

Performance Measure of Microsoft Compute Cluster Server (CCS) 2003 Predictor Performance Estimation

July 31, 2006

Purpose:

The experiments were conducted to show the benefits of distributed computing technologies, Microsoft CCS and MathWorks Distributed Computing Toolbox (DCT), for bioinformatics applications. The purpose was to show:

- (a) Given a fixed complexity of an algorithm, the time consumption T will be decreased by a factor $\frac{1}{N}$ as N increases, where N is the number of distributed computing workers in the cluster.
- (b) Increasing the number of distributed computing workers in the cluster will enable the cluster to handle jobs with higher complexity while maintain almost the same time consumption.

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, Breast Cancer 1 disease and 1 normal group, is from Nature publication (t Veer et al.). The number samples are 78. The number genes are 24,481.

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/workers available.

In the robust biomarker discovery experiment, leave-one-out (LOO) algorithm was implemented to perform biomarker (gene) selection. Total 199 independent gene sets (predictors) were selected. A prediction performance estimate will be calculated for each predictor through cross validation. Usually, the number of iterations in a cross validation test varies from tens to hundreds, the more the better in terms of statistics.

Experiment 1: In our first experiment, we tested the prediction performance of all 199 predictors using 3-fold cross validation. Each cross validation was repeated 50 iterations. Firstly, we loaded these 199 independent tests to only one Matlab Distributed Compute Engine (MDCE) worker in the cluster; it took 828.74 seconds to finish. Then we gradually increased the number of workers in the cluster and repeat the calculation of the same job. It took a 16-node cluster 65.63 seconds to complete the same job.

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. The total time consumption for one node to finish the job is 828.74 seconds. Since the total time consumption is large, adding more nodes into the cluster helps to reduce the computational time significantly.

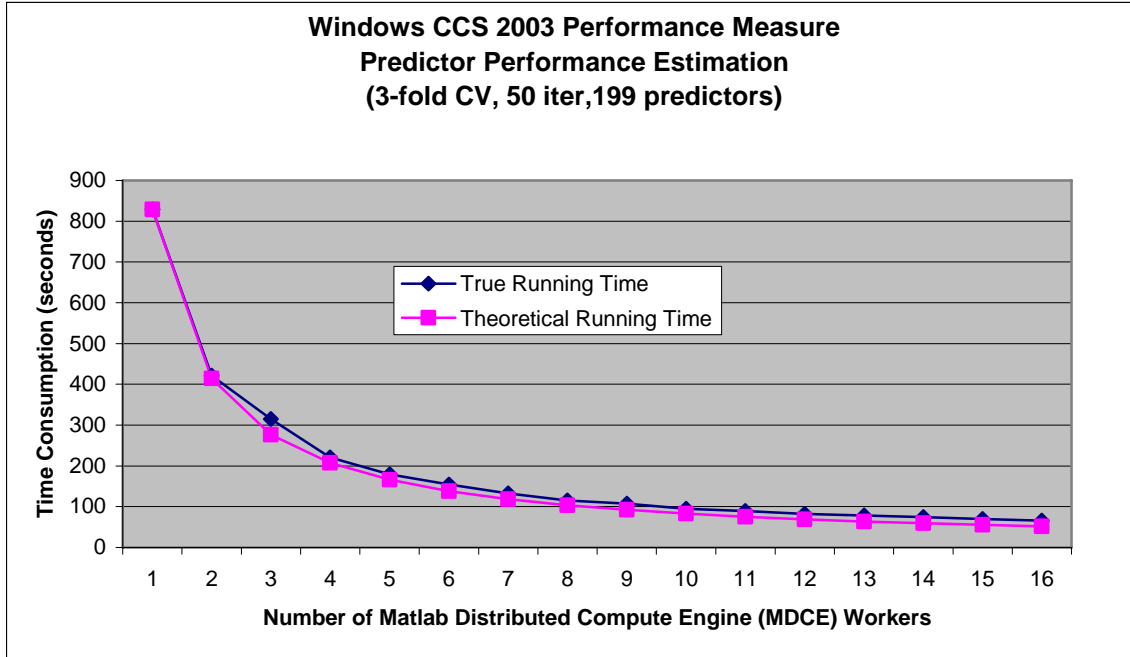


Figure 1. The running time to accomplish a job when the number of MDCE workers increase. Let T be the time consumption when number of MDCE workers equals to one. The theoretical running time is calculated as T/N , where N is the number of MDCE workers involved in computing.

Experiment 2: In the second experiment, we want to show when the number of independent tasks are fixed, by adding more nodes into a cluster, we are able to increase the computational complexity of each task without increase the running time to finish the same job.

In Figure 2, we want to test the prediction performance of 16 independent predictors. When there is only one MDCE worker in the cluster, it took 76.27 seconds to finish all 16 independent tests, where each estimate is based on 50 iterations of 3-fold cross validation. Then we double the number of MDCE workers in the cluster, and it took 76.26 seconds to finish all 16 tests with 100 iterations of 3-fold cross validation in each test. Similarly, we increased the number of MDCE workers to 4, 8, and 16 in the cluster. The total running time is about the same while the number of iterations in each independent test increased to 200, 400, and 800 respectively. Since the prediction performance of each predictor is a statistic based on the average of a number of single test on the same predictor, the more iterations it run, the more reliable the mean estimate we will get. But the computational cost will increase a lot when we increase the number of iterations. A compute cluster will help us to ease the increased computational cost in this situation. In

Figure 2, the area of the circle indicates the number of iterations of each independent task. As we can see from the plot, as the number of MDCE workers increases, we are able to run more iterations for each independent task while the total time consumption keeps about the same.

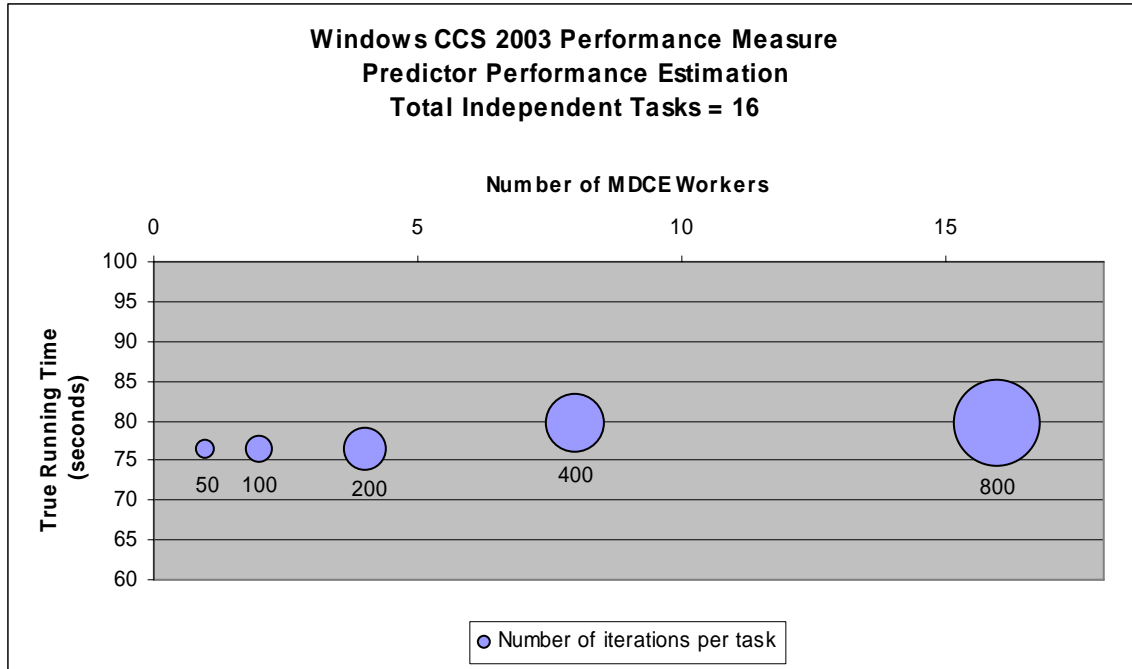


Figure 2. The running time keeps the same to accomplish a job with 16 independent tasks when the complexity of each task increases and the number of MDCE workers increase.

Reference List

't Veer, Laura J., et al. "Gene expression profiling predicts clinical outcome of breast cancer." Nature 415.6871 (2002): 530-36.