Resource Management on Clouds: the Multifaceted Problem and Solutions

Shikharesh Majumdar

Real Time and Distributed Systems Research Centre

Dept. of Systems & Computer Engineering,

Carleton University,

Ottawa, CANADA.

email: majumdar@sce.carleton.ca

Acknowledgments

Umar Farooq (Carleton University)
Orlando Melenedez (Carleton University)
Phuong Hoang (Carleton University)
Rvishankar Kanagasundaram
Eric Parsons (Nortel/Ericsson)
Biswajit Nandy (Solana Networks)
Nabil Seddigh (Solana Networks)

- Financial Support:
  - Natural Sciences and Engineering Research Council of Canada (NSERC)
  - Ontario Centers of Excellence
  - Cistel Technologies
  - Nortel

Cloud Computing

- Internet Based Computing
  - Multi-billion dollar market
  - Public, Private (Enterprise)

- Provides one or more resources (e.g. computing & storage) <u>on demand</u>
  - From a remote pool of resources

- Pay as you go
  - Resources are "rented" when needed

- Elastic

- Virtualization

Software as a Service (SaaS)

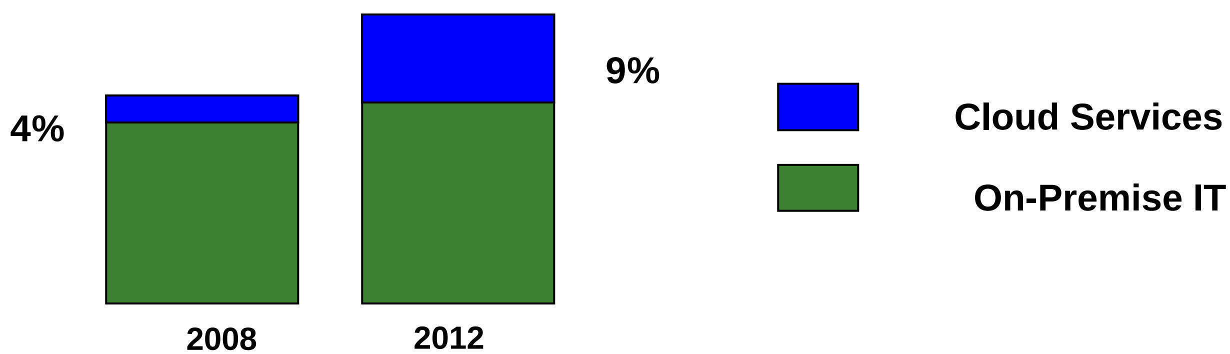
Platform as a Service (PaaS)

Infrastructure as a Service (IaaS)

From *Cloud Computing – A Place to Learn Cloud Computing, 2012* http://cloudcomputingx.org [Available via http//images.google.ca]

Market Demand

- Virtualization market is growing at the rate of 36% for last few years
- Cloud Computing market
  - Various forecasts from different agencies
  - $148.8 B by 2014 – Gartner
- Price Waterhouse Coopers summer Technology Forecast says that *cloud will be necessary for automating the world of IT*:
  - "...IT must adopt an architecture that creates loose coupling between the IT infrastructure and application workloads. It also must modernize and automate IT's own internal business processes for provisioning, managing, and orchestrating infrastructure resources.“
- <u>World-Wide Enterprise IT Spending (based on IDC October 2008)</u>



Advantages

- Start Up companies
  - Low IT investment
  - Pay as you go
- Enterprise Data Centres
  - No cost of upgrading
  - Reduced IT management
  - Getting resources on demand (Elasticity)
- Research & Engineering Data Centres
- Green Computing
  - Potential reduction in energy consumption

Challenges

- Lack of Control
- Effective Resource Management Techniques
- Inter-operability among multiple clouds
- Appropriate revenue model

What is Important?

- **Survey conducted for determining the relative importance of various challenges/issues for the cloud/on demand model**
- **Results**

Security

Performance

Availability

Hard to combine with in-house efforts

Inadequate ability to customize

Higher cost

Others (three)

IMPORTANCE

[Source: F. Gens, "Enterprise IT in the Cloud Computing Era - New IT Models for Business Growth & Innovation", IDC Corporation, 2008. [Available from : http://www.inst-informatica.pt/servicos/informacao-e-documentacao/dossiers-tematicos/teste-dossier-tematico-no-7-cloud-computing/tendencias/idc-enterprise-it-in-the-cloud-computing-era-new]

Top Concerns: Security, Performance

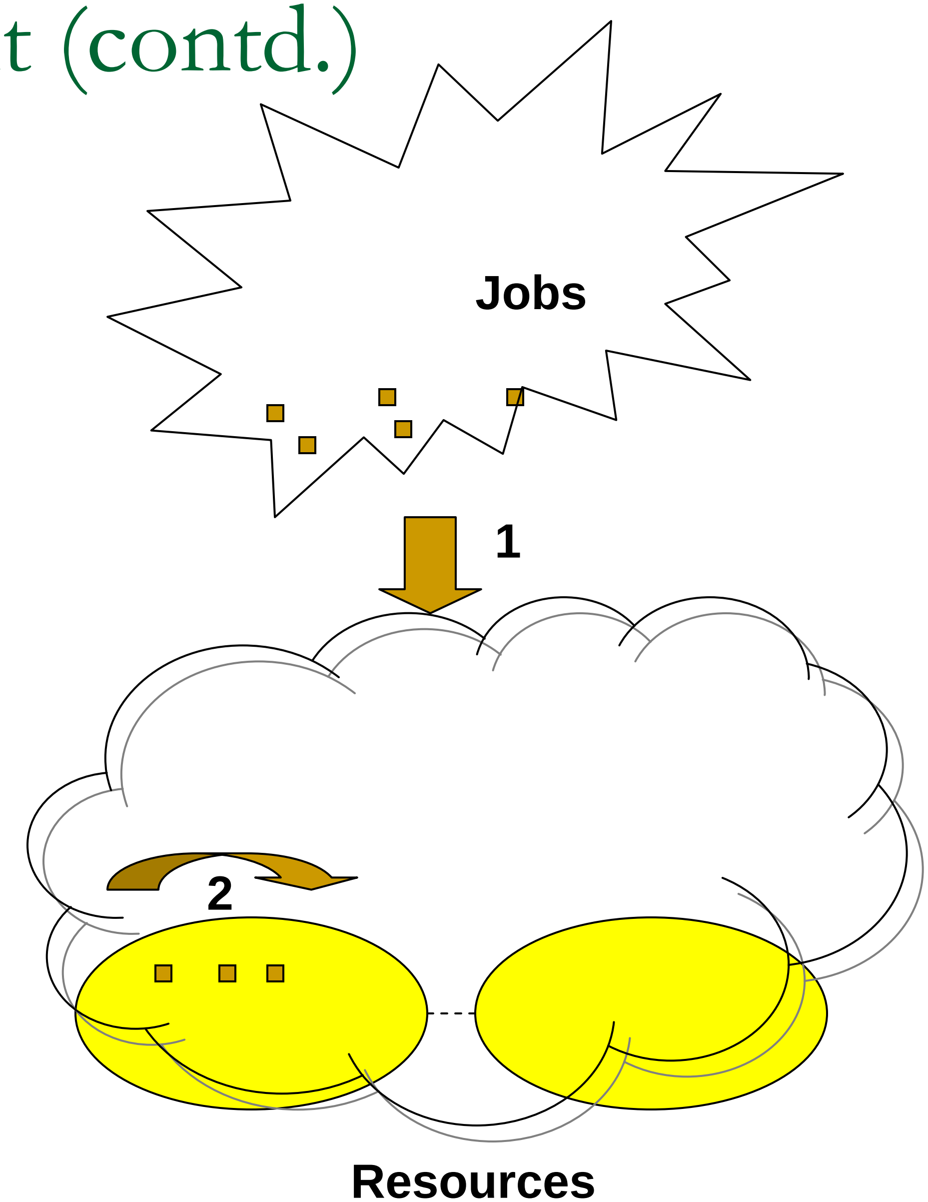
Resource Management

- Many different facets
  - Allocation (matchmaking) & Scheduling
    - Resource heterogeneity and dynamicity
    - Handling uncertainty in user estimates of application execution times
    - Coping with lack of knowledge of local resource management policies
    - Generating adequate revenue for service provider
    - Conserving energy
  - Providing Inter-operability (resource access)
  - System Monitoring
  - System Security

Resource Management (contd.)

Key Operations:

- **1. Allocation (Matchmaking)**
  - Choose resources on which jobs (requests) will be executed. E.g. [5]
- **2. Scheduling**
  - Order requests mapped on to a given resource.
- **Workload:**
  - On Demand Requests
    - Best effort
  - Service Level Agreements (SLAs)
    - Earliest start time, deadline and execution time
    - Also referred to as Advance Reservation requests



Scheduling on Clouds and Grids

- *Challenges : Devise effective* resource management strategies
  - *Multiple resources*
  - *Different resource types*
  - *Handling SLA*

Example Scheduling Strategy: SSS [1,2]

- *Goal: Devise effective scheduling strategy for a single resource*
- Input Traffic*:*
  - *Advance Reservation (AR) Requests:*
    - Introduced as part of Globus Architecture for Reservation and Allocation (GARA).
    - Characterized by a *Start Time* and an *End Time (SLA)*
    - Guaranteed service – QoS assurances
  - *On Demand (OD) Requests*:
    - Best effort

[1] Farooq, U., Majumdar, S., Parsons, E.W., "Impact of Laxity on Scheduling with Advance Reservations in Grids," in the *Proceedings of the 13th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'05),* pp. 319- 324, Atlanta, GA, September 2005.

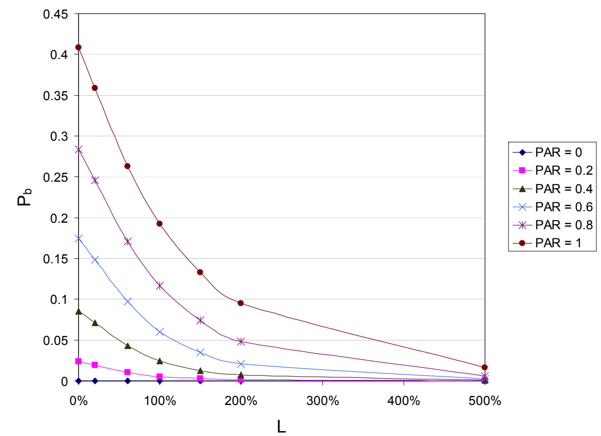
[2] Farooq, U., Majumdar, S., Parsons, E.W., "Dynamic Scheduling of Lightpaths in Lambda Grids," in the *Proceedings of the 2nd IEEE International Workshop on Networks for Grid Applications (GRIDNETS'05)*, pp. 540-549, Boston, MA, October 2005.

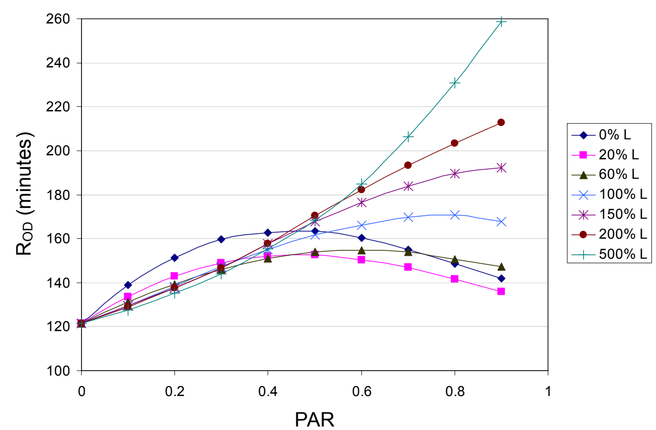
SSS Algorithm – a High Level Description

- Basic Idea: <u>S</u>caling through <u>S</u>ubset <u>S</u>cheduling
  - Whenever a new request arrives, the SSS algorithm first finds all those tasks in the resource schedule that can affect the feasibility of the new schedule with the new request and then tries to work out a feasible schedule for only that subset of tasks S.
- **Step 1 :** Obtain S – Set of all those tasks that can affect the scheduled-time of the new task and whose scheduled-time can be affected by the new task.
- **Step 2 :** Obtain an initial solution for tasks in S using the modified Earliest-Deadline-First Strategy that accounts for both preemptable and non-preemptable tasks.
- **Step 3 :** If the solution is feasible, accept the task and update resource schedule. Otherwise, calculate lower bounds on the lateness of the *critical task* and see if its lateness can be improved. If it cannot be improved reject the new task. Otherwise, go to step 4.
- **Step 4 :** Improve on the initial solution iteratively using *pruned* branch and bound technique.

Summary of Observations

- SSS can effectively handle ARs + ODs on a Grid.
- Can be adapted to Clouds
- Laxity in the reservation window can significantly improve system performance by reducing probability of blocking and increasing utilization.
  - The effect is more pronounced for the cases where proportion of advance reservations is high.
- Data segmentation can also improve system performance:
  - Depends on the distribution of service times.
  - More improvement in U and R_{OD} with high variance in service times.
- The results also show that the improvement in performance with segmentation is sensitive to L. At higher L values, difference in utilization diminishes. This suggests that laxity can be exchanged for data segmentation to achieve high utilization of lightpaths.





Handling Error Associated with User Estimated Runtimes

- User estimates for run times of jobs are often incorrect
  - Users often overestimate job execution times
    - Observed to be very large (even up to 25000%) in [Lublin et al. 2003]
    - Abnormal termination of jobs
    - Both of these contribute to unnecessary rejections of jobs leading to a poor useful utilization of the resource
  - Users can underestimate job run times as well
    - Leads to job abortions leading to a high job rejection rate
    - Can decrease useful utilization of resource

Solution for Handling Error

- **Schedule Exceptions Manager (SEM)** [3]
  - monitors the resource schedule
  - deals with exceptions resulting from abnormal terminations and inaccuracies in user-estimated runtimes.
- When a job leaves earlier than expected (over estimation)
  - SEM performs rescheduling of existing jobs
- When a job exceeds its specified run time (under estimation)
  - SEM consults *Abortion Policy Block*
- Two Abortion Policies: *Feasibility Policy (FP)* and *Penalize Underestimation Policy* (PU).

- **Feasibility Policy**
  - Consider providing additional time quanta to job:
    - Size of each quanta proportional to estimated job size τ = σ * eE,ib.
  - Abort job if providing additional quanta leads to deadline violations for already accepted jobs
- **Penalize Underestimation Policy**
  - Abort job that has exceeded its specified runtime

[3]. Farooq, U., Majumdar, S., Parsons, E., "Achieving Efficiency, Quality of Service and Robustness in Multi-Organizational Grids", *Journal of Systems and Software* (*Special Issue on Software Performance*), Vol. 82, Issue: 1, January 2009, pp. 23-3.

Solution for Handling Error (contd.)

- **<u>Pre-Scheduling Engine</u>**

  - Aims at combating over estimation of job runtimes

  - Underconstraints requests

  - **<u>Overbooking PE Mechanism</u>**

    Step 1. Artificially reduce user estimated runtimes

    Step 2. Perform schedulabilty analysis

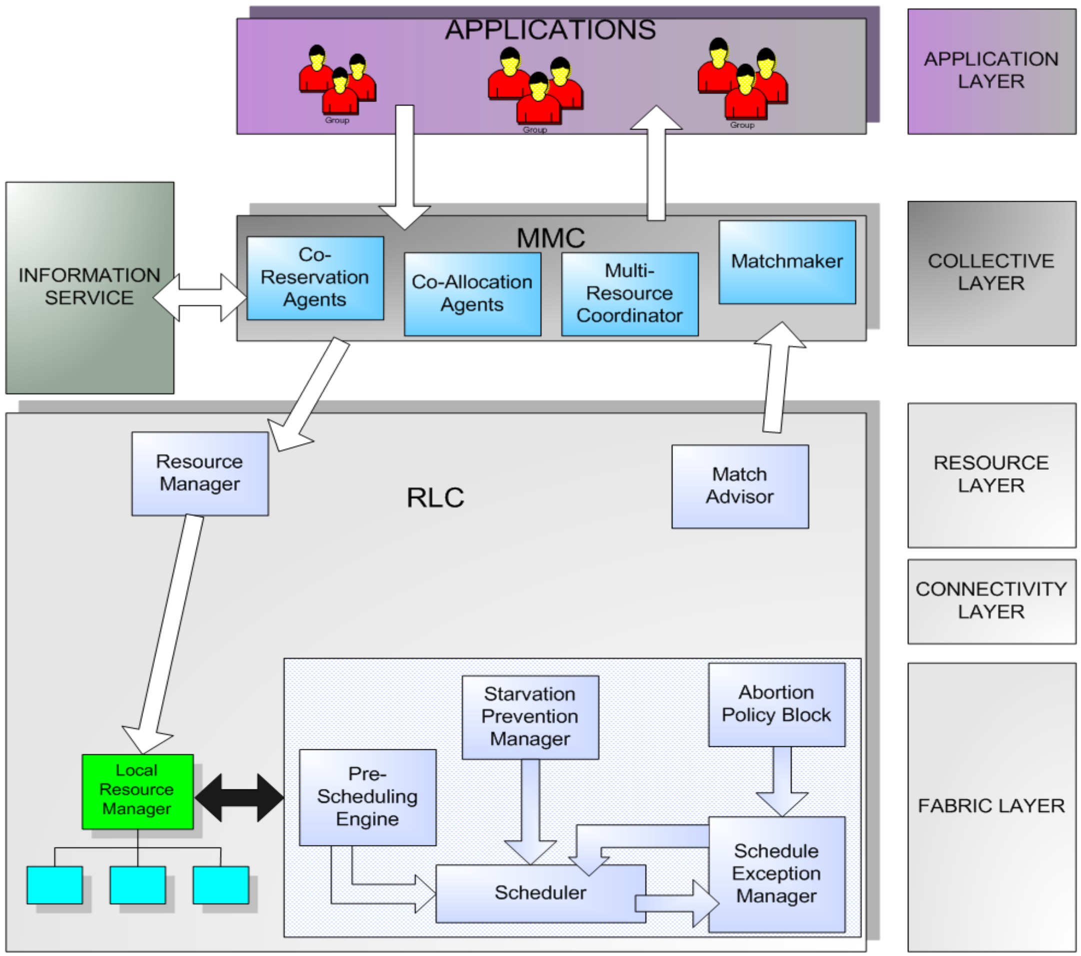
    Step 3. If proportion of jobs rejected (after accepting job) < threshold

    accept job

    else reject job

Resource Management Framework

QoS aware resource Management in multi-Organizational grid Systems (QoSMOS) *(Carleton-Nortel/NSERC) –* Extendable to Clouds



Workload:

•On Demand Requests
  •Best effort
•Advance Reservation Requests
  •Earliest start time, deadline and execution time

Challenges
✓Handling QoS (SLA)
✓How to predict request execution times?

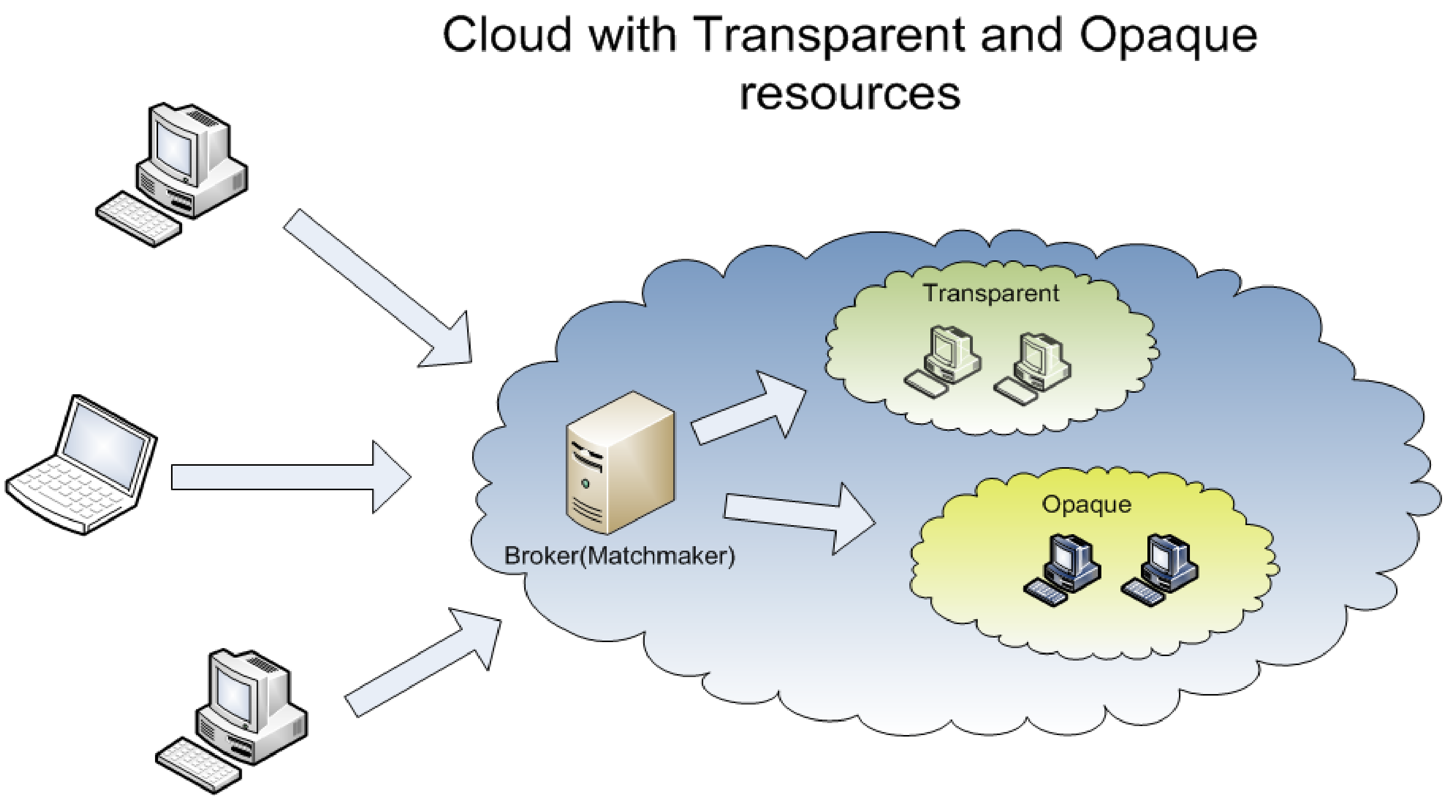
? Difficulty in acquiring a priori knowledge of local resource management policies in a large heterogeneous and dynamic environment

MMC: Matchmaker & Multi-Resource Coordinator
RLC: Resource Liaison & Controller

Matchmaking in Clouds

- Goal: Devise Effective matchmaking strategies for achieving high quality of service
- **Focus:** Handling Uncertainties: lack of knowledge of scheduling policies used at resources
- Heterogeneity in resources
  - Many different types of resources each with its own operating system are possible
- Resources are dynamic
- May not always be possible to know local scheduling policy used at a resource
- Even if local scheduling policy is know, it may be time consuming to simulate the policy
- How to perform matchmaking without *detailed a priori* knowledge of scheduling policy used at a resource

Matchmaking "in the dark"



- Transparent Resource: Known resource scheduling policy
- Opaque Resource: Unknown resource scheduling policy

Any Schedulability-Based Matchmaker [5]

- Any Schedulability (AS) Criterion [4] :
  - Given a set of ARs, AS criterion includes a set of inequalities involving AR characteristics
  - Satisfying the inequalities guarantees that each AR in the set will meet its deadline as long as a work conserving scheduling policy is used at the resources.
  - No further knowledge of scheduling policy deployed at the resource is required

- AS-Based matchmaking:
  - Broker selects a resource that satisfies the AS criterion

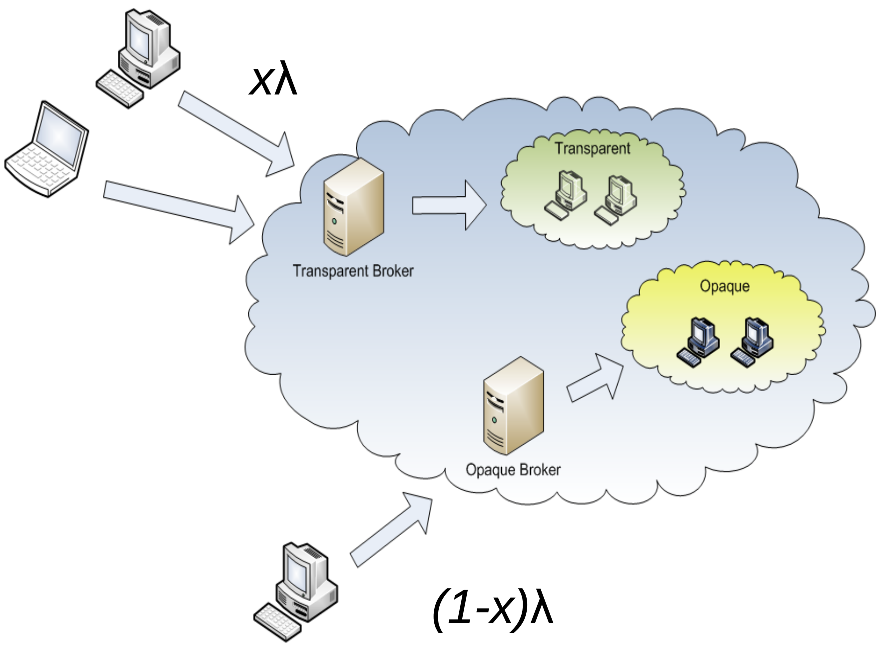
  upon arrival of an AR
    - Single resource : accept request iff any-schedulability criterion is satisfied

    - <u>Multiple Resources: Allocate request to that resource that satisfies the any-schedulability criterion</u>

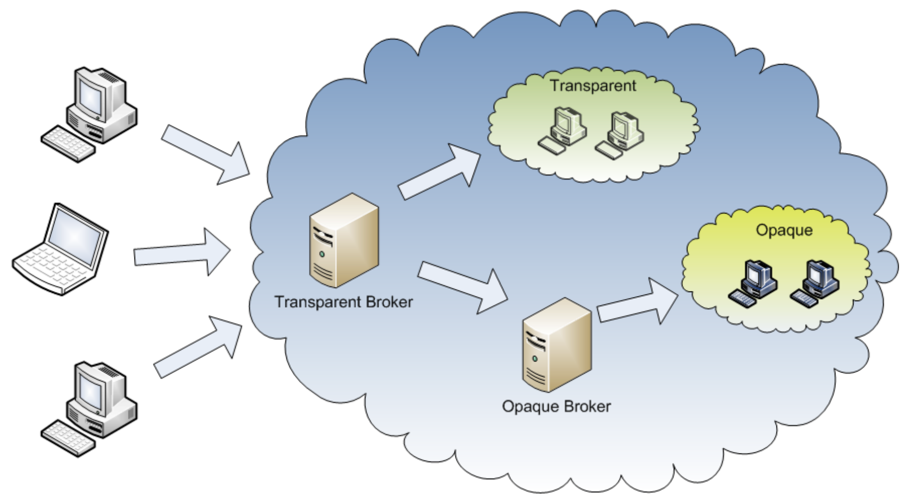
[4] Majumdar, S. "The "Any-Schedulability" Criterion for Providing QoS Guarantees Through Advance Reservation Requests", *Proc. Cluster Computing and the Grid* (*International Workshop on Cloud Computing)*, Shanghai (China), May 2009, pp. 490-495
[5] Melendez, J.O., Majumdar, S., "Matchmaking with limited Knowledge of Resources on Clouds and Grids", *Proc. 2010 International Symposium on Performance Evaluation of Computer and Telecommunication Systems SPECTS' 10),* Ottawa, July 2010.

Hybrid Matchmaking

Independent Strategy

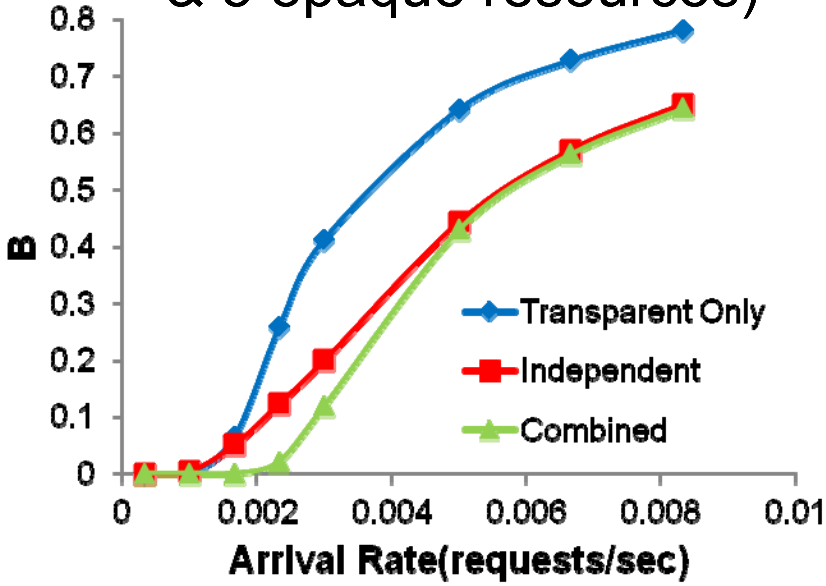


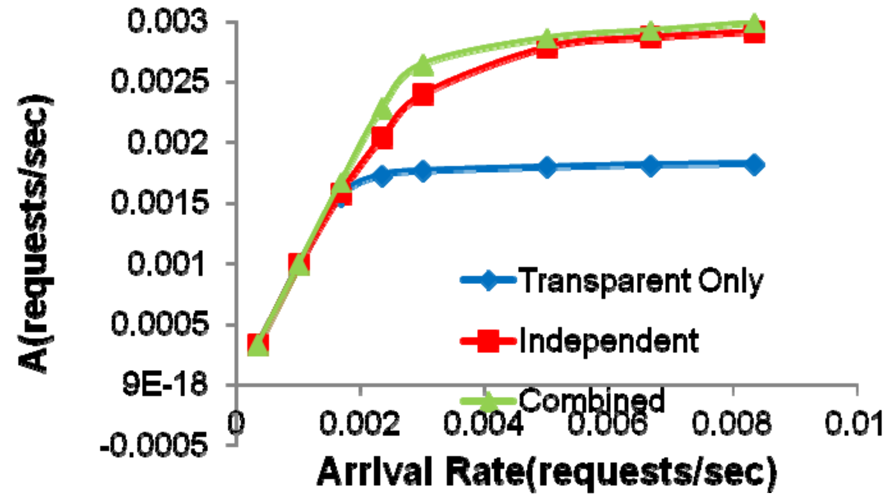
Combined Strategy



Sample Simulation Results Cont'd

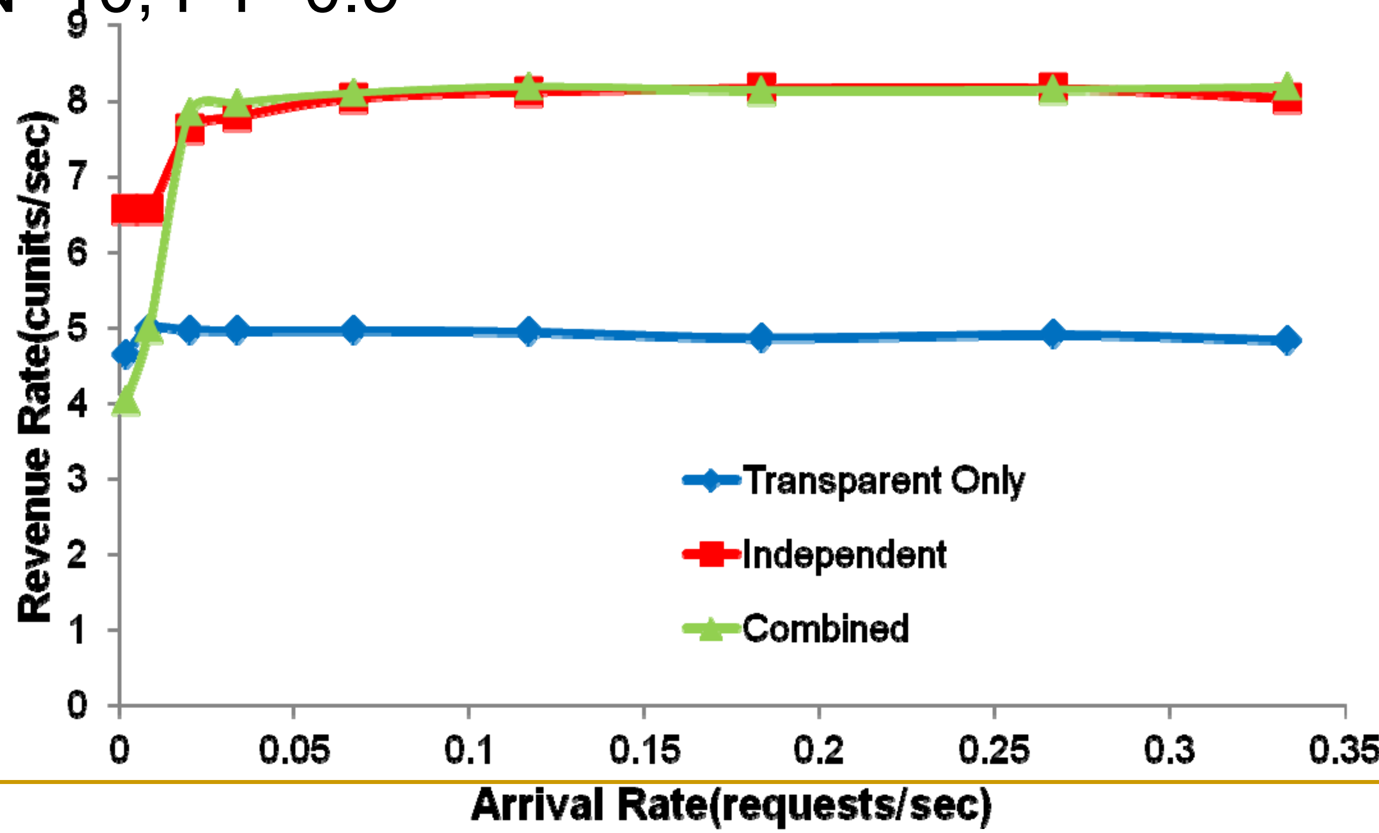
- System Strategy Comparison
  - Hybrid System parameters: N=10, PT=0.5 (5 Transparent & 5 opaque resources)





Sample Simulation Results Cont'd

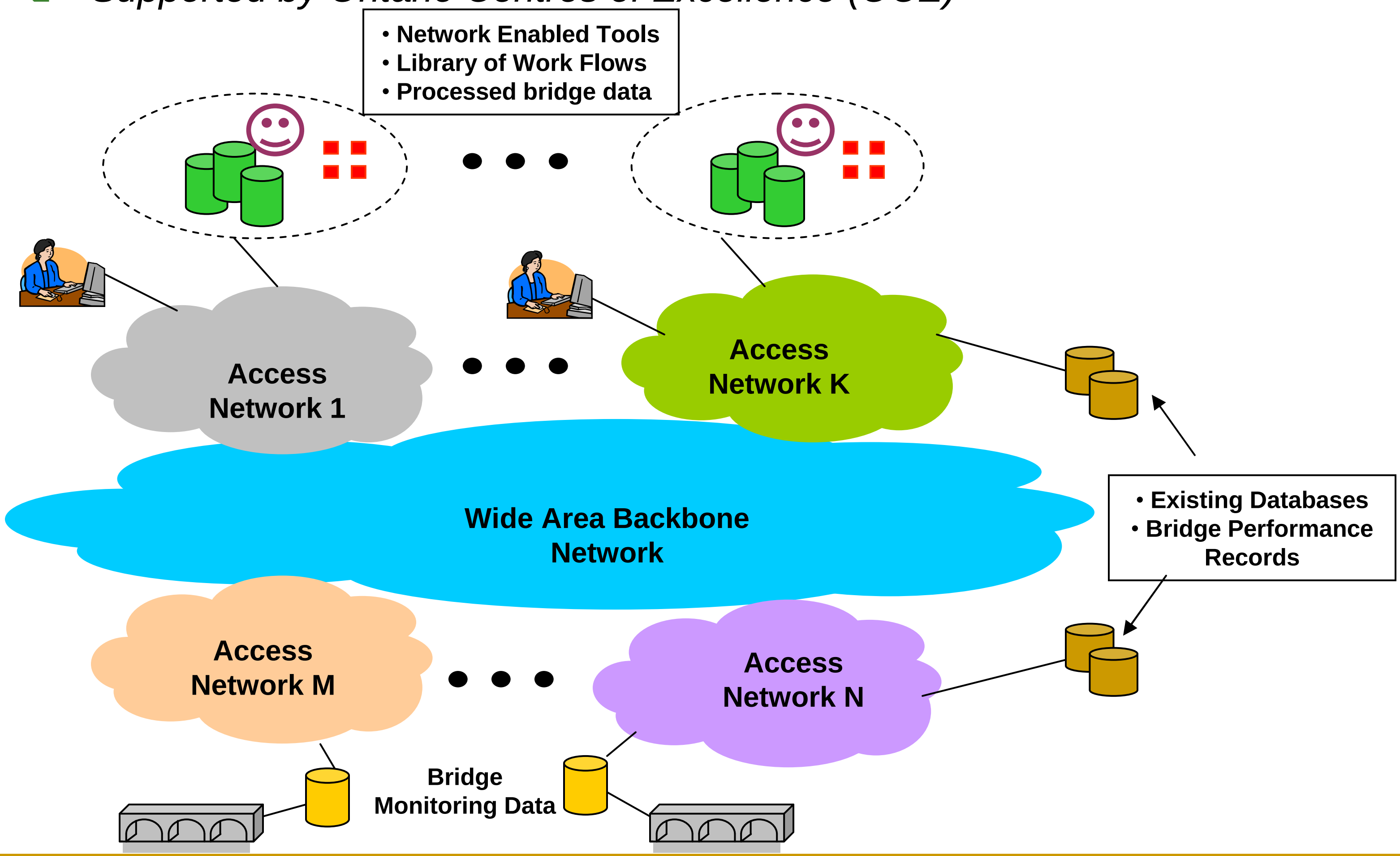
- System Strategy Comparison
  - N=10, PT=0.5



Resource Heterogeneity

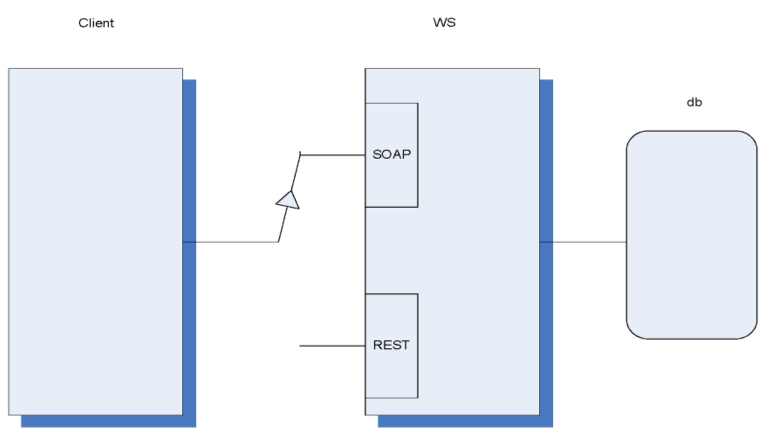
R&E Cloud for *Bridge Infrastructure Management (Carleton-Cistel)*

- *Supported by Ontario Centres of Excellence (OCE)*



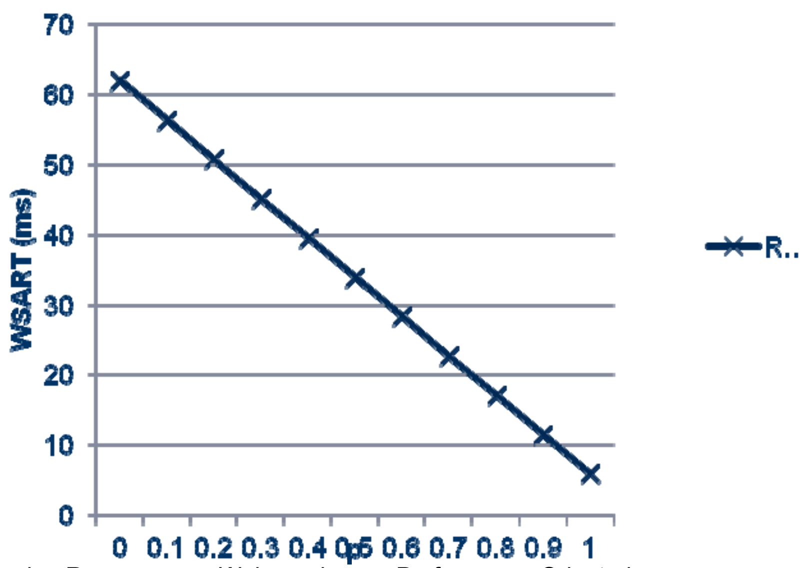
Interoperability

- Expose Resource as a Web service (WS)
  - SOAP-based (has support for additional standards)
  - RESTful (lightweight)
- Hybrid WS
  - Example: Database (DB) exposed as a Hybrid WS



Sample Performance Results

- Experiments performed with a prototype performing database operations.
- ***p*** - probability of using RESTful Web service
  - p = 0 – fully SOAP-based WS
  - p = 1 fully always RESTful WS



[6] Kanagasundaram, R., Majumdar, S., Zaman, M., Srivastava, P., Goel, N., “Exposing Resources as Web services: a Performance Oriented Approach”, *Proc. 2010 International Symposium on Performance Evaluation of Computer and Telecommunication Systems SPECTS‘ 10),* Ottawa, July 2010

Conclusions

- Resource management in grids and clouds has a strong impact on performance
  - Scheduling
  - Matchmaking
- Uncertainties regarding user specified job execution times and the local resource management policy used at individual resources add to the challenges for resource management techniques.
- Prescheduling Engine and Schedule Exceptions Manager effectively deal with error associated with user estimated job execution times.
- Matchmaking *in the dark*: Any Schedulability criterion enables augmentation of the resource pool by effectively utilizing opaque resources
- The benefit of incorporating opaque resources in the resource pool translates directly to an improvement in A and R
- Handling inter-opearability: Hybrid WS

Challenges:

- Multiple dimensional optimization problem: SLA, revenue, energy
- Handling mobility
- Handing real time transactions
- Security