Hybrid Cloud Resource Provisioning Policy in the Presence of Resource Failures

Bahman Javadi
University of Western Sydney, Australia

Jemal Abawajy
Deakin University, Australia

Richard O. Sinnott
The University of Melbourne, Australia

The 4th IEEE International Conference on Cloud Computing Technology and Science
Taiwan, December 2012

Agenda

- Introduction
- System Context
- Hybrid Cloud Architecture
- Proposed Provisioning Policies
- Performance Evaluation
- Simulation Results
- Conclusions

INTRODUCTION

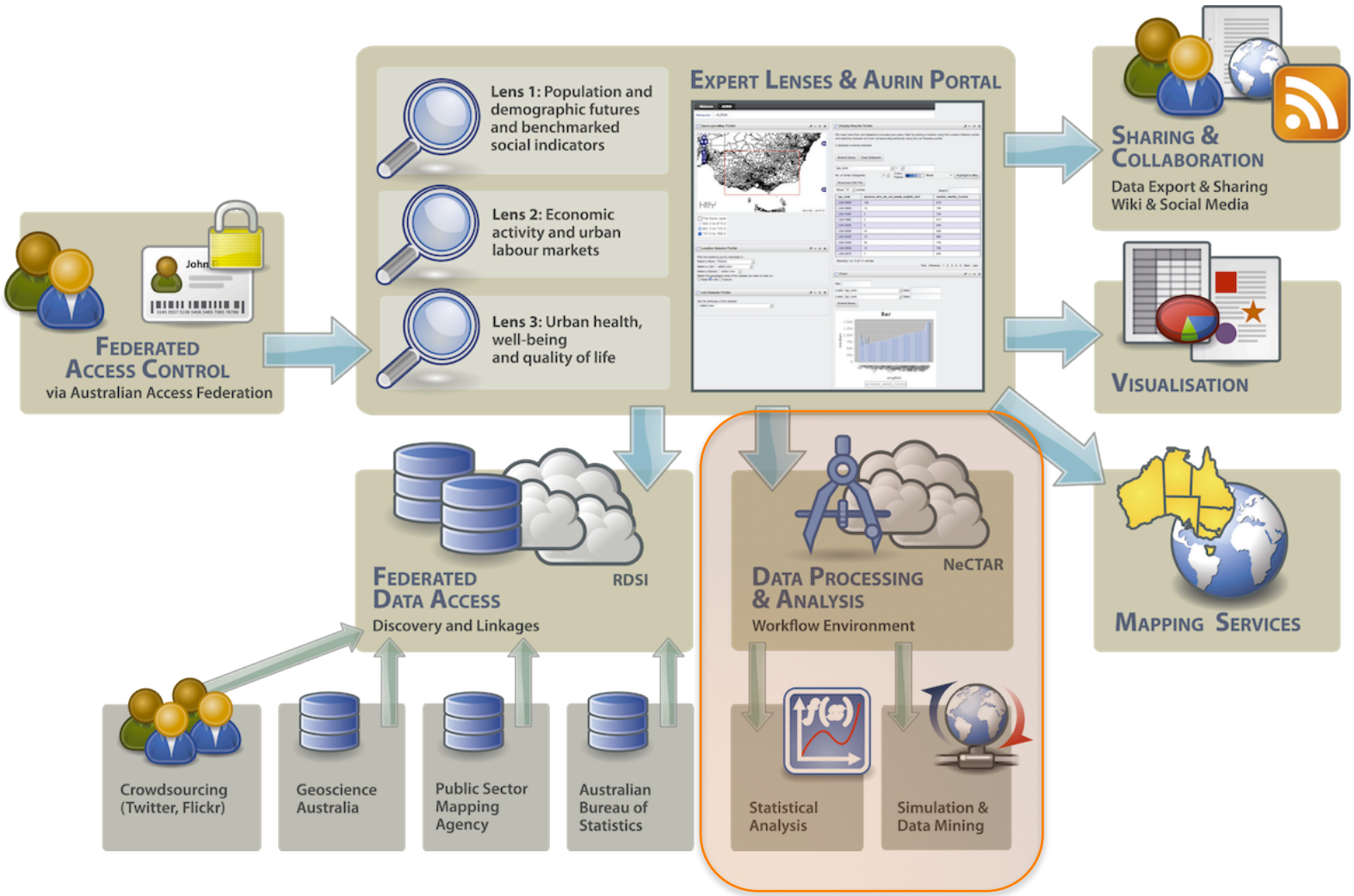
- Hybrid Cloud Systems
  - Public Clouds
  - Private Clouds
- Resource Provisioning in Hybrid Cloud
  - Users' QoS (i.e., deadline)
  - Resource failures
- Taking into account
  - Workload model → workflows in a scientific project
  - Failure correlations → real failure traces
    - *Knowledge-free approach*: not any information about the failure model

System Context

- Our policies are proposed in the context of the Australian Urban Research Infrastructure Network (AURIN) project

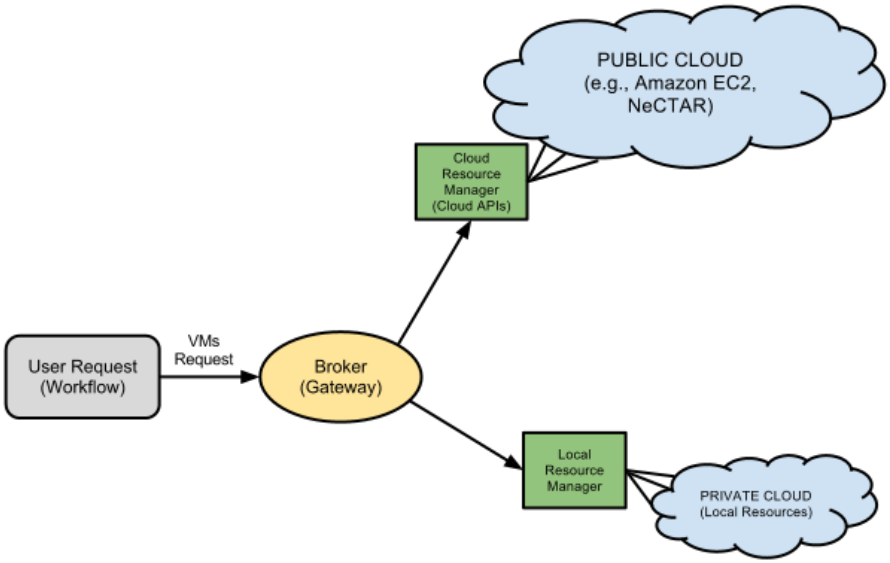
  - An e-Infrastructure supporting research in urban and built environment research disciplines
  - Web Portal Application (portlet-based)
    - A lab in a browser (http://portal.aurin.org.au)
    - Access to the federated data source
      - Web Feature Service (WFS)
    - Workflow environment based on Object Modeling System (OMS)
    - NeCTAR NSP and Research Cloud

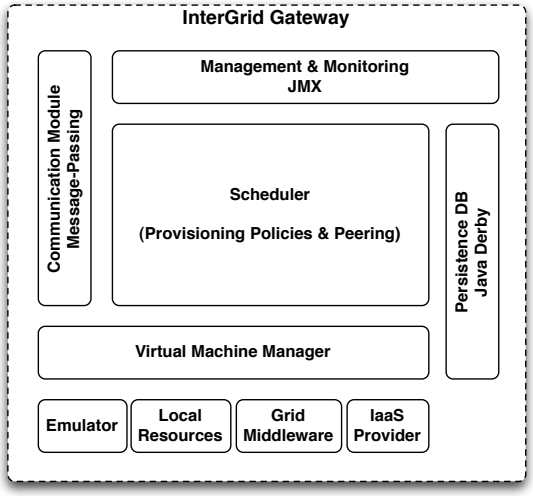
THE AURIN ARCHITECTURE



Hybrid Cloud Architecture

- Based on InterGrid components
- Using a Gateway (IGG) as the broker





IGG

Workload Model

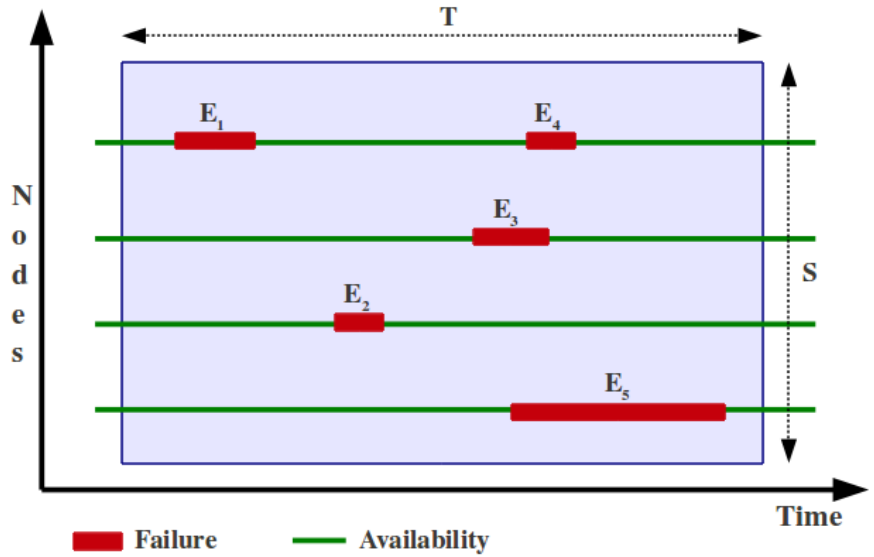
- Workflows in the AURIN project
  - Potentially large number of resources over a short period of time.
  - Several tasks that are sensitive to communication networks and resource failures (*tightly coupled*)
- User Requests
  - Type of virtual machine;
  - Number of virtual machines;
  - Estimated duration of the request;
  - Deadline for the request.

Failures in User Requests

- Resource failure is inevitable
  - Redundant components in public Clouds
    - highly reliable service
  - Leads to service failure in private Clouds

- Correlation in Failures → *overlapped failures*
  - Spatial
  - Temporal

Failures in User Requests (cont.)



- The sequence of overlapped failures

$$H = \{F_i \mid F_i = (E_1, ..., E_n), T_s(E_{i+1}) \leq T_e(E_i)\}$$

- Downtime of the service

$$D = \sum_{\forall F_i \in H} (max\{T_e(F_i)\} - min\{T_s(F_i)\})$$

Proposed Policies

- Size-based Strategy
  - Spatial correlation : multiple failures occur on different nodes within a short time interval
  - Strategy: sends wider requests to more reliable public Cloud systems
  - Mean number of VMs per request
    - P_1: probability of one VM
    - P_2: probability of power of two VMs

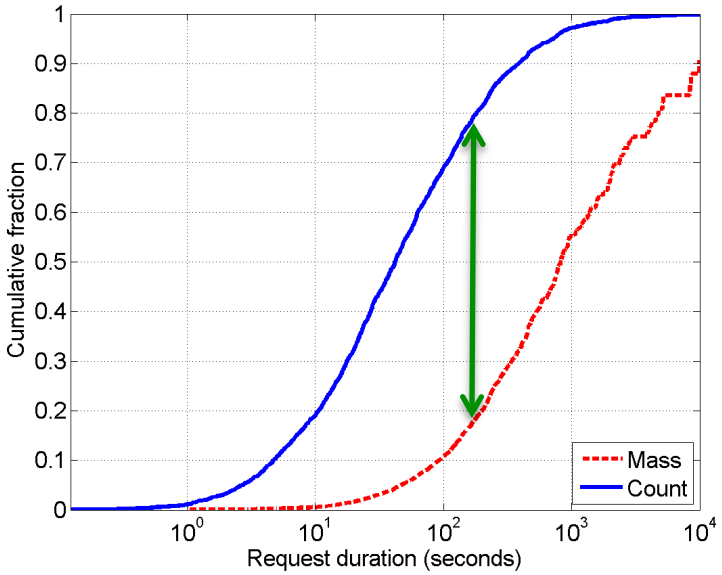
$$\overline{S} = P_1 + 2^{\lceil k \rceil}(P_2) + 2^k\,(1 - (P_1 + P_2))$$

    - Request size: two-stage uniform distribution (l,m,h,q)

$$k = \frac{ql + m + (1-q)h}{2}$$

PROPOSED POLICIES (CONT.)

- Time-based strategy
  - Temporal correlation: the failure rate is time-dependent and some periodic failure patterns can be observed in different time-scales
  - Request duration: are *long tailed*.



- The mean request duration
  - Lognormal distribution in a parallel production system

$$\overline{T} = e^{\mu + \frac{\sigma^2}{2}}$$

PROPOSED POLICIES (CONT.)

- Area-based strategy
  - Making a compromise between the size-based and time-based strategy
  - The mean area of the requests

$$\overline{A} = \overline{T} \cdot \overline{S}$$

  - This strategy sends long *and* wide requests to the public Cloud,
  - It would be more conservative than a *size-based* strategy and less conservative than a *time-based* strategy.

Scheduling Algorithms

- Scheduling the request across private and public Cloud resources
- Two well-know algorithms where requests are allowed to leap forward in the queue
  - Conservative backfilling
  - Selective backfilling

$$XFactor = \frac{W_i + T_i}{T_i}$$

- VM Checkpointing
  - VM stops working for the unavailability period
  - The request is started from where it left off when the node becomes available again

Performance Evaluation

- CloudSim Simulator
- Performance Metrics
  - Deadline violation rate
  - Slowdown

$$Slowdown = \frac{1}{M}\sum_{i=1}^{M}\frac{W_i + max(T_i, bound)}{max(T_i, bound)}$$

  - Cloud Cost on EC2

$$Cost_{pl} = (H_{pl} + M_{pl} \cdot H_u)\, C_n + (M_{pl} \cdot B_{in})\, C_x$$

  - Workload Model
    - Parallel jobs model of a multi-cluster system (i.e., DAS-2)

Input Parameters	Distribution/Value
Inter-arrival time	Weibull ($\alpha = 23.375, 0.2 \leq \beta \leq 0.3$)
No. of VMs	Loguniform ($l = 0.8, m, h = log_2 N_s, q = 0.9$)
Request duration	Lognormal ($2.5 \leq \mu \leq 3.5, \sigma = 1.7$)
P_1	0.02
P_2	0.78

Performance Evaluation (cont.)

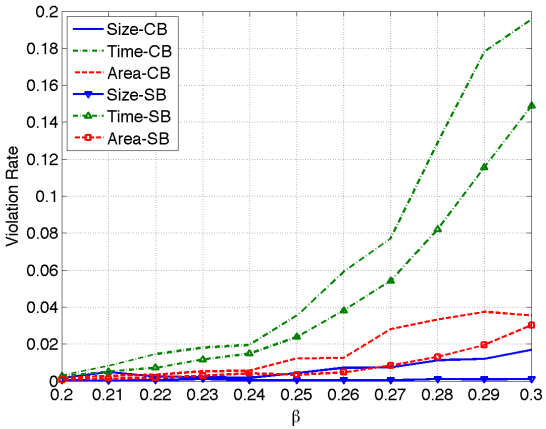
- Failures from Failure Trace Archive (FTA)
  - Grid'5000 traces
    - 18-month
    - 800 events/node
    - Average availability: 22.26 hours
    - Average unavailability: 10.22 hours
- Synthetic Deadline

$$d_i = \begin{cases} st_i + (f \cdot\ ta_i), & \text{if } [st_i + (f \cdot\ ta_i)] < ct_i \\ ct_i, & \text{otherwise} \end{cases}$$

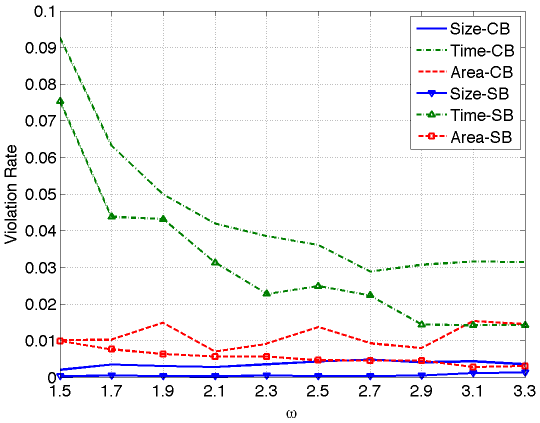
  - f: stringency factor
  - f>1 is normal deadline (e.g., f=1.3)
- $N_s = N_c = 64$

Simulation Results

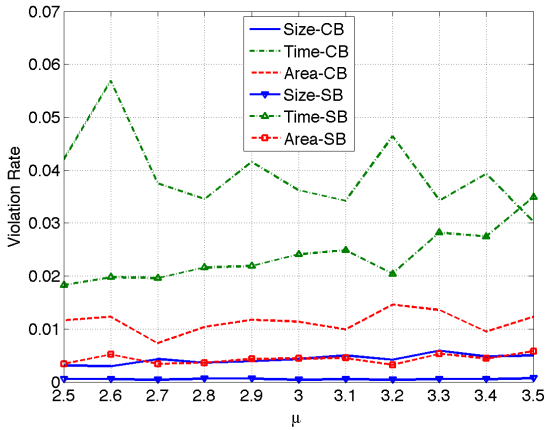
- Violation rate



Request arrival rate



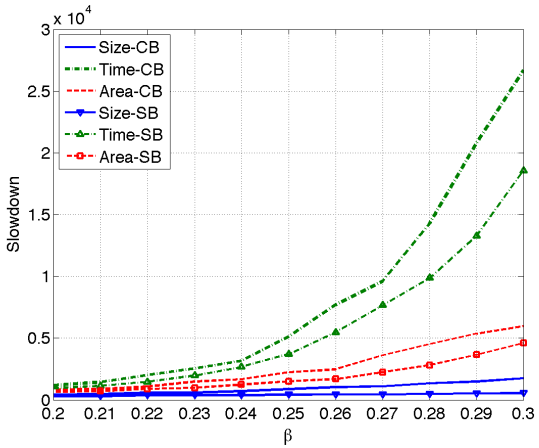
Request size



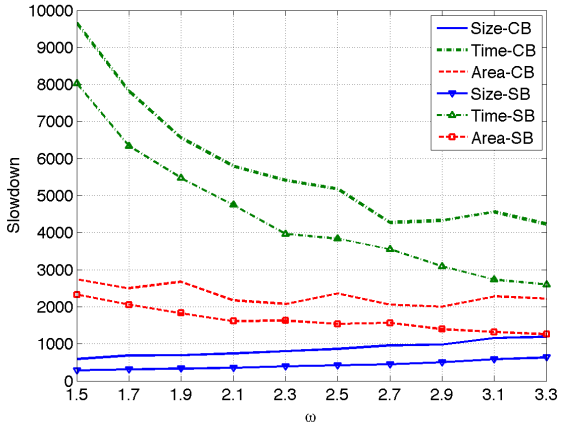
Request duration

Simulation Results (cont.)

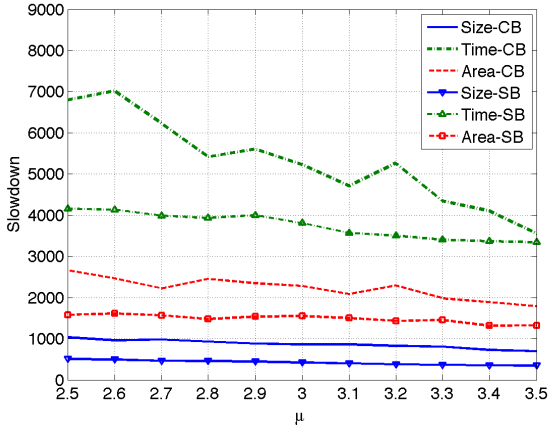
- Slowdown



Request arrival rate



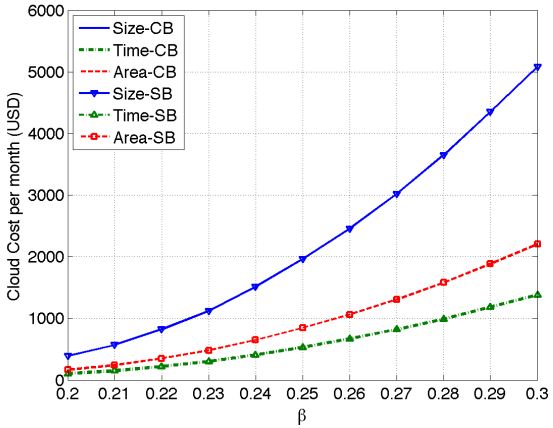
Request size



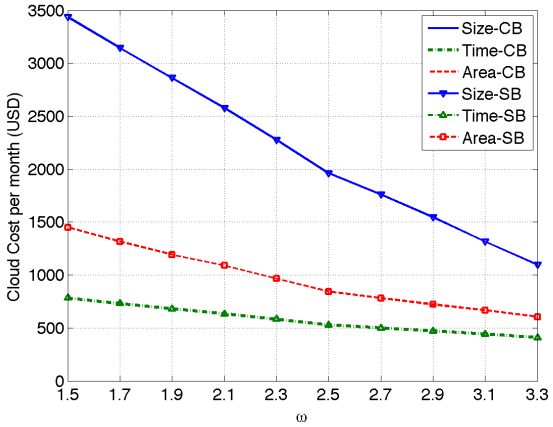
Request duration

Simulation Results (cont.)

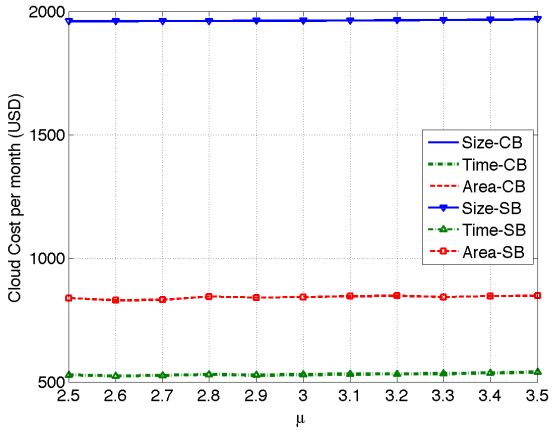
- Cloud Cost on EC2



Request arrival rate



Request size



Request duration

Conclusions

- QoS-based resource provisioning in a failure-prone hybrid Cloud system
- Three different *flexible* brokering strategies based on failure correlation and workload model
- Knowledge free approach
- Using time-based strategy (high load),
  - 20% violation rate
  - ~1200 USD per month on EC2
- Future Work
  - Use a set of real workflow applications from the AURIN project and run real experiments.

Thank You