA pilot



- For the HPC cloud based optimisation the energy consumption fluctuates over the interval of [0-38] Kwh whereas tradition optimisation [0-300] Kwh
- HPC cloud based optimisation assumes a continuous adaptation based on the values read from sensors in intervals of [15min,30min,1h]

58

Price with energy in FIDIA



- The costs with energy are significantly reduced when using HPC cloud optimisation
- This amount represents approximately 39% out the total cost with energy of the FIDIA pilot

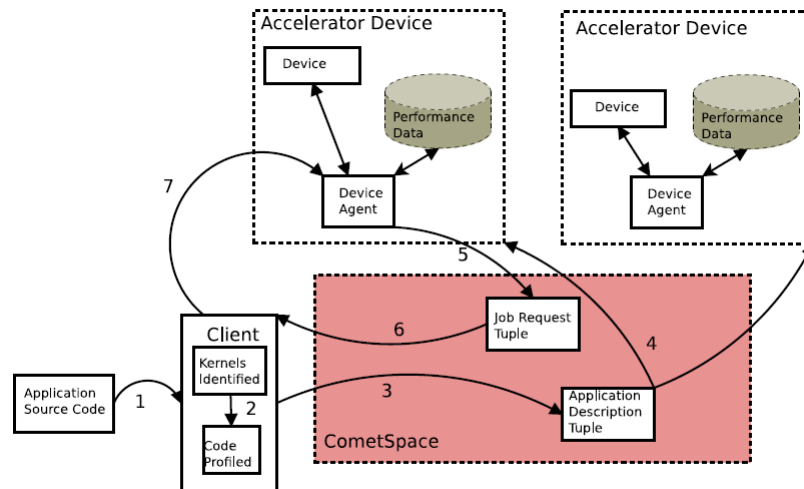
◀ ▶ ⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿

59

Heterogeneous CometCloud

- Workers can have different characteristics and therefore a more intelligent selection is necessary.
- Selection by worker capability is the first step and works well.
- There scope to gain performance improvement for more fine grained decision making.
- Consider how specialist workers perform based on prior execution history
 - Subsequent tune task allocation based on worker capability

CometCloud + GPU (NVIDIA Tesla & Kepler)



Code Analysis

- Investigate “kernels” in code that can be ported to GPUs
- Match kernel properties to capabilities of acceleration devices
- Decision made on simple rules
 - Could use software versions or hardware properties (e.g. CUDA5 compatibility)
- Device replies with
 - Estimated time
 - Estimated time to availability

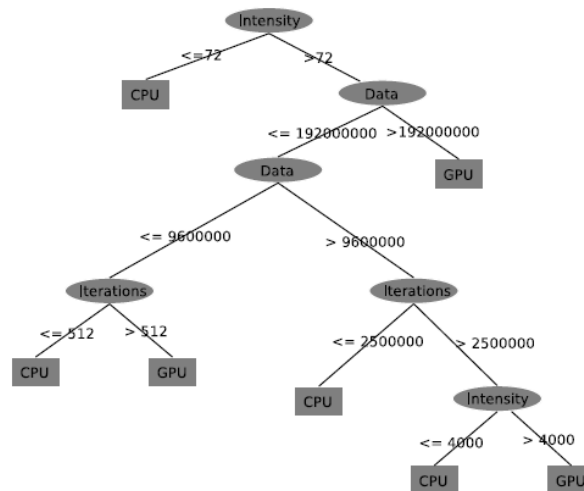
Metrics Considered

- Total number of iterations
- Intensity: mathematical operations per iteration
- I/O count: number of memory accesses (read/write) per iteration
- Number of branching operations per iteration
- Size of data loaded to/from device

Intensity	Highest Precision	Branching	Memory Access	Memory Write	Iterations	Data Moved	Performance	Device
815611	DOUBLE	4096	180388	19267	4096	536870912	2.546	Device
1774115	DOUBLE	8192	393393	36488	8192	2147483648	13.046	Device

Table 1: An Instance of Kernel Performance Data

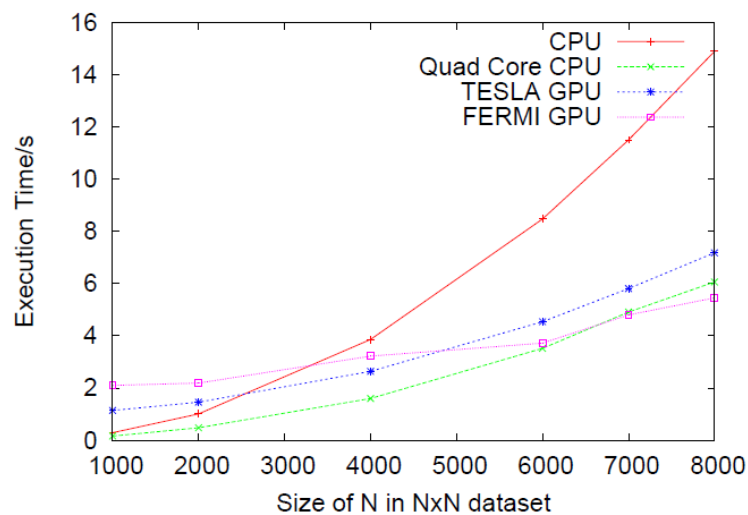
J48 Decision Tree



Brokering Mechanism

- A client enables a user to submit applications to CometSpace
 - contains a code profiler that enables kernels to be extracted from the application code submitted by a user.
- A device agent acts as an interface between the acceleration device and CometSpace
 - device agent must store properties of the acceleration device
 - store data about prior execution history on the device
- A database of performance data
 - used by device agent to undertake performance predictions as part of the matchmaking process

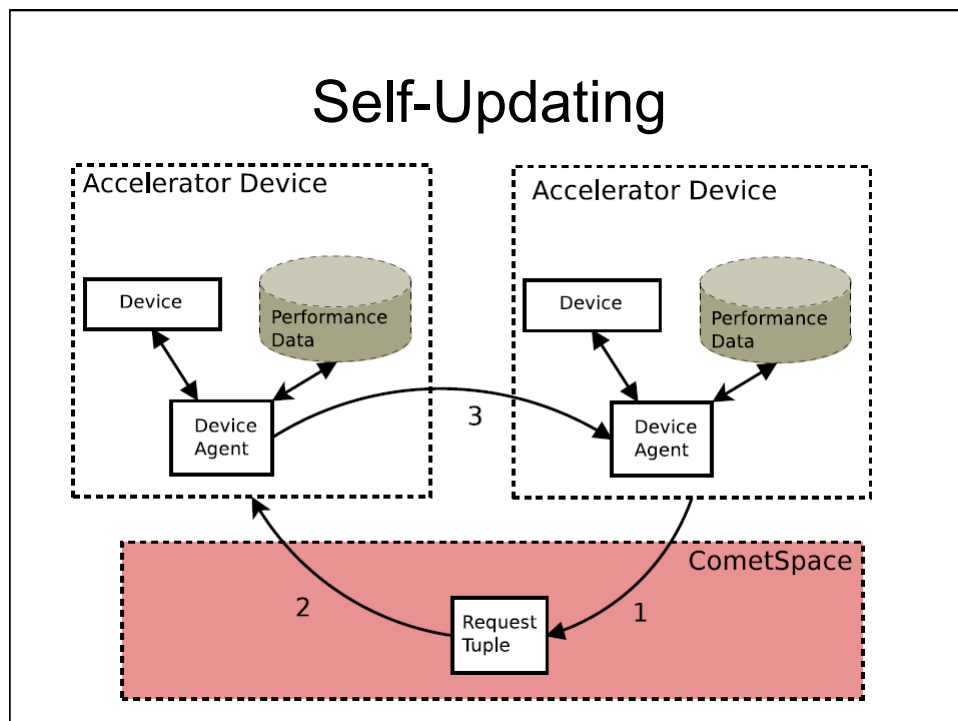
“Canny” edge detector

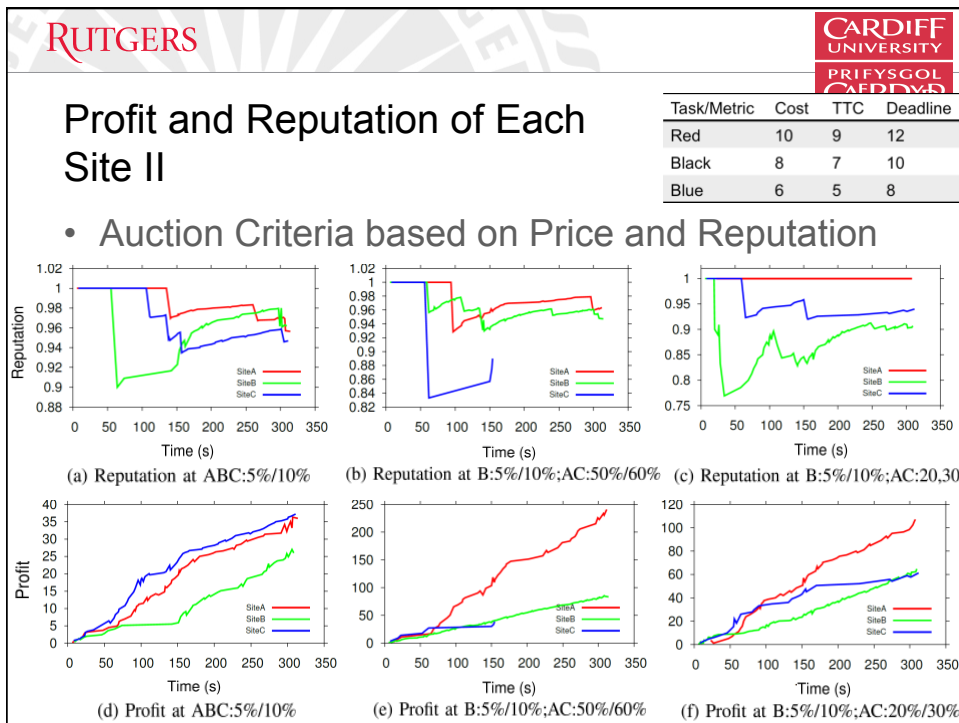
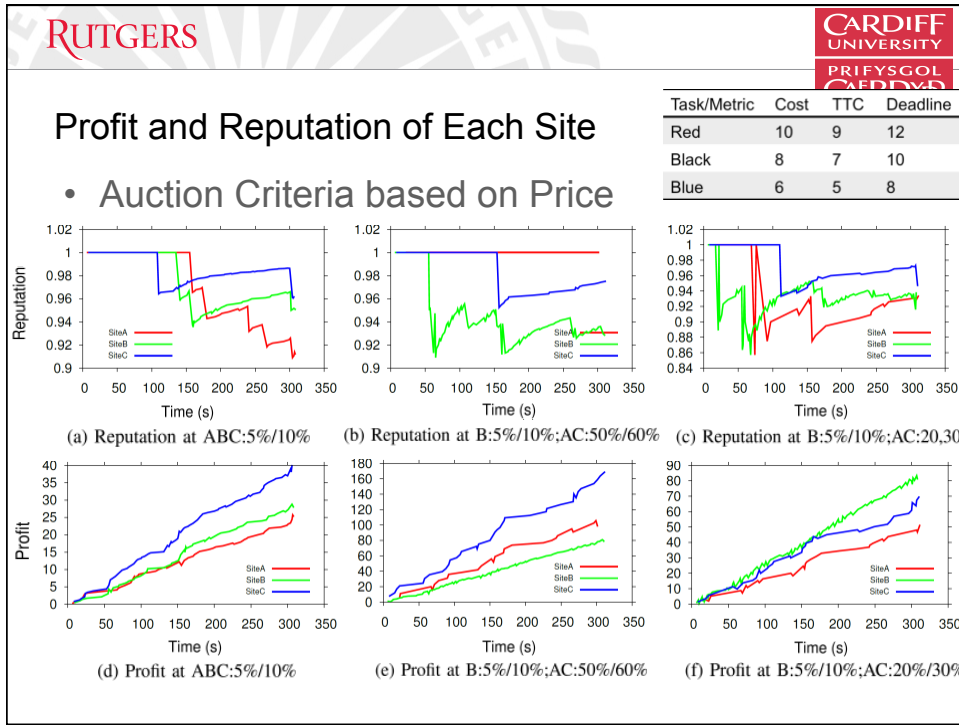




Self-Updating

The availability of accurate performance data therefore is critical:



- The system can self update.
- New devices entering the system or existing devices that may be under trained can update their training dataset.
- Inter worker communication using the CometSpace is used to achieve this.
- Takes advantage of idle runtime.





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.

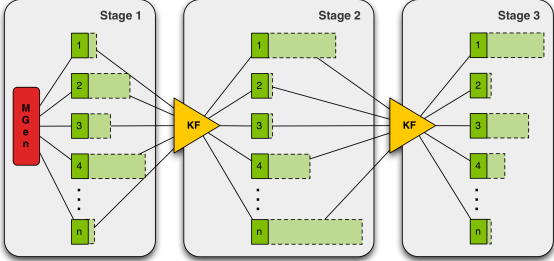
- Resilience – How Clouds can be used to handle:
 - General: Response to dynamic execution environments
 - Specific: Unanticipated HPC Grid downtime, inadequate allocations or unexpected Queue delays/QoS change

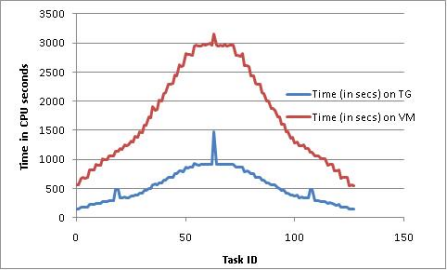






Reservoir Characterization: EnKF-based History Matching

- 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

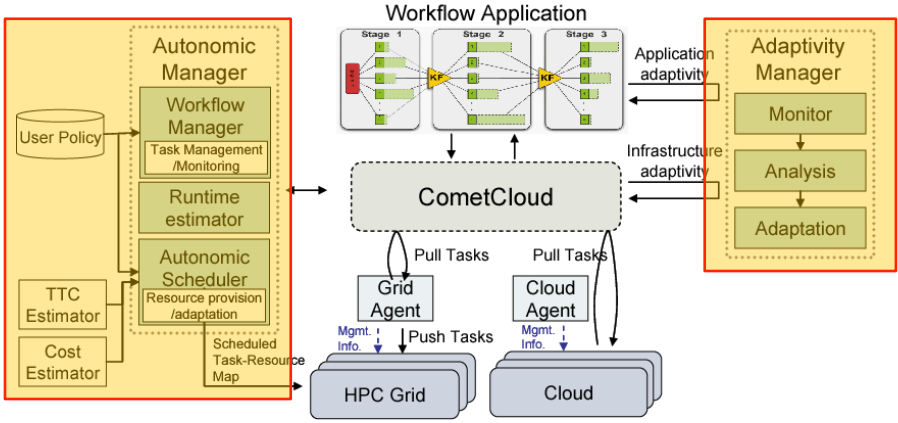








Autonomic HPC+Cloud Federation

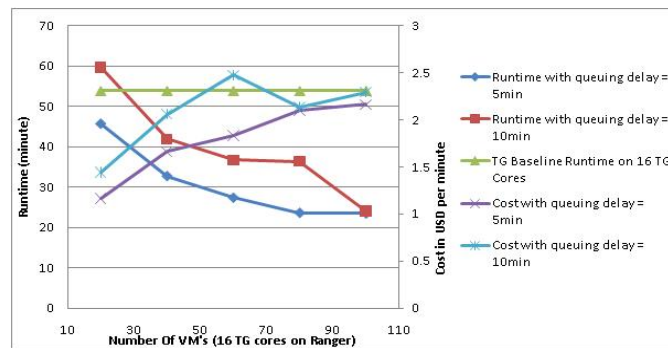



Using Clouds as Accelerators for HPC Grids


- 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

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

Time limit of TeraGrid CPU usage (minute)	total Time-to-Completion (minutes)	total EC2 cost (USD)
5	30	1.19
10	28	1.02
20	25	0.68
30	20	0.17

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

RUTGERS

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Summary & Conclusions

- The future is Cloudy...
 - Cloud becoming a part of production computational environments
 - Many Cloud Computing benefits: Shift CapEx to OpEx , Scale OpEx to demand (up/down/out); Startups and prototyping, One-off tasks (Wash. Post); Cost associativity; ...
- Clouds bring new paradigms and practices, and new complexity
 - New application formulations, new delivery models, new (hybrid) usage modes, new business models, new markets, etc.
- 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....

RUTGERS

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Thank You!

Omer Rana
<o.f.rana@cs.cardiff.ac.uk>

Manish Parashar
<parashar@rutgers.edu>

