Usage Patterns to Provision for Scientific Experimentation in Clouds

Eran Chinthaka Withana and Beth Plale
School of Informatics and Computing, Indiana University
Bloomington, Indiana, USA.

2nd International Conference on Cloud Computing Technology and Science, Indianapolis, IN, US

Summary

- Doing Science in Cloud
- Improving Scientific Job Executions in Cloud Resources
- Role of Successful Predictions to Reduce Startup Overheads
- System Architecture
  - Use of Reasoning
- Evaluation
- Discussion and Future Work

Clouds as a Complementary Solution to Grids for Science

- Issues with existing systems
  - Batch oriented HPC resources with long queue wait times, even under moderate loads
  - No access transparency
  - Quota system requires maximum resources to be known and approved in advance
- Advantages of using cloud resources
  - Availability of “unlimited” compute resources the instant they are needed
  - Pay-as-you-go model
    - eliminates up-front commitments
    - Encourages scientists to budget for the resources they are willing to pay
- Issues with Clouds
  - Slow interconnects
  - virtualization overhead and startup times
  - Consumption based billing
- Emergence of new programming paradigms to exploit the advantages of Cloud resources

Challenges with Cloud Computing Resources

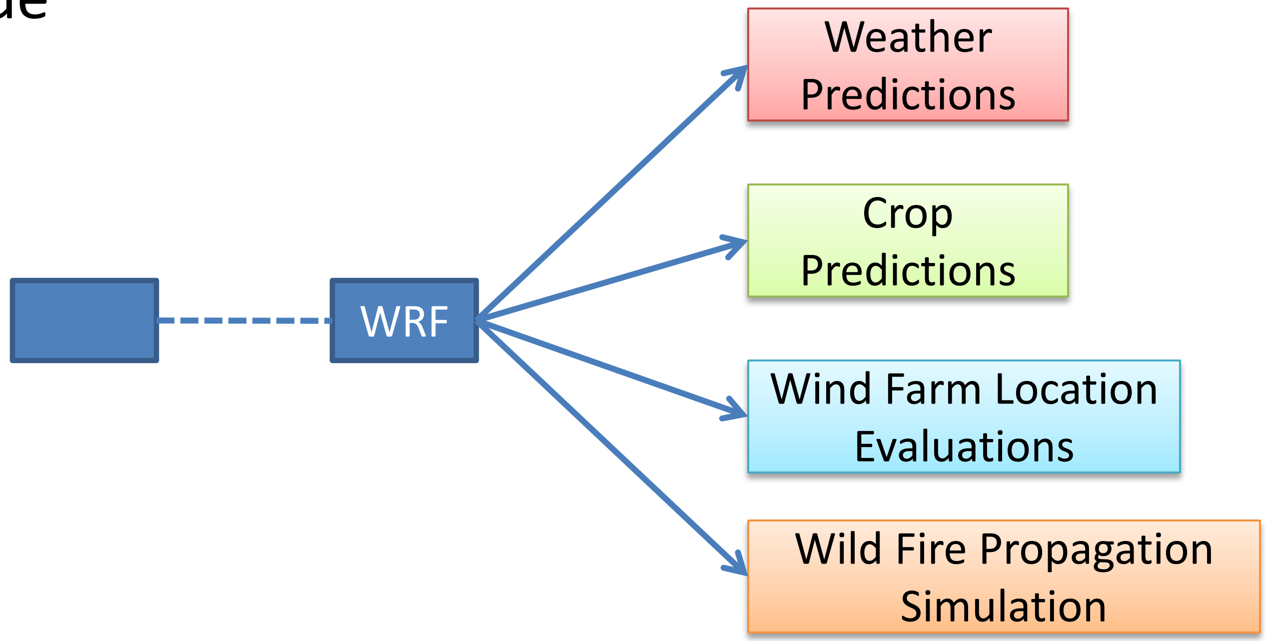
- Scheduling algorithms
  - Focused on optimal utilization of relatively homogeneous grid or cluster resources
  - Resources can be provisioned supporting user requirements in clouds
- Prediction Algorithms
  - Different hardware configurations forces execution time predictions to factor non-uniformity of resources

Improving Scientific Job Executions in Cloud Resources

- Solution Space
  - Meta-scheduler that uses historical information to anticipate future activity (AppleS, GRADS)
  - Resource abstraction service (Nimrod/G)
- Reducing the impact of startup overheads, learning from user behavioral patterns, by predicting future jobs
- Talk outline
  - Algorithm to predict future jobs by extracting user patterns from historical information
    - Reduces the impact of high startup overheads for time-critical applications
  - Use of knowledge-based techniques
    - Zero knowledge or pre-populated job information consisting of connection between jobs
    - Similar cases retrieved are used to predict future jobs, reducing high startup overheads
  - Algorithm assessment
    - Two different workloads representing individual scientific jobs executed in LANL and set of workflows executed by three users

Use Case

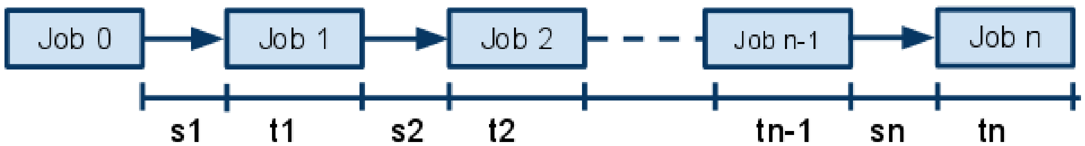
- Suite of workflows can differ from domain to domain
- WRF (Weather Research and Forecasting) as upstream node



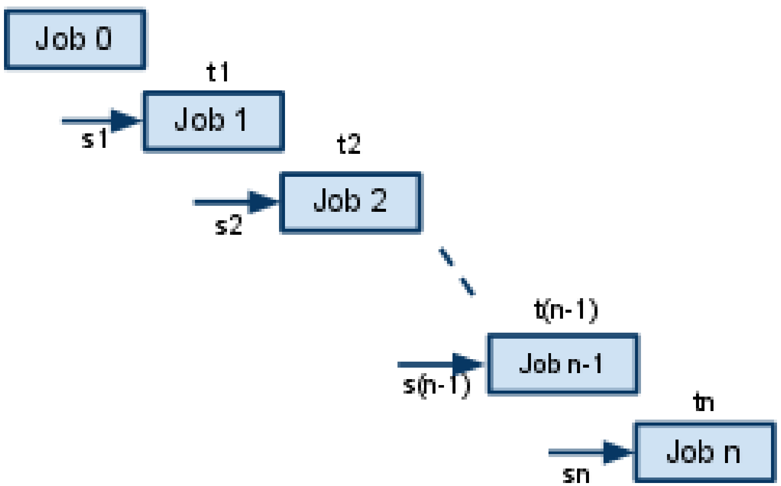
- User patterns reveal the sequence of jobs taking different users/domains into consideration
- Useful for a science gateway serving wide-range of mid-scale scientists

Role of Successful Predictions to Reduce Startup Overheads

- Largest gain can be achieved when our prediction accuracy is high and setup time (s) is large with respect to execution time (t)



$$T_{\propto} = \sum_{i=0}^{N} s_i + \sum_{i=0}^{N} t_i$$



$$T_{\propto} = (1-r)\sum_{i=0}^{N} s_i + \sum_{i=0}^{N} t_i$$

$$T_{\propto} = \sum_{i=0}^{N} s_i + \sum_{i=0}^{N} t_i - r\sum_{i=0}^{N} s_i$$

r = probability of successful prediction (prediction accuracy)

Percentage time reduction = $$\frac{r\sum_{i=0}^{N} s_i}{\sum_{i=0}^{N} (s_i + t_i)}$$

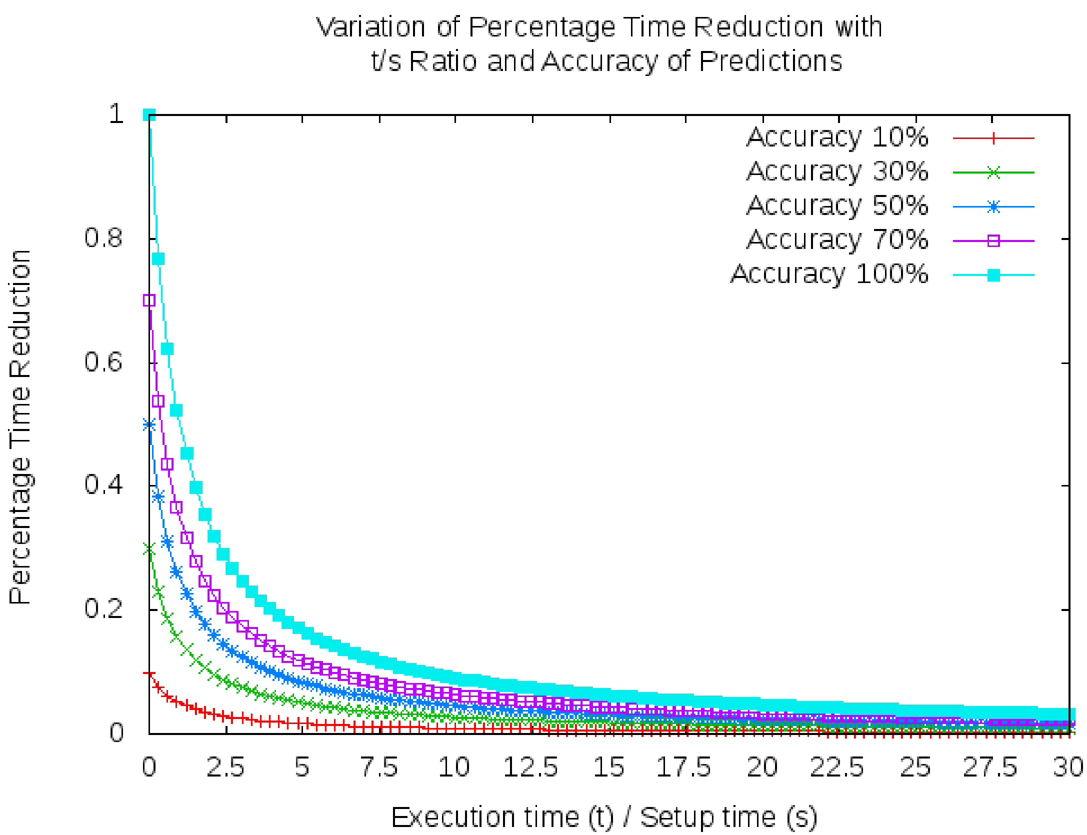
For simplicity, assuming equal job exec and startup times

Percentage time reduction = $$\frac{r*(s*N)}{(t+s)*N} = \frac{r*s}{(t+s)} = \frac{r}{\frac{t}{s}+1}$$

Eran Chinthaka Withana, Beth Plale "Usage Patterns to Provision for Scientific Experimentation in Clouds", CloudCom 2010, Indianapolis, IN, US

Relationship of Predictions to Execution Time

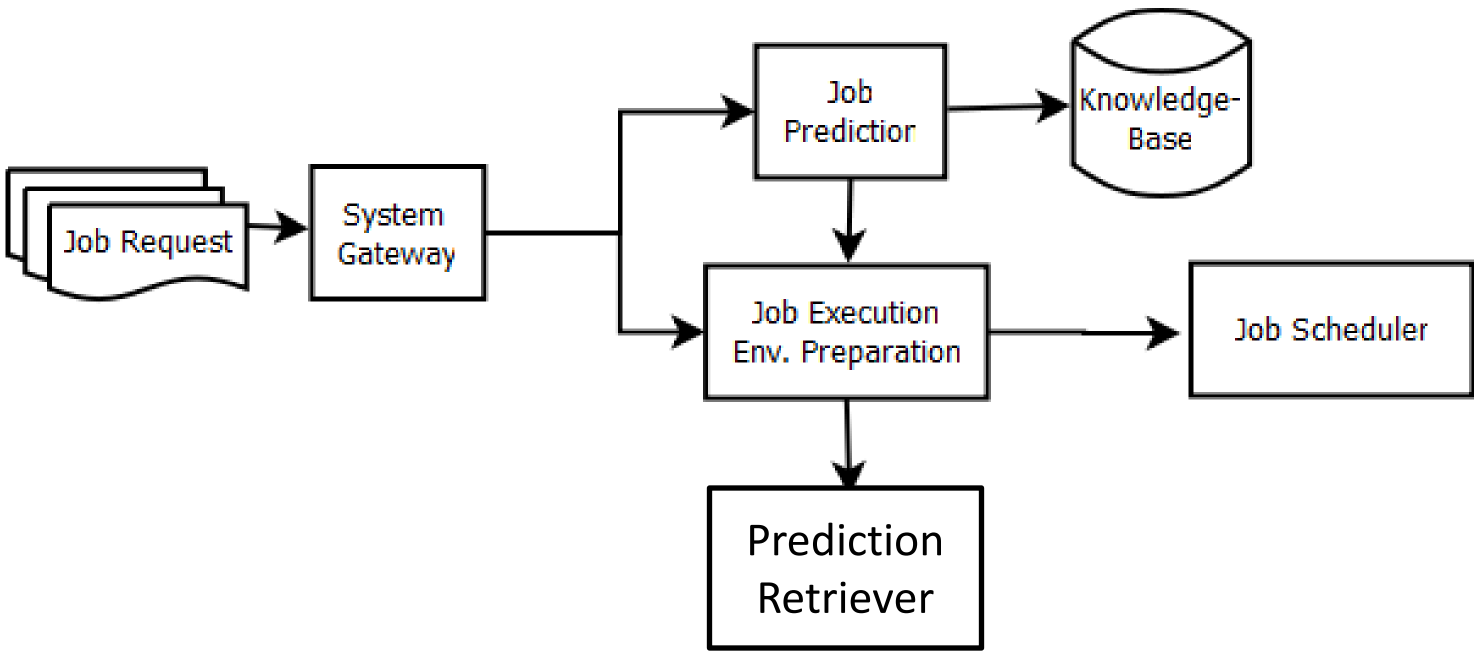
- Observations
  - Percentage time reduction increases with accuracy of predictions
  - Time reduction is reduced exponentially with increased work-to-overhead ratio
- Need to find the critical point for a given situation
  - Fixing the required percentage time reduction for a given t/s ratio and finding the required accuracy of predictions
- Cost of wrong predictions
  - Depends on compute resource



Accuracy of Predictions =
total successful future job predictions / total predictions

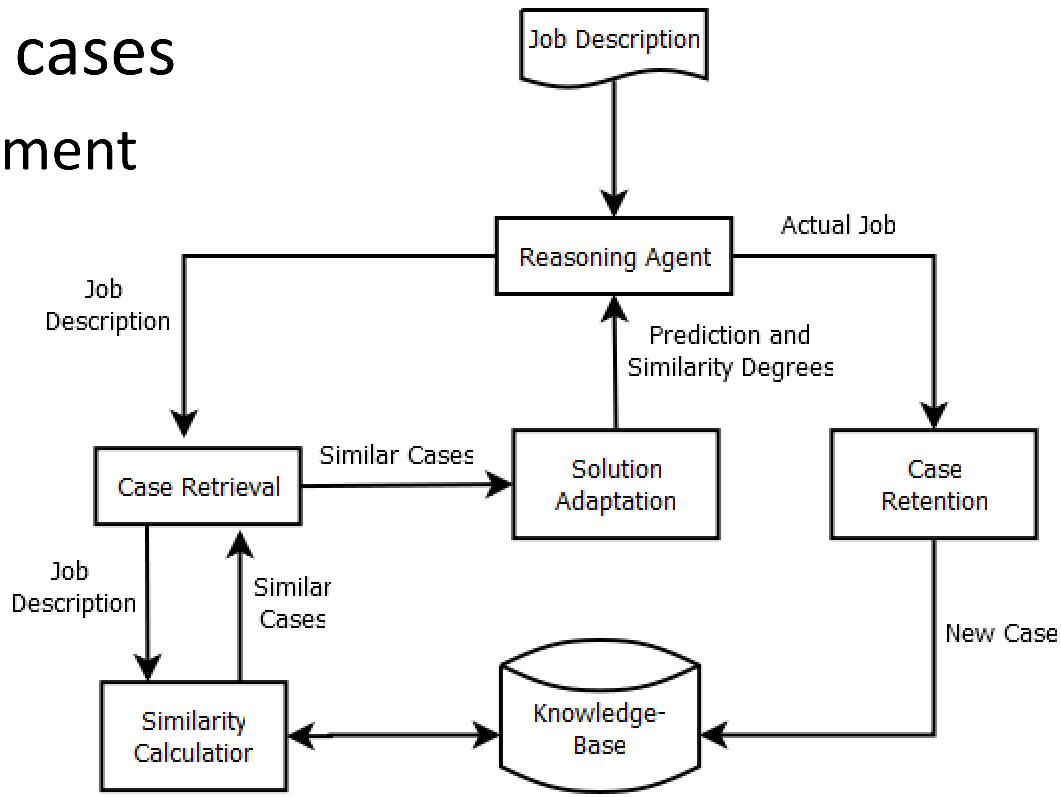
$$\text{Percentage time reduction} = \frac{r}{\frac{t}{s}+1}$$

Prediction Engine: System Architecture



Use of Reasoning

- Store and retrieve cases
- Steps
  - Retrieval of similar cases
    - Similarity measurement
    - Use of thresholds
  - Reuse of old cases
  - Case adaptation
  - Storage



Case Similarity Calculation

- Each case is represented using set of attributes
  - Selected by finding the effect on goal variable (next job)

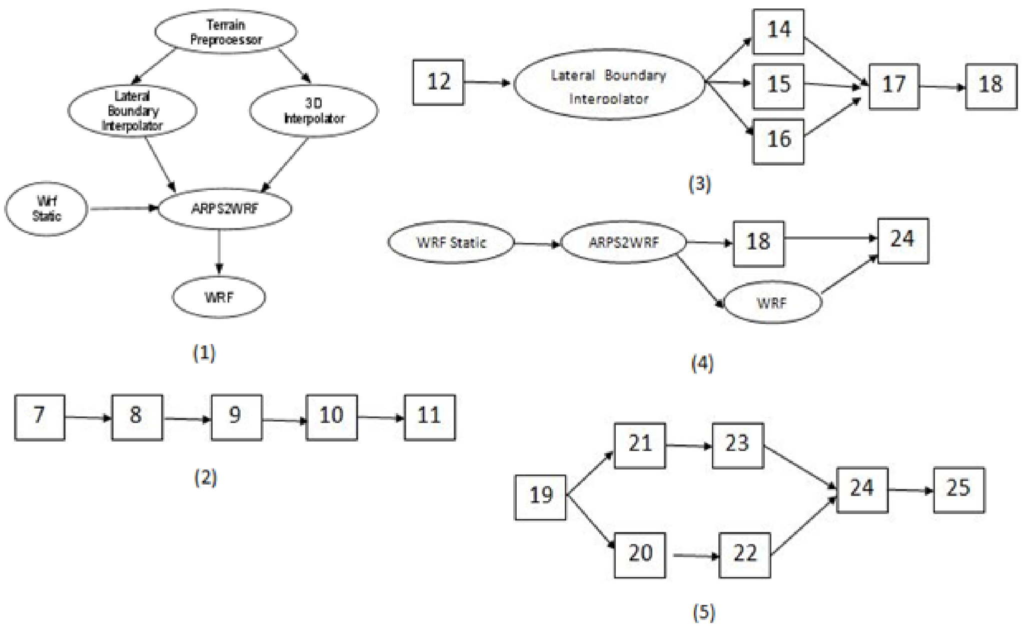
$$Sim(A,B) = \sum_{i=0}^{n} \omega_i(a_i b_i)$$

$$\forall X \neq S, Sim(X,G) < Sim(S,G)$$

$$Sim(S,G) > Threshold$$

Evaluation[1]

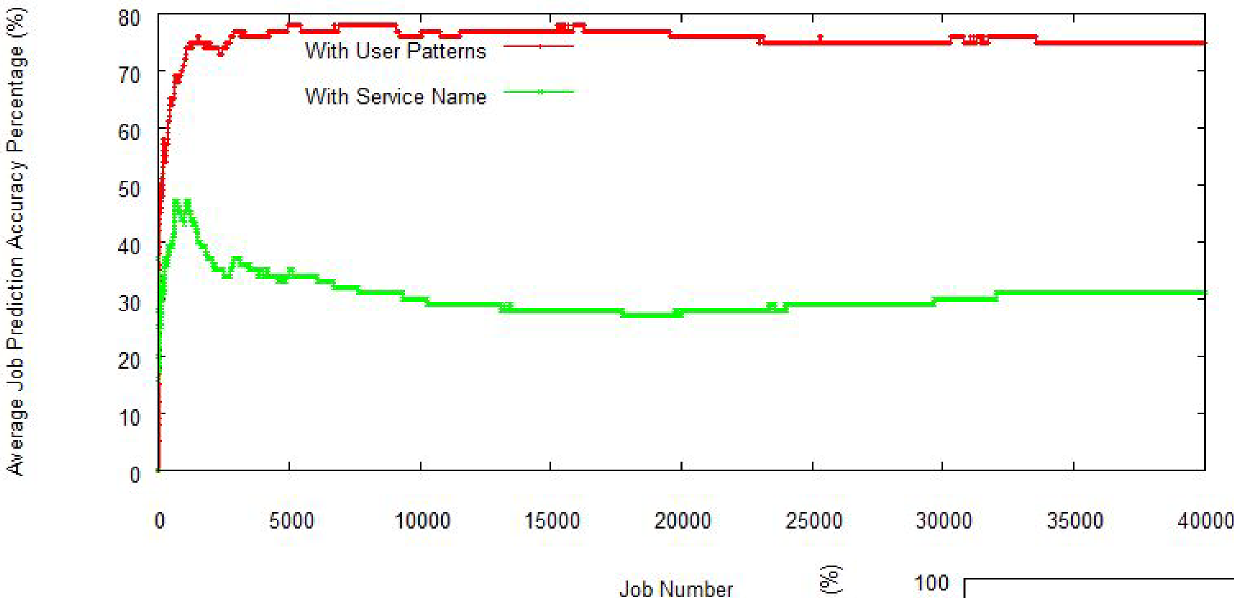
- Use cases
  - Individual job workload[1]
    - 40k jobs over two years from 1024-node CM-5 at Los Alamos National Lab
  - Workflow use case



User	Workflows in the experiment
User 1	Workflow 1, Workflow 2, Workflow 5
User 2	Workflow 2, Workflow 4
User 3	Workflow 2, Workflow 3, Workflow 4

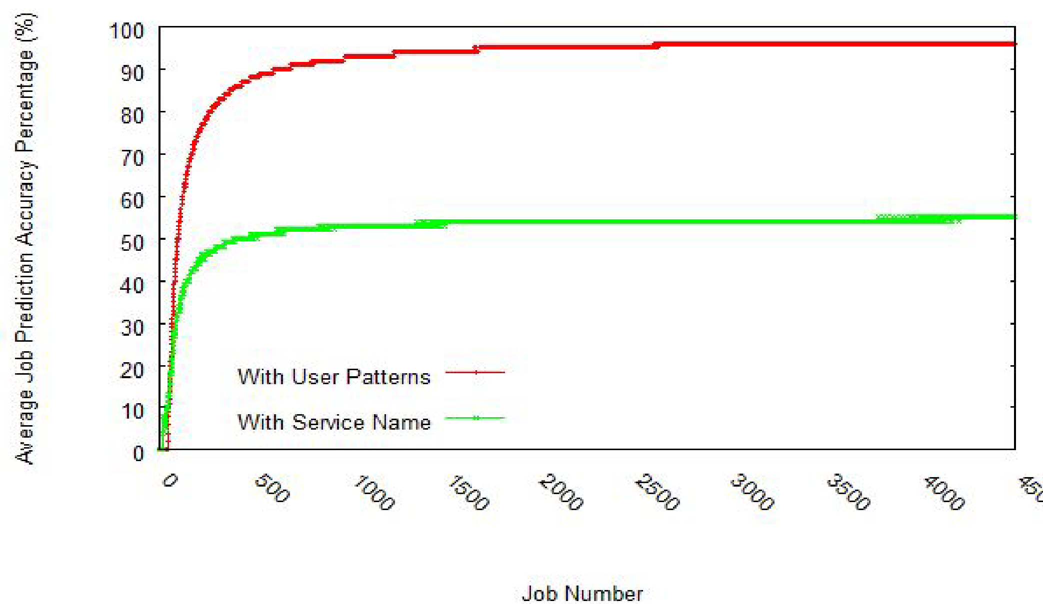
1: Parallel Workload Archive http://www.cs.huji.ac.il/labs/parallel/workload/

Evaluation: Average Accuracy of Predictions



Individual Jobs Workload

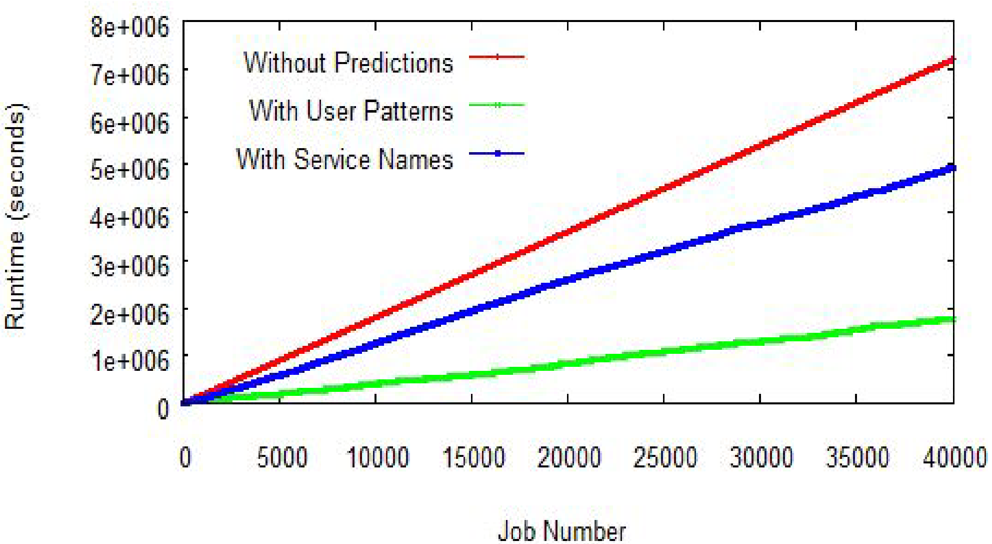
Workflow Workload



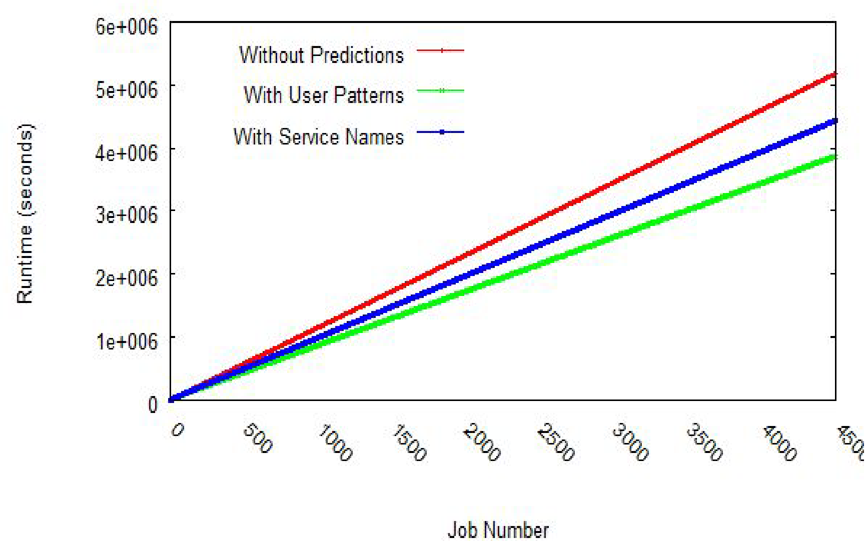
Evaluation: Time Saved

- Amount of time that can be saved, if the resources are provisioned, when the job is ready to run
- Startup time
  - Assumed to be 3mins (average for commercial providers)

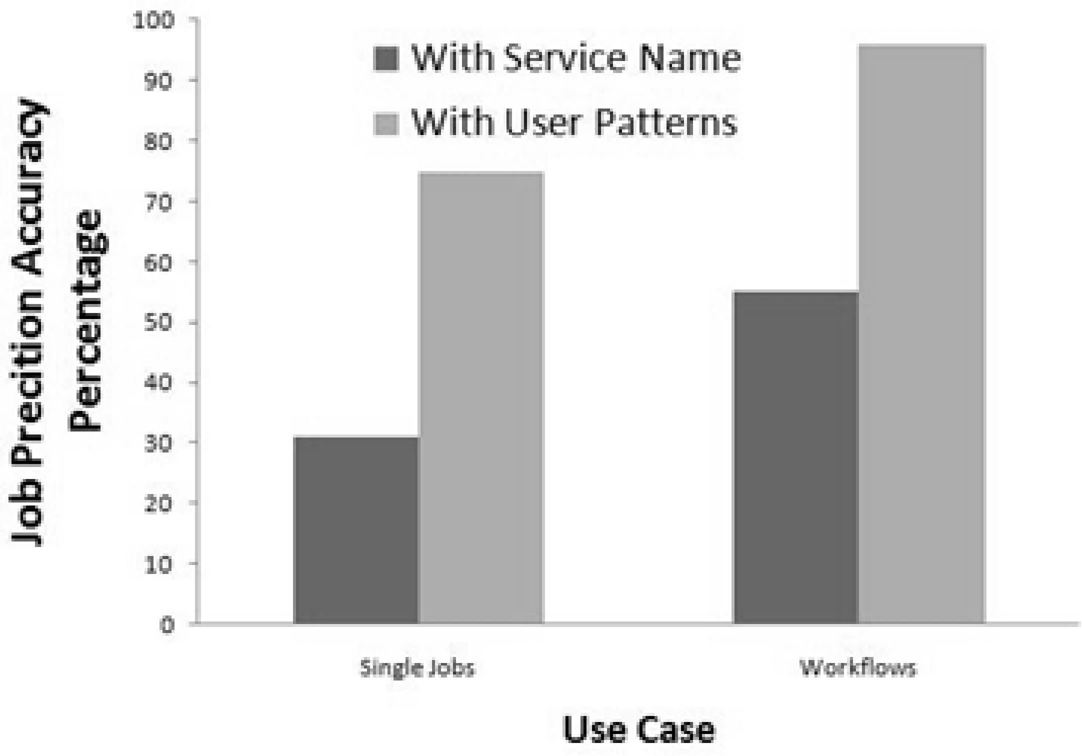
Individual Jobs Workload



Workflow Workload



Evaluation: Prediction Accuracies for Use Cases



Discussion and Future Work

- Accuracy
  - 78% for individual jobs
  - 96% for workflow workload
- Number of jobs required to make system stable depends on uniqueness and the distribution of unique applications
- Amount of time that can be saved, using future job prediction, is inversely proportional to t/s ratio
- More accurate methods to prune features and identify weights
- Evaluation of machine learning techniques as an alternative to knowledge-based systems
- Combining future job predictions with job reliability predictions to further improve throughput of job executions

Related Work

[1] M. Armbrust et al., “Above the clouds: A berkeley view of cloud computing,” EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, 2009.

[2] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2008.

[3] C. Catlett, “The philosophy of TeraGrid: building an open, extensible, distributed TeraScale facility,” in ACM International Symposium on Cluster Computing and the Grid. Published by the IEEE Computer Society, 2002.

[4] J. S. Chase, D. E. Irwin, L. E. Grit, J. D. Moore, and S. Sprenkle, “Dynamic virtual clusters in a grid site manager.” in HPDC. IEEE Computer Society, 2003, pp. 90–103.

[5] R. J. Figueiredo, P. A. Dinda, and J. A. B. Fortes, “A case for grid computing on virtual machines,” in ICDCS ’03: Proceedings of the 23rd International Conference on Distributed Computing Systems. Washington, DC, USA: IEEE Computer Society, 2003, p. 550.

[6] I. Foster, T. Freeman, K. Keahy, D. Scheftner, B. Sotomayer, and X. Zhang, “Virtual clusters for grid communities,” in CCGRID ’06: Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid. Washington, DC, USA: IEEE Computer Society, 2006, pp. 513–520.

[7] K. Keahey, T. Freeman, J. Lauret, and D. Olson, “Virtual workspaces for scientific applications,” Journal of Physics: Conference Series, vol. 78, p. 012038 (5pp), 2007.

[8] B. Sotomayor, K. Keahey, and I. Foster, “Overhead matters: A model for virtual resource management,” in VTDC ’06: Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing. Washington, DC, USA: IEEE Computer Society, 2006, p. 5.

………………………………………………………….

[12] F. Berman et al., “Adaptive computing on the grid using apples,” IEEE Transactions on Parallel and Distributed Systems, vol. 14, no. 4, pp. 369–382, 2003.

[13] F. Berman, A. Chien, K. Cooper, J. Dongarra, I. Foster, D. Gannon, L. Johnsson, K. Kennedy, C. Kesselman, J. Mellor-Crumme et al., “The GrADS project: Software support for high-level grid application development,” International Journal of High Performance Computing Applications, vol. 15, no. 4, p. 327, 2001.

[14] R. Buyya, D. Abramson, and J. Giddy, “Nimrod/G: An architecture for a resource management and scheduling system in a global computational grid,” in hpc. Published by the IEEE Computer Society, 2000, p. 283.

Thank You !!