

Batch Queue Prediction system for Supercomputing jobs

Abijit Bej

University of Southern California

Los Angeles CA 90089

abej@usc.edu

ABSTRACT

In this Directed Research, I describe how the predictions from Network Weather Services for availability of processors/nodes at different supercomputers can be used in job submission process. The prediction made by NWS is using a database containing the availability of nodes at different times at each of the sites in the last few years. The project involves analyzing the behavior of these predictions for selected sites for 7 days and then formulating an algorithm to split the required number of requested nodes into smaller chunks to check if the collective predictions for them improve.

1. INTRODUCTION

Batch queuing systems are used in most of the supercomputing centers to manage processor allocation. Many a times a user may have accounts at more than one facility and may wish to submit a job to a site where it will get completed the earliest and starts executing before its deadline. It becomes important to choose the best location/site for job execution as the amount of time a user's job will wait in batch queue can significantly affect the overall time the user waits from job submission to job completion.

Network Weather Services from University of California at Santa Barbara provides these queue delay predictions for individual jobs through Batch Queue Prediction services (QBETS). They provide this service for around 23 cluster supercomputers in the United States. I have used 9 of these sites for my project.

In my Directed Research I used these predictions from Network Weather Services and analyzed their behavior for a week's time. In the later part of the project I developed an algorithm that splits the required number of nodes into all possible permutations to query different sites in NWS.

1.1 ABOUT NWS

The Network Weather Service is a distributed system that periodically monitors and dynamically forecasts the performance various network and computational resources can deliver over a given time interval. The service operates a distributed set of performance sensors (network monitors, CPU monitors, etc.) from which it gathers readings of the instantaneous conditions. It then uses numerical models to generate forecasts of what the conditions will be for a given time frame. It is analogous to weather forecasting, and as such, the system inherits its name. NWS has been developed for use by dynamic schedulers and to provide statistical Quality-of-Service readings in a networked computational environment.

2. THE EXPERIMENT

The services provided by Network weather Services can be accessed using a client program that can query the NWS server before scheduling a batch job. To analyze the behavior of NWS at different times, the client program (in Perl) was designed to hit the server every hour for 7 days in a row. The 9 sites used for the DR are as follows with their corresponding queue types:

SITES	QUEUE(s)
ucteragrid	dque
ncsateragrid	dque, big, long
kittyhawk	dque
abe	dque
sdscteragrid	dque
mayhem	dque
uscbeuca	dque
utkeuca	dque
rencieuca	dque

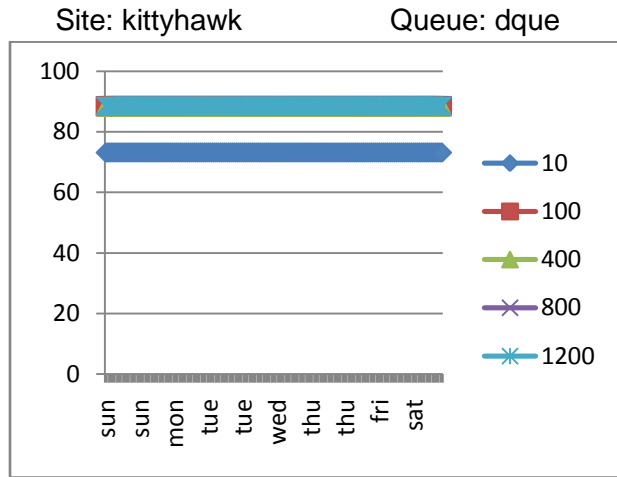
Table 1

Every hour, each of the 9 sites was queried for 10,100,400,800 and 1200 nodes for a

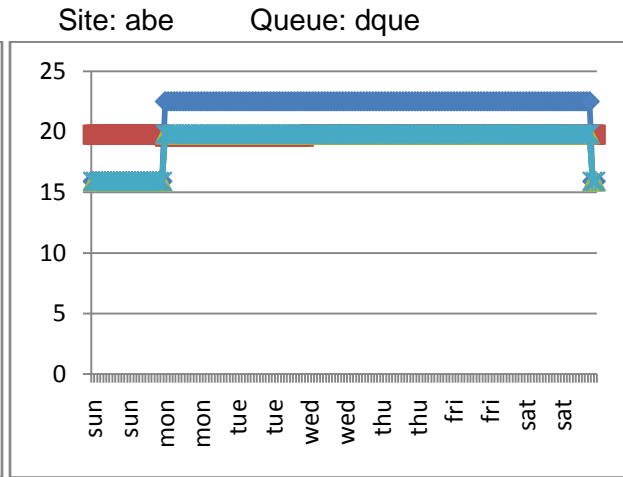
job having a fixed runtime of 3600 seconds and deadline of 1800 seconds. The corresponding queues at each site were selected appropriately and the confidence factor was assumed to be 0.95.

The response from NWS server for each hit is the probability (which is converted to percentage chances) of running the job by the specified deadline with 95% confidence at that particular site. The higher the probability the better it is to ask for those many nodes. The results from each of the sites were collected and graphs plotted for analysis.

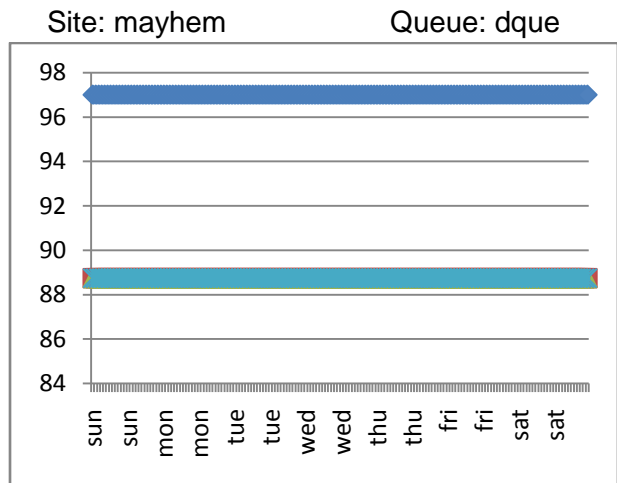
The Graphs for “Percentage chances of running by deadline” Vs “Time of the week”



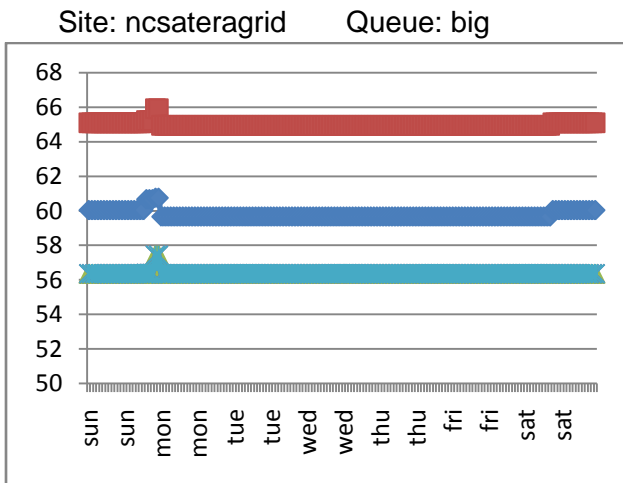
Graph 1



Graph 2

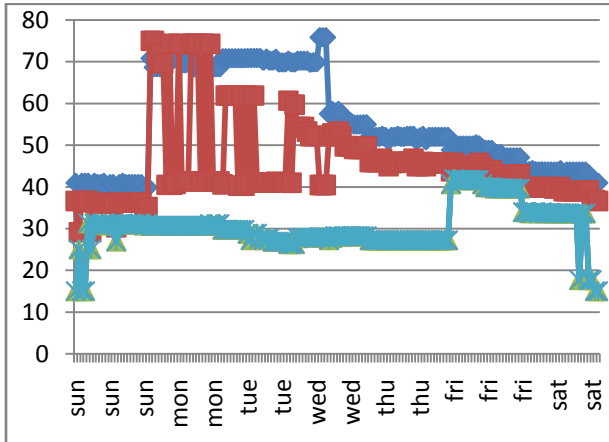


Graph 3



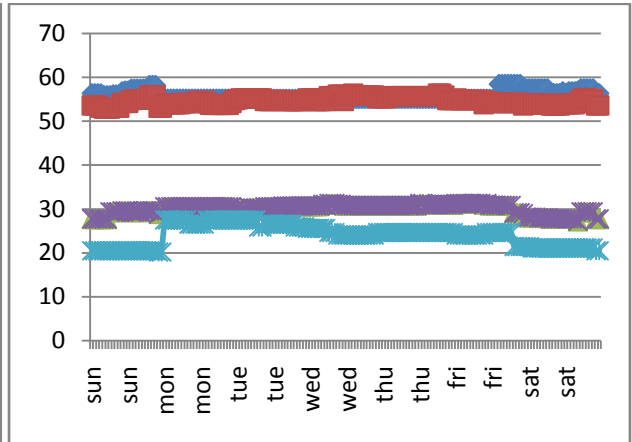
Graph 4

Site: ncsateragrid Queue: dque



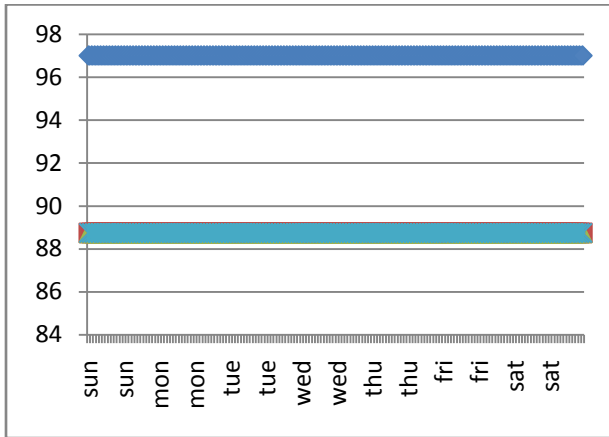
Graph 5

Site: ncsateragrid Queue: long



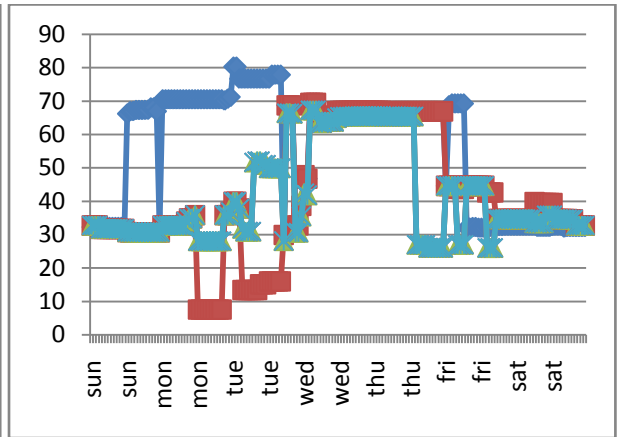
Graph 6

Site: rencieuca Queue: dque



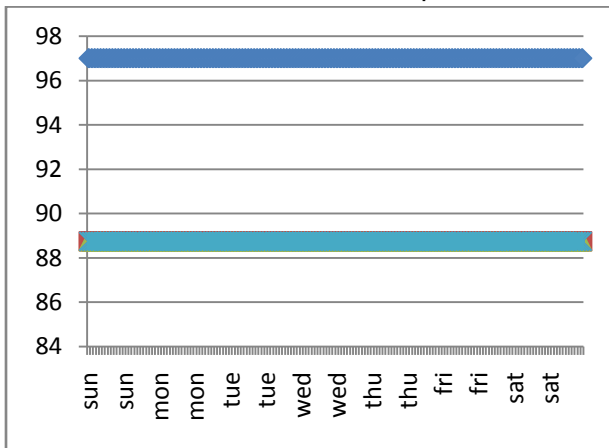
Graph 7

Site: sdscteragrid Queue: dque



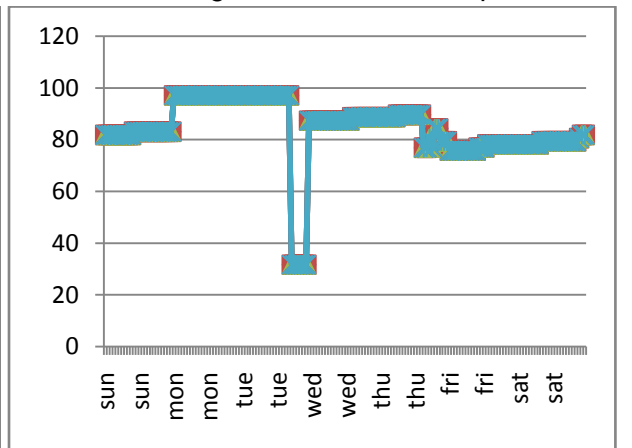
Graph 8

Site: ucsbeuca Queue: dque

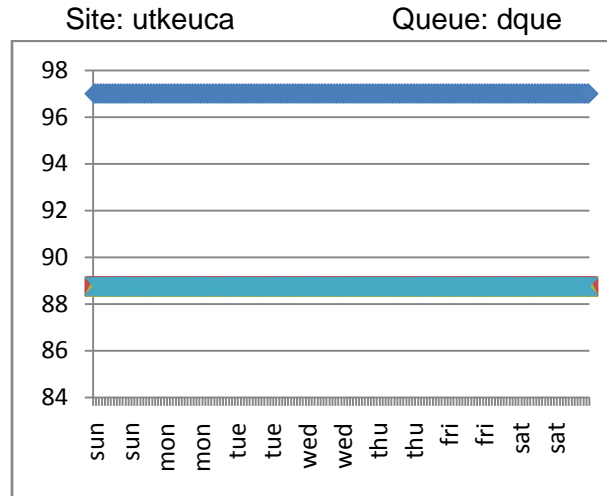


Graph 9

Site: ucteragrid Queue: dque



Graph 10



Graph 11

2.1 ANALYSIS

Amongst all the selected sites - abe, ncsateragrid (for all 3 queues), sdscteragrid and ucteragrid show remarkable variations in probabilities throughout the week for 10,100,400,800 and 1200 nodes. On the other hand, the rest of the sites - kittyhawk, mayhem, rencieuca, ucsbeuca and utkeuca remain flat throughout. The probabilities for them for specified number of nodes remains the same at all times. The above plots give an idea of which site to select at any given time of the week during job submission.

3. ALGORITHM

In the later part of the project an algorithm to break the total number of requested nodes was designed. It divides the requested nodes into equally sized requests that can then be requested from different sites. The intension of doing this was to check if the probability of obtaining the required number of nodes goes up if they are asked from multiple sites.

As an example, if there is an original request for 2000 nodes, it can be broken down into equally sized chunks of 1000,500,400 or 200. Each of the 11 queues (i.e. 8 sites with dque and ncsateragrid with dque, big and long) is then queried for some part of the total request in the

hope of getting higher probability of obtaining the desired number of nodes collectively. Only the ones that have high probability are selected. If say the top 4 sites respond with P_1 , P_2 , P_3 and P_4 probabilities of running by the deadline for 500 nodes each, the collective probability for 2000 nodes will be the multiplication of all the individual probabilities ($P_1 * P_2 * P_3 * P_4$).

3.1 RESULTS

The algorithm takes any number of requested nodes and breaks them equally into manageable pieces that can be acquired from different sites in NWS. For this experiment requested numbers of nodes have been taken to be 2000, 1000 and 500. Each of them is broken into equal chunks as long as they can be acquired from unique different sites. Since there are only 11 queues to acquire the nodes from, a maximum of 11 pieces can be made. A snippet of a single run of the algorithm is displayed below:

Output:

request,break-up,site,probability

```
2000,2000,ucteragrid,0.942801376777
    PROB of 1 sites : 0.942801376777
2000,1000,ucteragrid,0.942801376777
2000,1000,rencieuca,0.887455207346
    PROB of 2 sites : 0.836693991313727
2000,500,ucteragrid,0.942801376777
2000,500,rencieuca,0.887455207346
2000,500,utkeuca,0.887455207346
2000,500,mayhem,0.887455207346
    PROB of 4 sites : 0.65896073027802
2000,400,ucteragrid,0.942801376777
2000,400,rencieuca,0.887455207346
2000,400,utkeuca,0.887455207346
2000,400,mayhem,0.887455207346
2000,400,ucsbeuca,0.887455207346
    PROB of 5 sites : 0.584798131521752
2000,250,ucteragrid,0.942801376777
2000,250,rencieuca,0.887455207346
2000,250,utkeuca,0.887455207346
2000,250,mayhem,0.887455207346
2000,250,ucsbeuca,0.887455207346
2000,250,kittyhawk,0.884653156398
2000,250,sdscteragrid,0.733568046209
2000,250,ncsateragrid_dque,0.613128379147
    PROB of 8 sites : 0.23268630944131
2000,200,ucteragrid,0.942801376777
2000,200,rencieuca,0.887455207346
2000,200,utkeuca,0.887455207346
2000,200,mayhem,0.887455207346
2000,200,ucsbeuca,0.887455207346
2000,200,kittyhawk,0.884653156398
2000,200,sdscteragrid,0.733568046209
2000,200,ncsateragrid_dque,0.613128379147
2000,200,ncsateragrid_big,0.607483896919
2000,200,ncsateragrid_long,0.410850476896
    PROB of 10 sites : 0.0580750238867193
BEST : 0.942801376777,ucteragrid
-----
1000,1000,ucteragrid,0.942801376777
    PROB of 1 sites : 0.942801376777
1000,500,ucteragrid,0.942801376777
1000,500,rencieuca,0.887455207346
    PROB of 2 sites : 0.836693991313727
1000,250,ucteragrid,0.942801376777
1000,250,rencieuca,0.887455207346
1000,250,utkeuca,0.887455207346
1000,250,mayhem,0.887455207346
    PROB of 4 sites : 0.65896073027802
1000,200,ucteragrid,0.942801376777
1000,200,rencieuca,0.887455207346
1000,200,utkeuca,0.887455207346
1000,200,mayhem,0.887455207346
1000,200,ucsbeuca,0.887455207346
    PROB of 5 sites : 0.584798131521752
```

```
1000,125,ucteragrid,0.942801376777
1000,125,rencieuca,0.887455207346
1000,125,utkeuca,0.887455207346
1000,125,mayhem,0.887455207346
1000,125,ucsbeuca,0.887455207346
1000,125,kittyhawk,0.884653156398
1000,125,sdscteragrid,0.733568046209
1000,125,ncsateragrid_dque,0.613128379147
    PROB of 8 sites : 0.23268630944131
1000,100,ucteragrid,0.942801376777
1000,100,rencieuca,0.887455207346
1000,100,utkeuca,0.887455207346
1000,100,mayhem,0.887455207346
1000,100,ucsbeuca,0.887455207346
1000,100,kittyhawk,0.884653156398
1000,100,sdscteragrid,0.733568046209
1000,100,ncsateragrid_dque,0.613128379147
1000,100,ncsateragrid_big,0.607483896919
1000,100,ncsateragrid_long,0.410850476896
    PROB of 10 sites : 0.0580750238867193
BEST : 0.942801376777,ucteragrid
-----
500,500,ucteragrid,0.942801376777
    PROB of 1 sites : 0.942801376777
500,250,ucteragrid,0.942801376777
500,250,rencieuca,0.887455207346
    PROB of 2 sites : 0.836693991313727
500,125,ucteragrid,0.942801376777
500,125,rencieuca,0.887455207346
500,125,utkeuca,0.887455207346
500,125,mayhem,0.887455207346
    PROB of 4 sites : 0.65896073027802
500,100,ucteragrid,0.942801376777
500,100,rencieuca,0.887455207346
500,100,utkeuca,0.887455207346
500,100,mayhem,0.887455207346
500,100,ucsbeuca,0.887455207346
    PROB of 5 sites : 0.584798131521752
500,50,ucteragrid,0.942801376777
500,50,kittyhawk,0.884653156398
500,50,rencieuca,0.874012343602
500,50,utkeuca,0.874012343602
500,50,mayhem,0.874012343602
500,50,ucsbeuca,0.874012343602
500,50,ncsateragrid_dque,0.613128379147
500,50,ncsateragrid_big,0.607483896919
500,50,sdscteragrid,0.489682307464
500,50,ncsateragrid_long,0.410850476896
    PROB of 10 sites : 0.0364710192830339
BEST : 0.942801376777,ucteragrid
-----
```

The same algorithm is run for 24 hours to check if there is any variation in the results. Continuing with the above experiment for 24 hours gives the following plot (Graph 12):

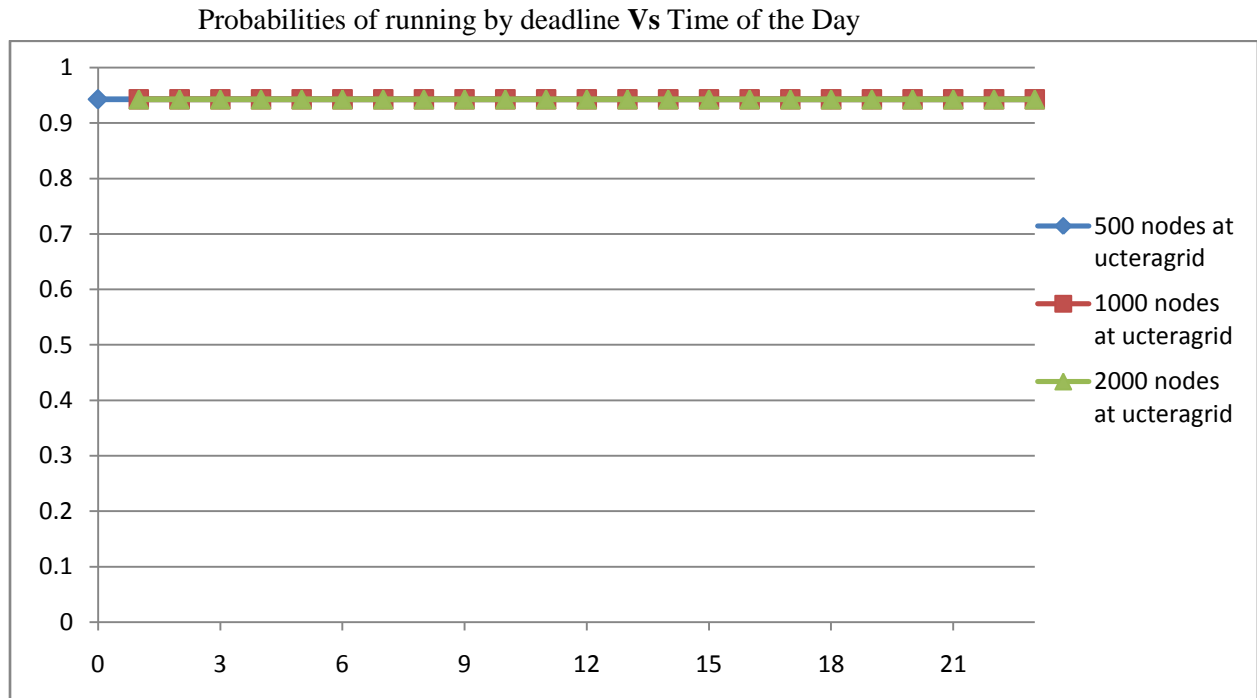


Figure 12

3.2 ANALYSIS

The algorithm is tested even for 4000 and 3000 nodes and the results are the same as above. ucteragrid returns 94.28% chances of running by the deadline for any number of nodes. So, no matter how much the size of requested nodes is sub-divided into, ucteragrid alone still becomes the ideal site to grab all the nodes from. This leaves us to question the accuracy of Network Weather Services. According to what is mentioned on the NWS website “The system tracks the accuracy (using prediction error as an accuracy measure) of all predictors, and uses the one exhibiting the lowest cumulative error measure at any given moment to generate a forecast. In this way, the NWS automatically identifies the best forecasting technique for any given resource.”

Thus, even if NWS tried to identify and use the best forecasting technique, all that can be inferred from the results is that the heuristics used by NWS for forecasting these predictions may not be accurate at all times and for every individual site.

4. CONCLUSIONS

In this directed research, I showed how the predictions for availability of nodes at different supercomputing sites can be used effectively in the job submission process. It can be incorporated into the Pegasus system before it schedules any workflow.

The algorithm used for breaking up the requested number of nodes into manageable requests did not perform as expected. I was expecting a higher collective prediction value from the smaller requests. The reason being, when a small number of nodes are requested from a site the chances of acquiring them and executing before the deadline is expected to be higher as compared to requesting a large number of nodes.

In future, I would like to break the requests into unequal granularities and would like to work on trying to avoid inconsistencies from individual sites so as to make the predictions as accurate as possible.

5. ACKNOWLEDGEMENTS

I am obliged to Dr. Ewa Deelman of Information Sciences Institute (ISI) for giving me an opportunity to work on this Directed Research. It was because of her constant guidance and involvement in the DR, I could complete it. I would also like to acknowledge Prof. Aiichiro Nakano of USC Computer Science Dept. and Philip Maechling of Southern California Earthquake Center (SCEC) for their encouragement and help in formulating the topic during the inception of this DR. I am also grateful to Karan Vahi and Gaurang Mehta of ISI for their valuable guidance in understanding the Pegasus workflow management system.

6. REFERENCES

[1] Ewa Deelman, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Sonal Patil, Mei-Hui Su, Karan Vahi, Miron Livny “*Pegasus: Mapping Scientific Workflows onto the Grid*”

[2] Ewa Deelman, Gurmeet Singh, Mei-Hui Sua, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Karan Vahi, G. Bruce Berriman, John Good, Anastasia Laity, Joseph C. Jacob and Daniel S. Katz “*Pegasus: A framework for mapping complex scientific workflows onto distributed systems*” Scientific Programming 13 (2005) 219–237

[3] Arun Ramakrishnan, Gurmeet Singh, Henan Zhao, Ewa Deelman, Rizos Sakellariou, Karan Vahi, Kent Blackburn, David Meyers and Michael Samidi “*Scheduling Data-Intensive Workflows onto Storage-Constrained Distributed Resources*”

[4] Kevin Lee, Norman W. Paton, Rizos Sakellariou, Ewa Deelman, Alvaro A. A. Fernandes, Gaurang Mehta “*Adaptive Workflow Processing and Execution in Pegasus*”

[5] John Brevik, Daniel Nurmi, and Rich Wolski “*Using Model-based Clustering to Improve*

Predictions for Queuing Delay on Parallel Machines”

[6] John Brevik, Daniel Nurmi, and Rich Wolski “*Predicting Bounds on Queuing Delay for Batch-scheduled Parallel Machines*”

[7] Daniel Nurmi, Rich Wolski, John Brevik, Graziano Obertelli “*QBETS: Batch Queue Prediction System (formerly known as BQP)*” Presentation

[8] UCSB Network Weather Services “<http://nws.cs.ucsb.edu/ewiki/nws.php?id=Batch+Queue+Prediction>”