Performance Analysis of Parallel Collaborative Filtering on Spark and Hadoop cluster

Kundjanasith Thonglek
High Performance Computing and Network Center
Deparment of Computer Engineering, Kasetsart University
Bangkok, Thailand
kundjanasith.t@ku.th

Kohei Ichikawa
Software Design and Analysis Laboratory
Graduate School of Information Science, NAIST
Nara, Japan
ichikawa@is.naist.jp

***Abstract*—Spark is a cluster computing technology for large-scale data processing that provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. Following the SparkContext can connect to several types of cluster manager which allocate resource across applications. Once connected, Spark acquires executors on nodes in the cluster, which are processes that run computations and store data for the application. Next, it sends application code to the executors. SparkContext represents the connection to a Spark cluster and can be used to create resilient distributed datasets (RDDs). it also accumulators and broadcast variables on the cluster. This research approaches the problem by comparing Spark cluster and Hadoop cluster performance are used the parallel collaborative filtering program?s execution time as criteria to find when the application should execute on Spark or Hadoop. This research is primarily intended for cluster computing benchmark.**

***Index Terms*—Large scale data processing; Resilient Distributed Datasets; Collaborative Filtering; Spark;**

I. Introduction

Big data and advanced analytics technology is not just because the size of data is big but also because the potential of impact is big such as recommendation system, clustering system and the others. They were not possible previously because they were too costly to implement or they were not capable of handling the large volumes of data involved in a timely manner. But in the present technology can make it possible due to the new data management systems that handle a wide variety of data from sensor data to web and social media data improved analytical capabilities including event, predictive and text analytics, faster hardware ranging from faster multi-core processors and large memory space to solid-state drives and tried data storage for handling data. Supporting big data involves combining these technologies to enable new solutions that can bring significant benefits to the business.

Recommendation system is the key to success for discovering and retrieval of content in this era of big data. It have become increasingly popular in recent years, and are utilized in a variety of areas including movies, music, news, books, and product in general. These are also recommendation for experts. Collaborative Filtering (CF) technique is used to create recommendation system, is a technique of making automatic predictions about interest of a user by collecting preference or taste information from many users. The advantages of this technique are not rely on machine analyzable content, it is capable of accurately recommending complex items such as movies movies without requiring an understanding of the item similarity in recommendation system. The technique is based on the assumption that people who agreed in the past will agree in the future, and they will like similar kinds of items as they liked in the past. The algorithm is widely applied to this technique is Alternating Least Square (ALS) algorithm that aim at learning the two types of items as they liked in the past.

Large-scale data processing is the process of applying data analysis technique to a large amount of data. Typically, large-scale data analysis is performed through two popular techniques: parallel database management systems or map reduce powers systems. The parallel database management system requires that the data be in a database management system supported schema, whereas map reduce system supports data in any form. Moreover, the dat extracted or analyzed in large-scale data analysis can be displayed in various different forms such as tables, graphs, figures and statistical analysis that depending on the analysis system. There are two popular open source cluster computing for large-scale data processing based on the map reduce model: Spark framework which is the framework process data by in-memory processing, has Spark ML library to be scalable machine learning library and Hadoop framework which is the framework process data by on-disk processing, has Mahout library to be scalable machine learning library.

II. Literature review

A. Large-scale Parallel Collaborative Filtering for Netflix

Spark ML library and Mahout library are scalable machine learning library use this paper to be the reference for parallel collaborative filtering technique using alternating least square algorithm to implement in the library of both. However, this paper execute the parallel collaborative filtering technique using Matlab library on Linux cluster only.

This research will further advance paper by executing parallel collaborative filtering technique using more efficient framework than using Matlab library on Linux cluster with Spark ML library on Spark cluster and Mahout library on Hadoop cluster.

B. Comparing Apache Spark and MapReduce with Performance Analysis using K-Means

K-means clustering algorithm is the algorithm that simpler than Alternating Least Square algorithm is used to apply with parallel collaborative filtering technique. Spark and Hadoop cluster with 1 node and 2 nodes is too small to find the impact of the number of node on the executing performance.

This research will further advance paper by using the alternating least square algorithm of parallel collaborative technique instead of K-means algorithm because of the complexity. The environment of experimental cluster is extended to 5 worker nodes that can find the impact or relation to the performance with increasing the number of worker nodes.

III. Design and Implementation

There are 2 large-scale data processing framework which we use in the experiment are Spark framework and Hadoop framework. Spark using in-memory processing to process data but Hadoop using on-disk processing.

Alternating Least Square (ALS) algorithm is CPU intensive algorithm more than memory intensive so we design experiment by vary the number of CPU cores per each worker node with fix the size of memory per each worker node. We interest the number of worker nodes effect to the performance or not then we design the experiment by changing the number of worker nodes in the cluster with the same total number of CPU cores and memory size in the cluster.

We find the dataset that suitable to create recommendation system by parallel collaborative filtering technique is dataset from The Netflix Prize that is a large-scale data mining competition held by Netflix for the best recommendation system algorithm for predicting user ratings on movies, based on a training set of more than 100 million ratings given by over 480,000 users to nearly 18,000 movies. Each training data point consist of a quadruple (user, movie, date, rating) where the rating is an integer from 1 to 5, the size of dataset is 2.8 gb. Afterward, we separate the dataset using sampling to 20 datasets by starting from 5 million records then add up 5 million util 100 million records.

Experiment is divided into 2 sections to find the suitable configuration of Spark cluster and Hadoop cluster in the limited resource to executer parallel collaborative filtering technique for create the recommendation system: First section is the number of CPU cores per each worker node effect analysis to observe parallel computing by changing the number of CPU cores per each worker node to 2, 4, 8, 16 with fix the size of memory is 8 gb per each worker node and Second section is the number of worker node effect analysis to observer parallel I/O by changing the number of worker node with the total number of CPU cores is 20 and size of memory is 40 gb.

Implementation is divided into three main parts in this experiment. The first part is installation part, the second part is training data part, the third part is performance analysis part. Installation part is a part for installation and configuration the Spark and Hadoop on cluster. Training data part is a part for training data for train data by Spark ML library on the Spark cluster and Mahout library on Hadoop cluster. Performance analysis part is a part for analyst the execution time of each configuration.

A. Installation Part

Installation Part has 2 sections: First section is the installation part for observe parallel computing effect and second section is the installation part for observe parallel I/O effect.

The configuration in cluster of parallel computing observation:

- Case 1 : 5 worker nodes by each node has CPU 2 cores and memory 8 gb
- Case 2 : 5 worker nodes by each node has CPU 4 cores and memory 8 gb
- Case 3 : 5 worker nodes by each node has CPU 8 cores and memory 8 gb
- Case 4 : 5 worker nodes by each node has CPU 16 cores and memory 8 gb

The configuration in cluster of parallel I/O observation:

- Case 1 : 5 worker nodes by each node has CPU 4 cores and memory 8 gb
- Case 2 : 4 worker nodes by each node has CPU 5 cores and memory 10 gb
- Case 3 : 2 worker nodes by each node has CPU 10 cores and memory 20 gb
- Case 4 : 1 worker node which has CPU 20 cores and memory 40 gb

B. Training data Part

Training data Part has 2 sections: Using Spark ML library on Spark cluster and Mahout library on Hadoop cluster which have the same alternating least square algorithm implementation on both libraries.

The Parallel Alternating-Least-Squares with Weighted-λ-Regularization (ALS-WR) algorithm is parallelized by parallelizing the updates of the user feature matrix and of the movie feature matrix.

C. Performance analysis Part

Performance analysis Part has 3 sections: Recording time information using time command, Calculate the average value of time in second unit due to we give a trial three times per each condition and evaluate the result from the experiment by R programming language.

For the number of CPU cores per each worker node effect analysis to observer parallel computing. There are 2 dependent variables on time variable: the number of CPU cores per each worker node and the size of dataset.

Let $\boldsymbol{t}$ be the execution time (seconds)

Let $\boldsymbol{c}$ be the number of CPU core per worker node (cores)

Let $\boldsymbol{d}$ be the size of data set (MB)

The Multiple linear regression (MLR) model that describes a dependent variable $\boldsymbol{t}$ by independent variables $\boldsymbol{c}$ and $\boldsymbol{d}$ is expressed by the equation as follow where the number α and β are the parameters (1).

For the number of worker nodes effect analysis to observer parallel I/O. There are 2 dependent variables on time variable: the number of worker nodes and the size of dataset.

Let ***t*** be the execution time (seconds)

Let ***n*** be the number of worker node (nodes)

Let ***d*** be the size of data set (MB)

The Multiple linear regression (MLR) model that describes a dependent variable ***t*** by independent variables ***n*** and ***d*** is expressed by the equation as follow where the number α and β are the parameters (2).

$$t = \alpha_1 + \beta_1 c + \beta_2 d \tag{1}$$

$$t = \alpha_2 + \beta_3 n + \beta_4 d \tag{2}$$

IV. Experiment Result

After we give a trial for execute the different size of data sets with the same algorithm but different type of cluster configuration. The experimental result is divided to 2 sections.

First section is the experimental result of executing parallel collaborative filtering using alternating least square algorithm with Spark ML library on Spark cluster and Mahout library on Hadoop cluster by changing the number of CPU cores per worker node in the cluster which has 1 master node and 5 worker nodes for observe the impact from parallel computing on the performance.

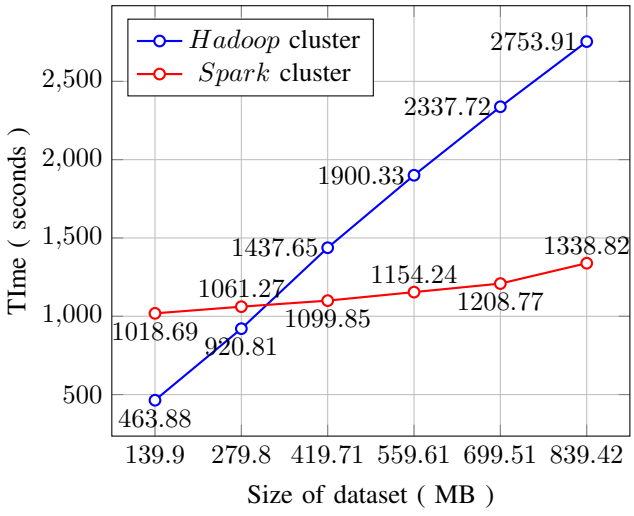

Fig. 1. Graph of testing on 5 worker nodes by each node has CPU 2 cores and memory 8 GB

From Fig 1, we can analyze that the intersection point between Hadoop cluster line and Spark cluster line at the size of dataset is 319.51 MB and the execution time is 1074.05 seconds. As the result, we can execute parallel collaborative filtering on Hadoop cluster faster than Spark cluster when the size of input data is less than 319.51 MB by 5 worker nodes by each node has CPU 2 cores and memory 8 gb.

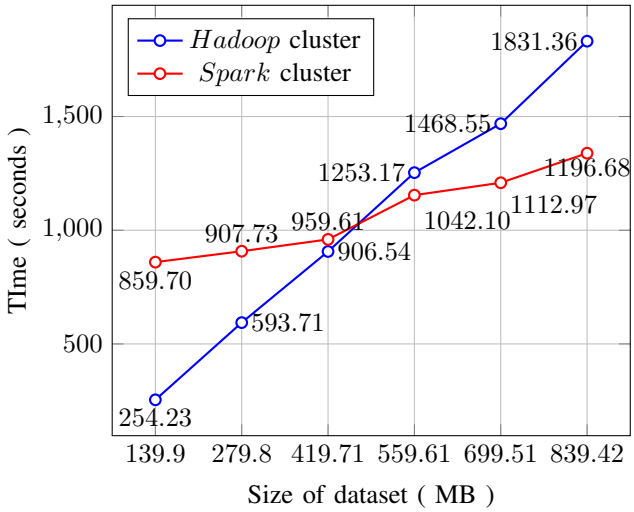

Fig. 2. Graph of testing on 5 worker nodes by each node has CPU 4 cores and memory 8 GB

From Fig 2, we can analyze that the intersection point between Hadoop cluster line and Spark cluster line at the size of dataset is 462.38 MB and the execution time is 990.78 seconds. As the result, we can execute parallel collaborative filtering on Hadoop cluster faster than Spark cluster when the size of input data is less than 462.38 MB by 5 worker nodes by each node has CPU 4 cores and memory 8 gb.

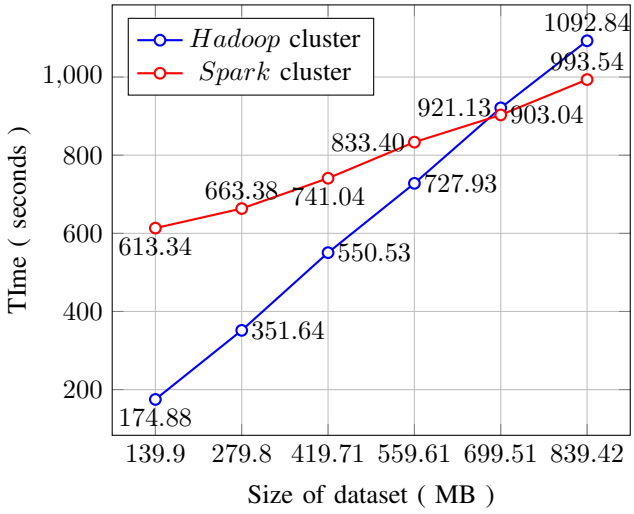

Fig. 3. Graph of testing on 5 worker nodes by each node has CPU 8 cores and memory 8 GB

From Fig 3, we can analyze that the intersection point between Hadoop cluster line and Spark cluster line at the size of dataset is 691.07 MB and the execution time is 902.86 seconds. As the result, we can execute parallel collaborative filtering on Hadoop cluster faster than Spark cluster when the size of input data is less than 691.07 MB by 5 worker nodes by each node has CPU 8 cores and memory 8 gb.

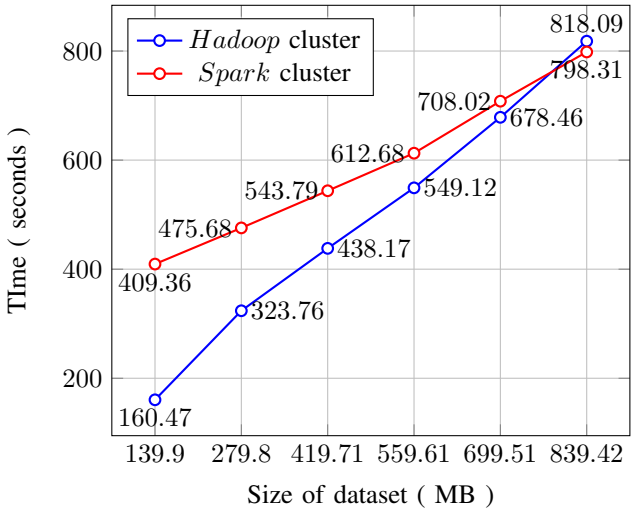

Fig. 4. Graph of testing on 5 worker nodes by each node has CPU 16 cores and memory 8 GB

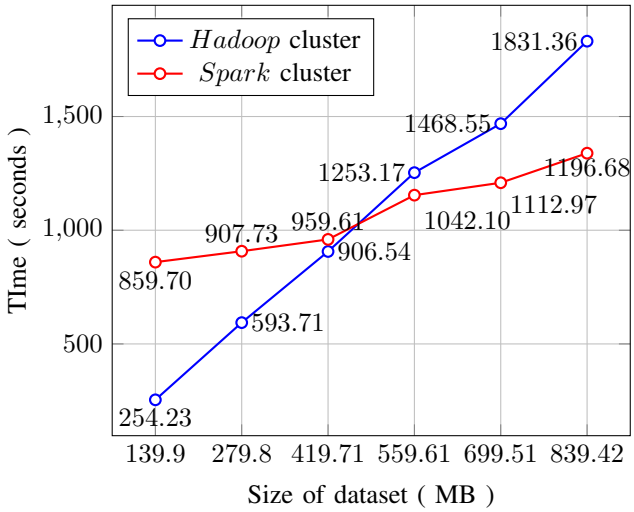

Fig. 5. Graph of testing on 5 worker nodes by each node has CPU 4 cores and memory 8 GB

From Fig 4, we can analyze that the intersection point between Hadoop cluster line and Spark cluster line at the size of dataset is 759.62 MB and the execution time is 740.75 seconds. As the result, we can execute parallel collaborative filtering on Hadoop cluster faster than Spark cluster when the size of input data is less than 759.62 MB by 5 worker nodes by each node has CPU 16 cores and memory 8 gb.

TABLE I
THE GRADIENT OF CLUSTER LINE BY EACH CONFIGURATION THAT OBSERVE PARALLEL COMPUTING

The configuration in cluster	Hadoop cluster	Spark cluster
5 worker nodes by each node has CPU 2 cores and memory 8 gb	3.27	0.46
5 worker nodes by each node has CPU 4 cores and memory 8 gb	2.25	0.48
5 worker nodes by each node has CPU 8 cores and memory 8 gb	1.31	0.54
5 worker nodes by each node has CPU 16 cores and memory 8 gb	0.94	0.56

After execute the multiple linear regression algorithm as the equation (1), we discover that the coefficient of size of dataset is 0.5056 on Spark cluster and 1.938 on Hadoop cluster, the coefficient of the number of CPU cores per each worker node is -38.5424 on Spark cluster and -70.894 on Hadoop cluster, the interception is 927.1458 on Spark cluster and 537.263 on Hadoop cluster.

Second section is the experimental result of executing parallel collaborative filtering using alternating least square algorithm with Spark ML library on Spark cluster and Mahout on Hadoop cluster by changing the number of worker in the cluster nodes which has the total number of CPU cores is 20 cores and the total number of memory is 40 GB for observe the impact from parallel I/O on the performance.

From Fig 5, we can analyze that the intersection point between Hadoop cluster line and Spark cluster line at the size of dataset is 462.38 MB and the execution time is 990.78 seconds. As the result, we can execute parallel collaborative filtering on Hadoop cluster faster than Spark cluster when the size of input data is less than 462.38 MB by 5 worker nodes by each node has CPU 4 cores and memory 8 gb.

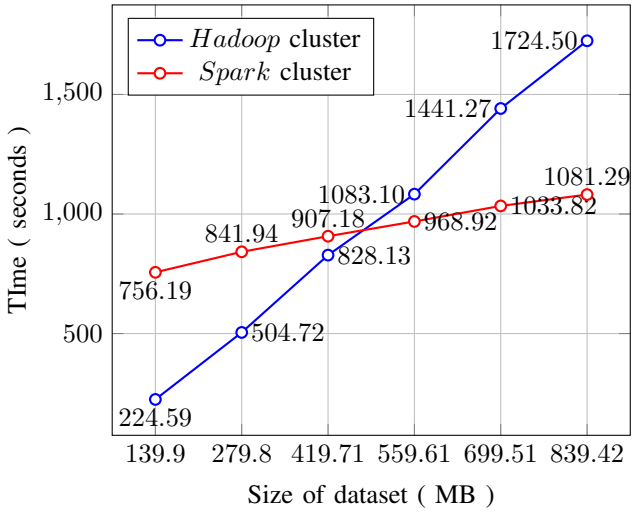

Fig. 6. Graph of testing on 4 worker nodes by each node has CPU 5 cores and memory 10 GB

From Fig 6, we can analyze that the intersection point between Hadoop cluster line and Spark cluster line at the size of dataset is 468.33 MB and the execution time is 921.70 seconds. As the result, we can execute parallel collaborative filtering on Hadoop cluster faster than Spark cluster when the size of input data is less than 468.33 MB by 4 worker nodes by each node has CPU 5 cores and memory 10 gb.

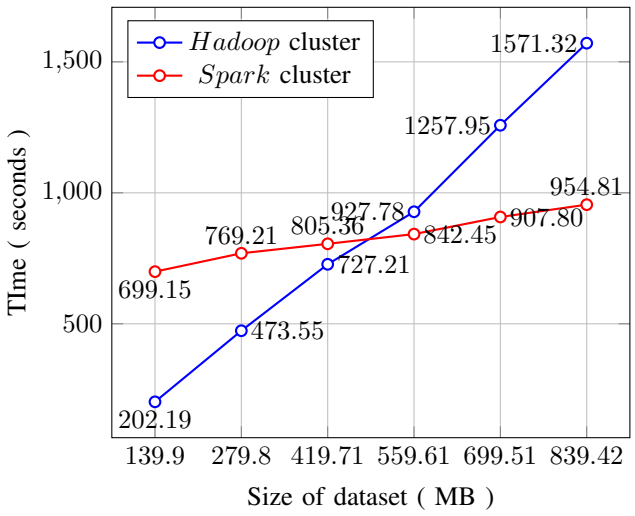

Fig. 7. Graph of testing on 4 worker nodes by each node has CPU 5 cores and memory 10 GB

From Fig 7, we can analyze that the intersection point between Hadoop cluster line and Spark cluster line at the size of dataset is 470.37 MB and the execution time is 822.98 seconds. As the result, we can execute parallel collaborative filtering on Hadoop cluster faster than Spark cluster when the size of input data is less than 470.37 MB by 2 worker nodes by each node has CPU 10 cores and memory 20 gb.

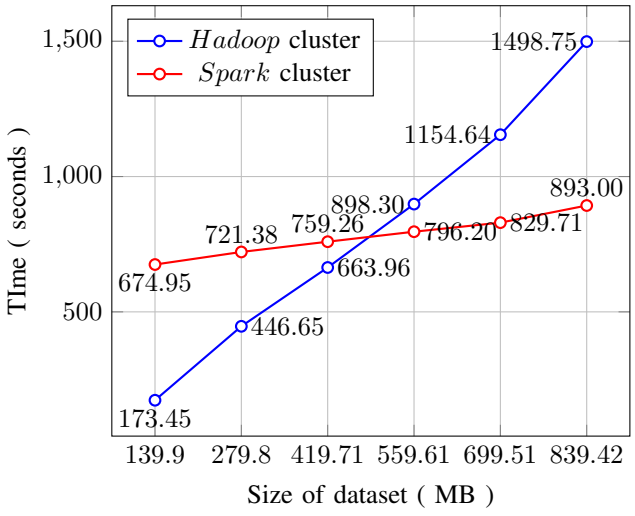

Fig. 8. Graph of testing on 4 worker nodes by each node has CPU 5 cores and memory 10 GB

From Fig 8, we can analyze that the intersection point between Hadoop cluster line and Spark cluster line at the size of dataset is 472.19 MB and the execution time is 773.9 seconds. As the result, we can execute parallel collaborative filtering on Hadoop cluster faster than Spark cluster when the size of input data is less than 472.19 MB by 1 worker node which has CPU 20 cores and memory 40 gb.

TABLE II
THE GRADIENT OF CLUSTER LINE BY EACH CONFIGURATION THAT OBSERVE PARALLEL I/O

The configuration in cluster	Hadoop cluster	Spark cluster
5 worker nodes by each node has CPU 4 cores and memory 8 gb	2.25	0.48
4 worker nodes by each node has CPU 5 cores and memory 10 gb	2.14	0.46
2 worker nodes by each node has CPU 10 cores and memory 20 gb	1.96	0.37
1 worker node which has CPU 20 cores and memory 40 gb	1.89	0.31

After execute the multiple linear regression algorithm as the equation (2), we discover that the coefficient of size of dataset is 0.3997 on Spark cluster and 2.032 on Hadoop cluster, the coefficient of the number of worker nodes is 56.9851 on Spark cluster and 59.832 on Hadoop cluster, the interception is 521.7023 on Spark cluster and -253.391 on Hadoop cluster.

TABLE III
THE VALUE OF COEFFICIENT IN MULTIPLE LINEAR REGRESSION

The kind of cluster	α_1	α_2	β_1	β_2	β_3	$\beta 4$
Hadoop cluster	537.26	-253.39	-70.89	1.94	59.83	2.03
Spark cluster	927.15	521.70	-38.54	0.51	56.99	0.40

The Multiple Linear Regression (MLR) equation

- Spark cluster :

$$t = 927.15 - 38.54c + 0.51d \quad (3.1)$$

$$t = 521.70 + 56.99n + 0.40d \quad (4.1)$$

- Hadoop cluster :

$$t = 537.26 - 70.89c + 1.94d \quad (3.2)$$

$$t = -253.39 + 59.83n + 2.03d \quad (4.2)$$

V. DISCUSSION

The purpose of this research was to find the suitable configuration of Spark cluster and Hadoop cluster to execute parallel collaborative filtering technique for create recommendation system and discover the impact from parallel computing and parallel I/O on the performance.

To observe the parallel computing impact on performance by changing the number of CPU cores per each worker node with fix the size of memory.

To observe the parallel I/O impact on performance by changing the number of worker nodes with the same total number of CPU cores and size of memory.

VI. CONCLUSION

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.