

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

July 7, 2006

Purpose:

The experiment was conducted to show scalability performance measure of CCS when the complexity of a job is fixed using distributed computing technologies, Microsoft CCS and MathWorks Distributed Computing Toolbox (DCT), for bioinformatics applications.

Demo Codes:

Demo codes are generated by CBIL graduate students Yibin Dong and Yuanjian Feng, Jos 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 (CNMC, <http://www.cnmcresearch.org>). The number samples are 125. The number genes are 11,252.

Experiments

The testing bed is a 16-node CCS cluster that consists of 16 HP Proliant DL145 Generation 2 Server (dual core AMD Opteron processor 270, 2.01 GHz, 1G RAM). There are total 16 Matlab Distributed Computing Engine (MDCE) licenses available.

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.

A job is created with n tasks (n equals to the number of MDCE licenses used in the cluster, and $1 \leq n \leq 16$). The total number of independent experiments (125 in our study) is divided into stratified n subsets. Each task is corresponding to a subset. 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.

Time consumption is shown in Figure 1. Since the complexity of the job is fixed, when the number of MDCE workers increases, the time consumption nonlinearly decreases and tends to converge. Adding more workers to the cluster doesn't help to reduce the computational time too much. For example, when the number MDCE workers increase from 10 to 11, theoretically, it only decreases 0.9191% of the running time. When the number workers are above 13, the time consumption reduction is neglectable.

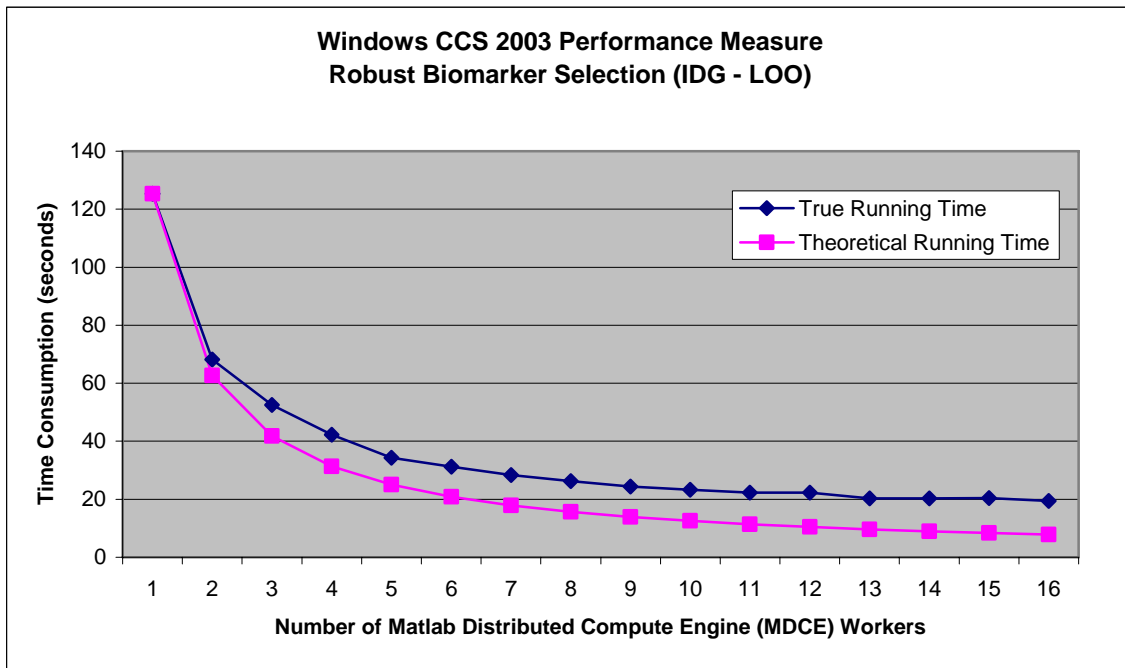


Figure 1. The running time to accomplish a job when the number of MDCE workers increase