ANALYSIS

(1)

There is no notable time difference in the job execution times for the same class of job. If there is any difference it is because of network latency. There is a difference in execution time when the number of computers servicing the tasks are different. Processing is faster with when more computers are running.

But by Amdhal's law we cannot increase the execution speed by more than 1/(1-P) where P is the percentage of tasks which can be parallelized. We observe that with the following data :

For c=1, the execution time is approximately 6ms
For c=2, it is approximately 4ms
For c=4 it is approximately 2ms
But For c=8 it is still close to 2ms and there is not much change. As N (no of computers increases) the improvement in speed is not significant.

(2)

The measurements which can be added to the architecture are:

a) Time that a task has spent in the task queue - This helps in the analysis of the optimal rate of parallization that can be done for a given task. If the tasks are part of a large queue, it means the number of parallel tasks that are trying to be executed is more than the threshold of the given distributed system.

b) Measuring the time taken for decomposition - This can be used to analyse the rate of generation of the tasks to the rate at which they are serviced by the computer.

c) Measuring the time taken for composition of the task - This again will help in finding the optimal parallization that can be done for a task. If composition takes a lot of time then it defeats the purpose of parallization to increase the speed of execution, as the client spends a long time in composing the results of the parallel tasks.

(3)

The architecture can be improved to handle multiple tasks at the same time.
In a real world scenario there are multiple tasks that will be required by the client which have to be excecuted in parallel. In such a situation the current architecture does not support the parallel running of two or more tasks. Since the result queue can hold

objects of one type when one particular task is run. It is not possible to handle multiple tasks and process their results separately.