

**Performance Measure of Microsoft Compute Cluster Server (CCS) 2003
Robust Biomarker Discovery
-Performance Tuning**

June 27th, 2006

Purpose:

The experiment was conducted to show two performance tuning issues when using distributed computing technologies, Microsoft CCS and MathWorks Distributed Computing Toolbox (DCT), for bioinformatics applications.

- (a) It is better to load data locally on each compute node than loading data from head node.
- (b) It is better to create the number of tasks that are equal to the number of compute nodes in a cluster than to create the number of tasks that are equal to the number of independent experiments in a job.

Demo Codes:

Demo codes are generated by CBIL graduate students Yibin Dong and Yuanjian Feng, Joseph Martin from MathWorks, and Ming Xu from Microsoft.

Dataset:

The dataset, Muscular Dystrophy 12 groups disease and 1 normal group, is provided by the Children's National Medical Center. The number samples are 125. The number genes are 11,252.

Experiments

The testing bed is a CCS cluster that consists of two DELL XPS x64 machines (Intel Pentium 3.8GHz, 2G RAM).

In this robust biomarker discovery experiment, leave-one-out (LOO) algorithm was implemented to perform biomarker (gene) selection. A sample is excluded from the whole dataset a time. The remaining dataset will be used as the input to the gene selection algorithm. This independent test will be repeated for a number of times, which equal to the number of samples in the study. The size of the dataset after LOO is about 11.5M.

We tested on two scenarios:

Scenario 1: A job with 2 tasks (= number of compute node in the cluster) is created for testing. Each task takes approximately half of the LOO input data indices ($125/2 = 62$ or 63) as input. The compute node will load the data locally. After all tasks are finished, the head node collects the results from each compute node and saves in an output file.

Scenario 2: A job with 125 tasks is created for testing. Each task takes a different LOO input data, and generates an output. After all tasks are accomplished, the head node collects the results from each compute node and saves in an output file.

Time consumption in two scenarios is shown in Table 1 and Figure 1.

	# of nodes	# of seconds	
		1	2
Scenario #1: locally loading data; number of tasks equal to number of nodes in a cluster.	Create Tasks	0.09	0.06813
	computing	153.732926	83.371983
	scenario 1	153.82	83.440113
Scenario #2: load data from head node; number of tasks equal to number of independent tasks in the study.	Create Tasks	161.722278	161.722278
	computing	624.343	424.07207
	scenario 2	786.065278	585.794348

Table 1. Time spent on creating tasks and computing in two scenarios.

Figure 1 shows the running time of completing the job when varying the number of nodes in a cluster.

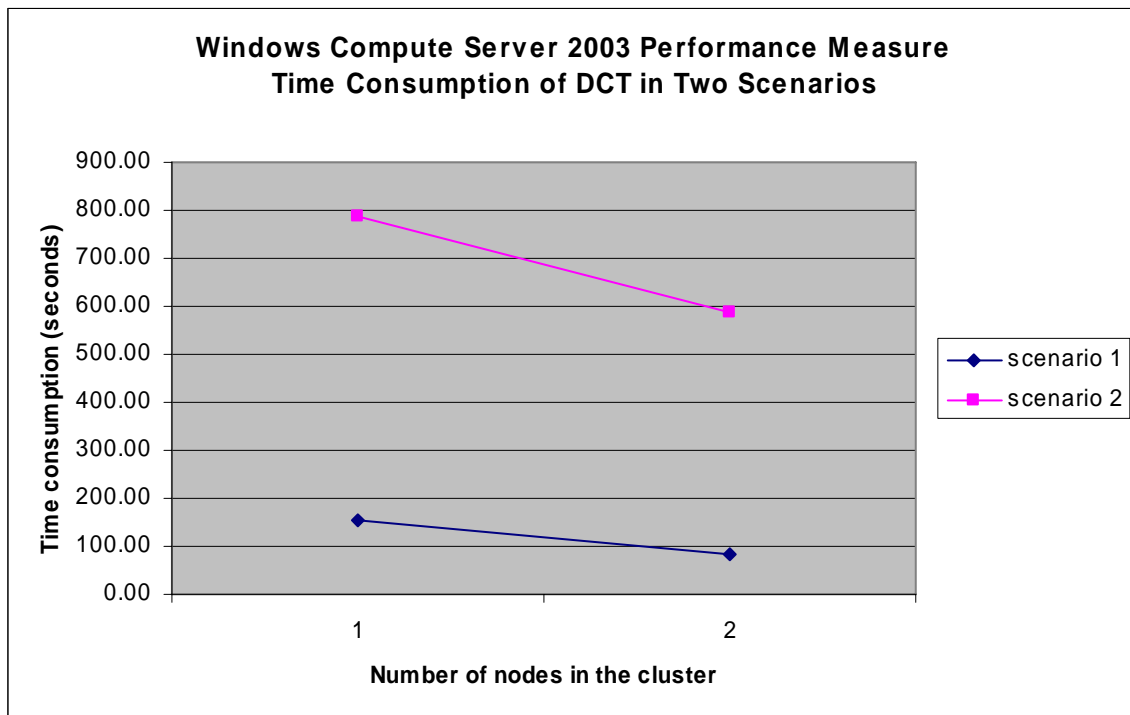


Figure 1. The running time to accomplish a job in two scenarios.