Evaluation of a Workflow Scheduler Using Integrated Performance Modelling and Batch Queue Wait Time Prediction

Daniel Nurmi, Anirban Mandal, John Brevik, Chuck Koelbe,
Rich Wolski, Ken Kennedy

Motivation...

To schedule application components which have dependencies, workflow schedulers are used.

Workflow schedulers produces an efficient mapping of the components on to the processors.

The waiting time of the jobs in queue are not considered.

To improve the efficiency scheduling, batch queue wait times need to be taken into consideration.

To combine workflow scheduling, performance modelling and batch queue prediction

Performance Modelling

- To evaluate the execution time of the application / component on any given set of resources.
- Floating-point operation counts are gathered
- Memory access pattern of the component is analyzed
- Memory Reuse Distance (MRD) data is used to model the behaviour.
- The Execution time is evaluated as below:

$$EstExecTime(psize) = \frac{A + B + C + D}{CpuClock(arch)}$$

$$A = k_0 \times \frac{totalFp(psize)}{FpPipelineNum(arch)} \times FpRptRt(arch)$$

$$B = k_1 \times L1MissCnt(psize) \times L1MissPnlty(arch)$$

$$C = k_2 \times L2MissCnt(psize) \times L2MissPnlty(arch)$$

$$D = k_3 \times L3MissCnt(psize) \times L3MissPnlty(arch)$$

Prediction of Wait Times in Batch Queues

- Prediction of exact value is nearly impossible & Average / Expected wait time is of limited practical use
- A bound on the queue wait time is more relevant
- Binomial Method Batch-Queue Predictor (BMBP) is used

X – Random Variable, X_q – q quantile distribution of X
==> any x from X will be greater than X_q with prob (1-q)

Prob. that k or fewer observations are > X_q is given by

$$\sum_{j=0}^{k} \binom{n}{j} \cdot q^{n-j} \cdot (1-q)^{j}$$

We find the smallest value of k for which the above value is greater than a specified confidence level. The kth value in a sorted set of observations will then be >= X_q quantile with the specified level of confidence.

Scheduler Design

Original Workflow Algorithm -->

It is modified to consider batch queue wait times

The Availability time of each resource is populated from using the predicted wait time for that resource.

95% upper bound on the median queue wait time prediction is used as the predicted wait time.

```
foreach heuristic do
    while all components not mapped do
        Find availComponents; // satisfy dependencies
        Calculate the rank matrix;
        findBestSchedule(availComponents, heuristic);
    endwhile
endforeach
Select mapping with minimum makespan among three;
Output selected mapping;
```

Algorithm 1. Workflow Scheduling

```
while all availComponents not mapped do
    foreach Component, j do
        foreach Resource, R do
            ECT(j,R)=rank(j,R)+EAT(R);
        endforeach
        Find minECT(j,R) over all R;
        Find 2nd-minECT(j,R) over all R;
    endforeach
    Calculate min(minECT(j,R)) over all j; //min-min
    Calculate max(minECT(j,R)) over all j; //max-min
    Calculate min(2nd-minECT(j,R)-minECT(j,R))
    over all j; //sufferage
    Store mapping;
    Update EAT(R) and makespan;
endwhile
```

Algorithm 2. findBestSchedule

Experiments & Results

Comparing the makespans for workflow schedules with and without considering the queue wait times.

Five different supercomputers across the country constituted the experimental environment. All the these systems are monitored by batch-queue prediction software and have had performance models pre-calculated for the application used.

EMAN application is used as a test workflow application (Bio-imaging application dealing with 3D reconstruction of single particles from electron micrographs)

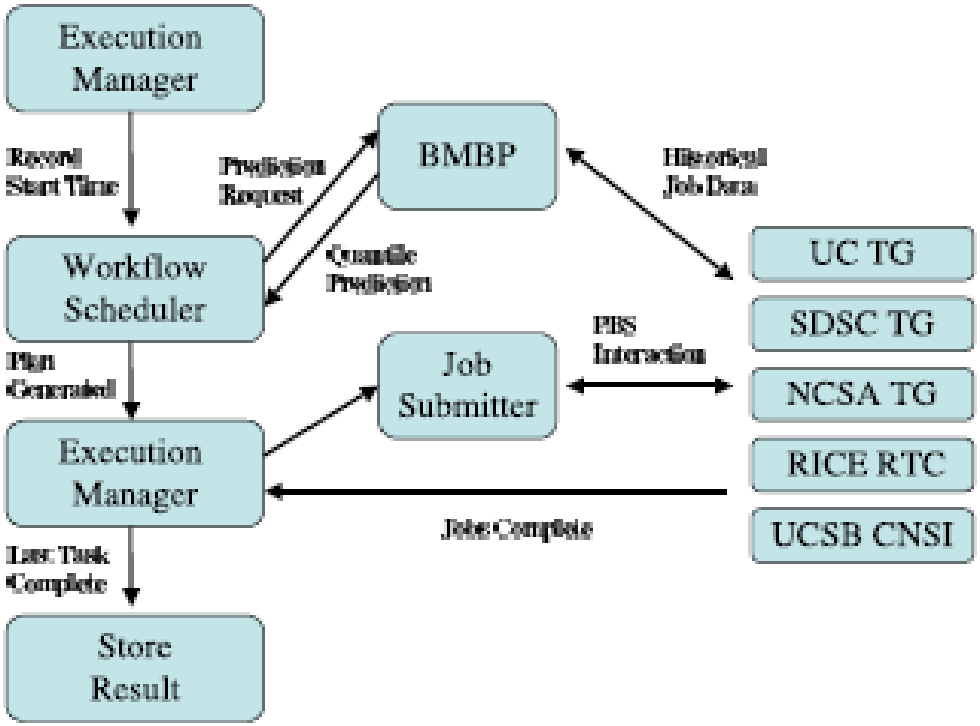

Figure 3. Experimental testbed architecture.

Two experiments were conducted, one with small EMAN runs and the other with realistic EMAN data.

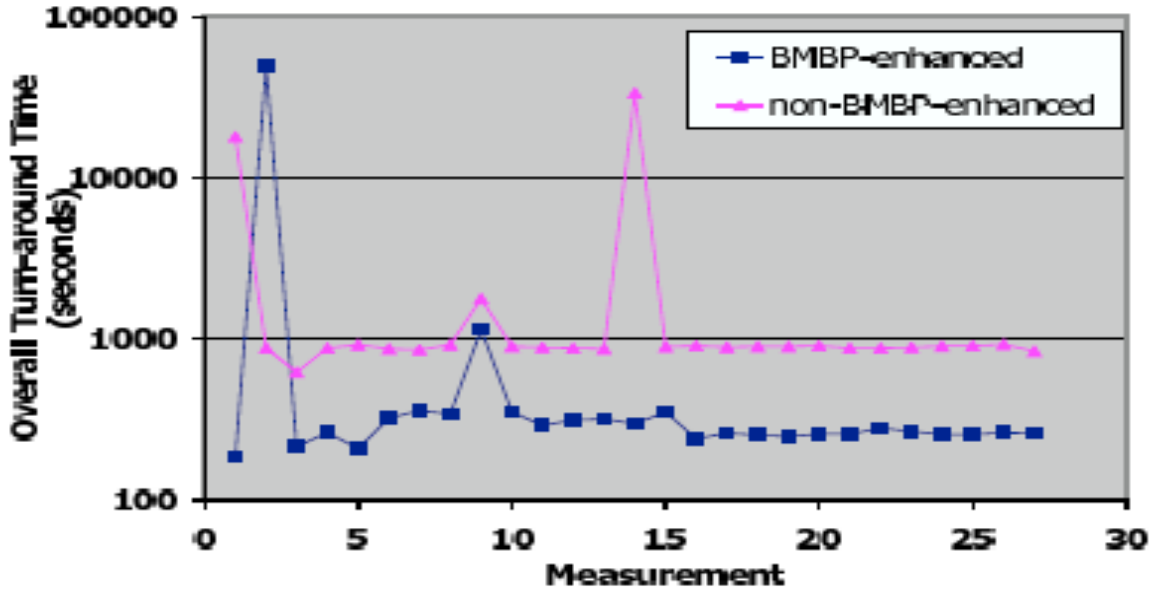

Figure 4. Total turn-around time for small EMAN runs for both BMBP enhanced schedules and non-BMBP enhanced schedules

	Avg. Res. Count	Avg. Site Count
BMBP	90.0	1.8
Non-BMBP	98.0	2

Table 1. Table of average number of unique resources and unique sites used for each conducted experiment.

- BMBP enhanced scheduler contacted, on average, fewer unique nodes
- Indicated that scheduler had decided to run few tasks in serial on a single node than use more in parallel.
- Couldn't have happened without considering queue wait times
- For realistic data, BMBP schedule took about half a day.
- Non-BMBP schedule could not be completed after two days