DYNAMIC DATA REPLICATION USING FDDRA

[1]S.Sasikala Devi, [2]Dr.Antony Selvadoss Thanamani
[1]Research Sccholar, [2]Associate and Head, Department of Computer Science,
[1]Manonmaniam Sundaranar University, Thirunelveli, Tamilnadu, India
[2]NGM College of Arts and Science, Tamilnadu.

Abstract : Data grids have emerged as a useful technology for managing large amounts of distributed data in many fields like scientific experiments and engineering applications. In this regard, replication in data grids is an efficient technique that aims to improve response time, reduce the bandwidth consumption and maintain reliability. Unfortunately, most of existing replication strategies consider a single file based granularity and neglect correlations among different data files. However, the analysis of many real data intensive applications reveals that jobs and applications request groups of correlated files. In this paper, we propose a new data replication strategy which is an enhanced algorithm Modified Bandwidth Hierarchy Replication Algorithm. The data in this work is read-only and so there are no consistency issues involved. The evaluation metrics we analyze in the experiments are mean scalability, stress, average connectivity, eccentricity, number of replications. Using the MATLAB tool, extensive experimentations show that our proposed strategy has better performance in comparison to other strategies under most of access patterns.

IndexTerms **- Data Grid; Replication; File Correlation; Data Mining; Mining Grid Data.**

I. INTRODUCTION

Nowadays, huge volume of data is being collected by many scientific, engineering and other applications such as Particle Physics, High Energy Physics and Genetics, to quote but a few. These applications are ever more demanding in terms of their computing requirements as well as their storage requirement. In this context, data grids emerge as a suitable solution for these applications. Data Grid is an integrating architecture that allows connecting a collection of hundreds of geographically distributed computers and storage resources located in different parts of the world to facilitate the sharing of data and resources. Data grid represents a set of connected sites. Each site locally executes jobs that were scheduled to it by a given resource broker. A job may require data which can be found in the site where it is executed as it may be found in other sites. In this regard, effective data management is one critical issue in data grid systems and involves many challenges. One way to effectively cope with these challenges is to rely on the replication technique. The main idea of replication is to create multiple copies of the same data (aka replicas) in several storage resources. Hence, data availability is increased since data are stored at more than one site. In contrast, if data are not replicated in grid sites, all data requests issued from grid users or grid jobs must wait at a single node. The grid system is then challenged by many problems such as the increased risk of failures, the overloading of popular sites, and access latency. Indeed data replication helps to address these problems by ensuring considerable benefits. Mainly, it offers data availability by increasing the probability that there is an operational (i.e., not failed) grid site that has a copy of the data when a request is made. Grid system reliability is therefore significantly improved since when more replicas are created; the chance that requests of users will be serviced properly is increased. In addition, it provides the opportunity to share the load generated by user requests among the different grid sites that have replicas. Also, data replication improves response time and bandwidth consumption.

II. RELATED WORK

PDDRA is based on an assumption: members in a VO (Virtual Organization) have similar interests in files. Based on this assumption and also file access history, PDDRA predicts future needs of grid sites and pre-fetches a sequence of files to the requester grid site, so the next time that this site needs a file, it will be locally available. PDDRA consists of three phases: storing file access patterns, requesting a file, performing replication, prefetching and replacement. The simulation results show that PDDRA has better performance in comparison with other algorithms in terms of job execution time, effective network usage, and total number of replications, stress and percentage of storage filled.

By considering spatial locality, PHFS uses predictive techniques to predict the future usage of files and then pre replicates them in hierarchal manner on a path from source to client in order to increase locality in access. The correlation between files is inferred from previous access patterns by association rules and clustering techniques of data mining. PHFS operates in three steps. In the first phase, a software agent in the root node collects the file access information from all the clients in the system and puts them in a log file. In the second phase, data mining techniques are applied on log files with the aim to find strongly related files that are accessed together. Finally, whenever a client requests a file, PHFS finds the predictive working set (PWS) of that file. The PWS is composed by the most related files to a requested file or in other words the predicted subsequent requests after a requested file. Then, PHFS replicates all members of PWS along with the requested file on the path from the source to the client. Let us notice that in this algorithm, the

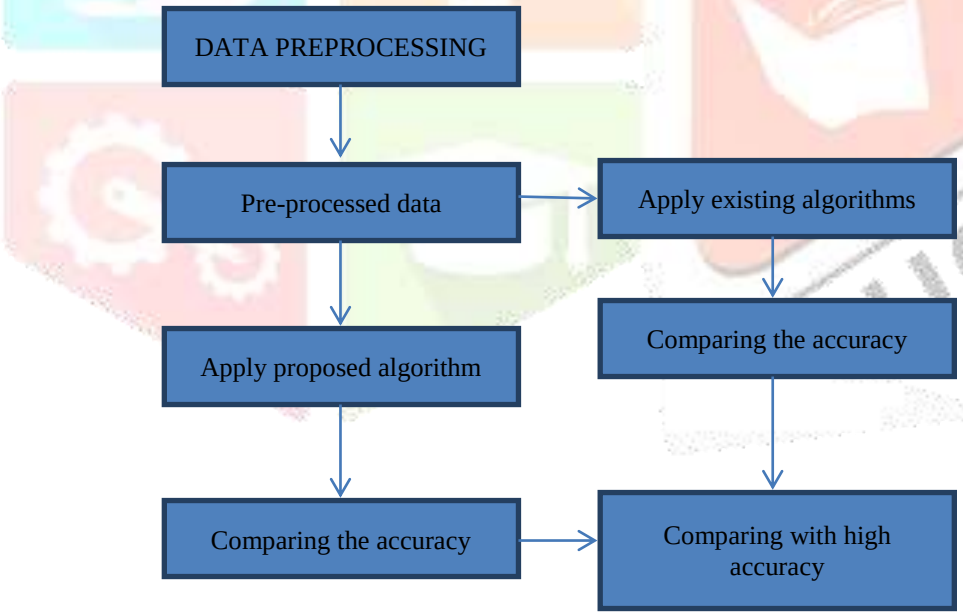
authors did not describe how to mine the file correlations. However, they assumed that correlations are found by data mining techniques such as association rules and clustering without going into detail. In addition, they did not do experiments to compare the proposed algorithm with other methods. They just used instances to compare the access latency with the Fast Spread strategy.

The major idea of the algorithm is to pre-fetch frequently accessed files and their associated files to the location near the access site. It finds out the correlation between the data files through data access number and data access serial. It is based on support, confidence and access numbers. BSCA has two sub algorithms: data mining algorithm and replication algorithm. The data mining algorithm applied to identify frequent files is FPGrowth. Moreover, support and confidence of association rules between these frequent files are computed. If the support and the confidence values between files exceed respective minimum thresholds, frequent files and their associated ones are replicated. The replication algorithm, which is applied in a decentralized manner in each node, sorts the file serial and finds out all files whose access numbers are greater than a threshold. Then, the algorithm constructs sets of related files and replicate files that do not exist in nodes. If free space is lacking, the algorithm deletes weakly correlated files. If the storage space is still insufficient, files whose access number is less than a threshold will be deleted. The simulation performed using the OptorSim simulator shows that BSCA outperform SBU, ABU and Fast Spread replication strategies by giving the lowest average response time.

A new replication algorithm named Modified BHR was proposed. The proposed algorithm was based on the network level locality. The algorithm tries to replicate files within a region and stores the replica in a site where the file has been accessed frequently based on the assumption that it may require in the future. This algorithm increases the data availability by replicating files within the region to the region header and also storing them in the site where the file has been accessed frequently. It also reduces unnecessary replication. Instead of storing files in many sites, they can be stored in a particular site so that the storage usage can be reduced.

A new algorithm for automatic replication for Grid environment was proposed. Automatic replication is a complex task, which requires a set of algorithms including creation, removal, selection and coherency of replicas as well as replica update propagation algorithm. The replica creation algorithm is responsible for automatic creation of new replicas. The replica removal algorithm is responsible for replicas removal intended to save storage space. The replica selection algorithm is responsible for optimal replica selection for the specific read/write operation. Finally, the replica update propagation algorithm is responsible for updating out of date replicas. The proposed algorithm was tested for two types of grids: Clusterix and SGIgrid. The results indicate that the automatic replication can decrease total data access time and increase storage usage.

III. FRAMEWORK



IV RESEARCH METHODOLOGY

It consist of methods and tools used to accomplish the methodology, as per the framework process, the fourth process is using our algorithm for accomplish a result with a high accuracy. There are three major steps are followed in this section. First one is replication process, second one is data mining based optimization and last one is mathematical process model. The replication process is used to help replicate the files as most frequently accessed and also removes the most duplicate replicas.

Before that , the major process was accomplished, that is pre-processing the data. There are four major steps was processed in the pre-processing the data. That is,

1. Data reduction 2. Data integration 3. Data cleaning 4. Data transformation

4.1 USING EFFICIENT PROPOSED ALGORITHM

4.1.1 Algorithm description

The FDDRA (Fuzzy based dynamic replication algorithm) is based on the network level locality. The enhanced algorithm tries to replicate the files within a region and stores a replica in a site where the files has been frequently accessed based on the assumption that it may require in the future.

- According to this algorithm a group of sites are located on the same network region. A network region is a network topological space where sites are closely located.
- Initially all data are produced in the region the job completion is found within the region the job completion is fast.
- Then data are produced in the master site and distributed in region header.
- The region headermaintains and produces the data to the sub regions.
- Access frequency of all files in distributed and replicas of popular files are stored in sites where they are accessed for maximum time. With the consideration of geographical and temporal locality

Inputs: Grid Topology, Bandwidth, Data and Storage Space

Output: scalability, stress, average connectivity, eccentricity, number of replications.

Method:

1. Submit Data to Grid

2. Every request sends to Master of Sites and Regional Servers

3. Data are produced in Master sites

4. Master node has the global view of all regions in the grid

5. Initially Do { Send request to head nodes asking for [available bandwidth, load gauge] } End;

6. Find maximum available bandwidth and load gauge

7. Schedule job to region

8. Files replicated to region header (RH)

9. RH maintains sites and replica details

10. Jobs are scheduled to grid sites

11. Call Replica Optimizer for to get best replica

4.1.2 Processing of algorithm:

In this algorithm considered a two layered hierarchical structure for dynamic replicating file and scheduling in data grids was proposed. To achieve good network bandwidth utilization and reduce data access time, it considers not only computational capability, job type and data location but also it considers cluster information in job placement decision.

The proposed FDDRA (Fuzzy based dynamic replication algorithm)also based on the network level locality. The enhanced algorithm tried to replicate files with the region and also the neighbouring region and store replica in a site where the files has been accessed frequently based on the assumption that it may require in the future.

Dynamic replication is an optimization technique which aims to increase network bandwidth and availability of data and reduce total access time by considering different issues. The abovementioned issues have been addressed in the proposed algorithm, that is, FDDR Algorithm using agents that needs to be addressed before replicating.

4.1.3 Data mining based optimization:

In this approach, Apriori algorithm is implemented in MATLAB. Implementing Apriori algorithm in MATLAB reduces execution time and scans the database only once. In this approach, the file is created in which the transactions on which the Apriori algorithm is to be applied are stored. The file created is stored in MATLAB in matrix form. Then the Apriori algorithm is applied on matrix.

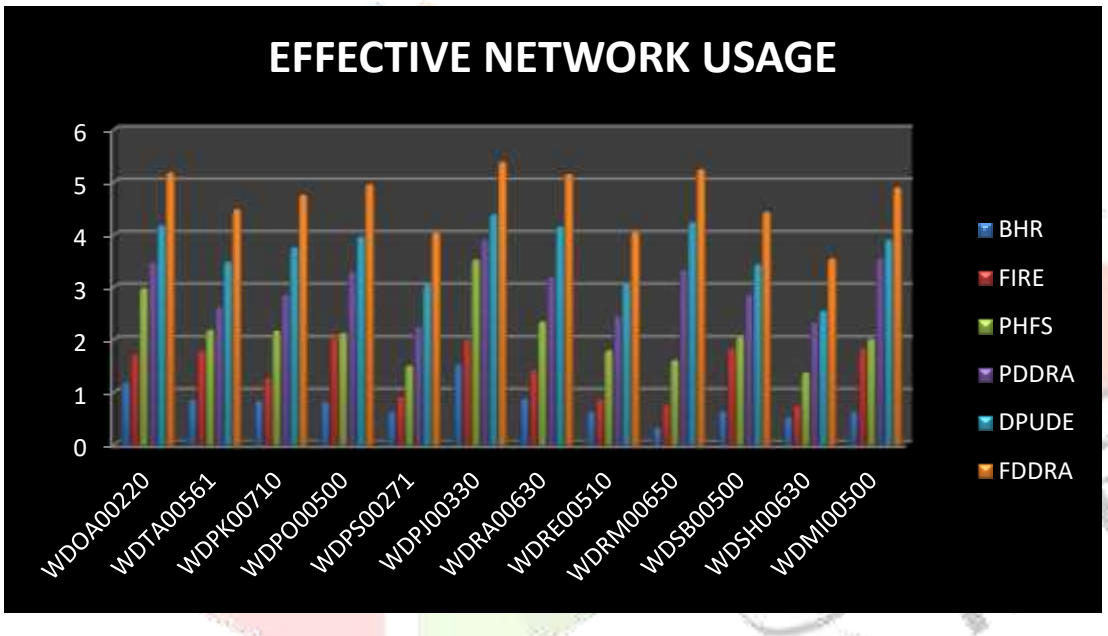
4.1.4 General Process

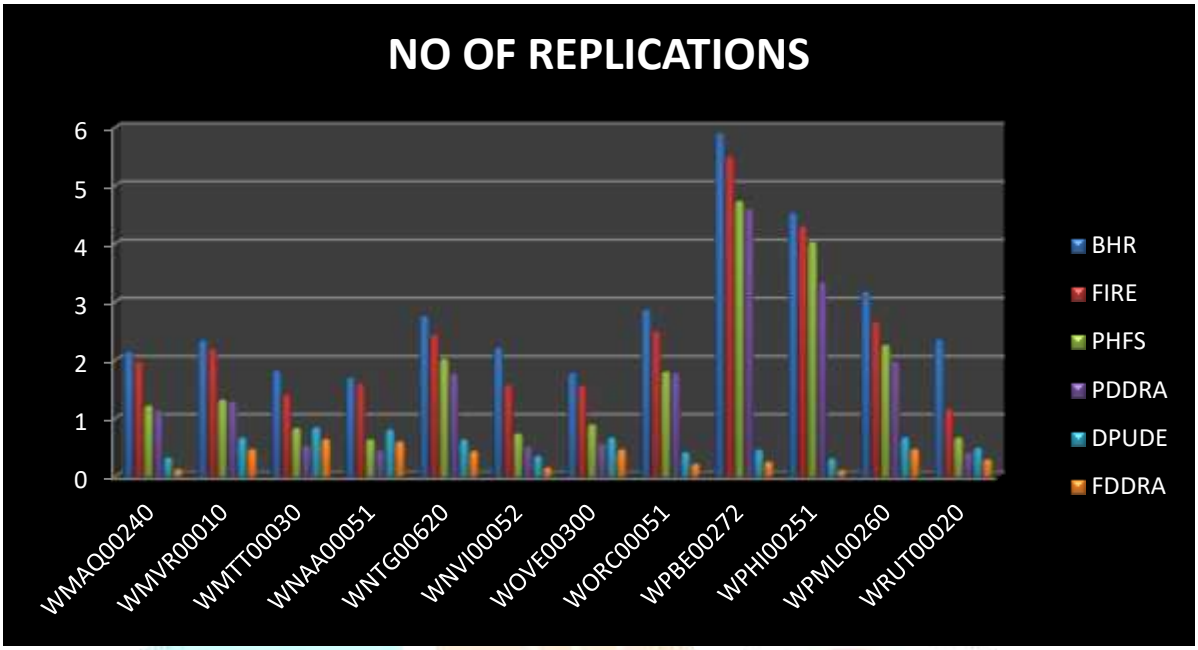
Association rule generation is usually split up into two separate steps:

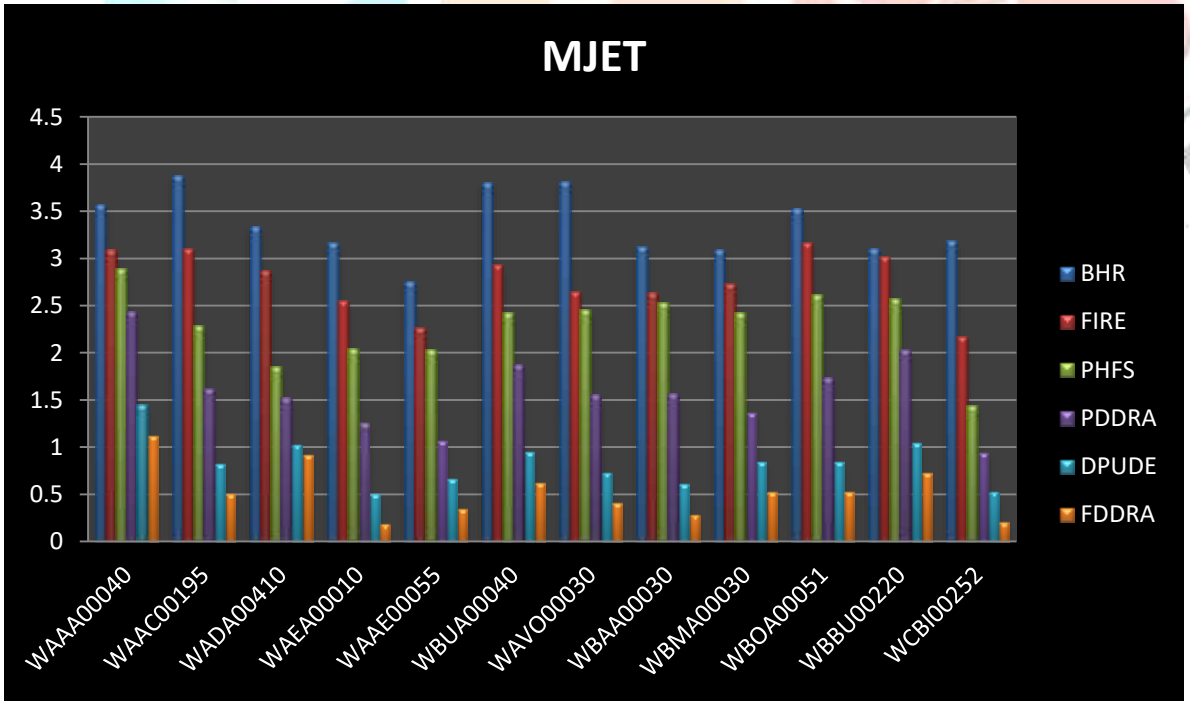
1. First, minimum support is applied to find all frequent item-sets in a database.
2. Second, these frequent item-sets and the minimum confidence constraint are used to form rules.

V. RESULTS AND DISCUSSION

The log path of the Data Grid , in which average of 12 logs over random period of time is taken and calculated for the shortest path, comparing BHR, FIRE, PHFS, PDDRA, DPUDE and FDDRA (Fuzzy based dynamic replication algorithm). The shortest path of FDDRA (Fuzzy based dynamic replication algorithm) proves that it is highly optimal when comparing BSCA, MBHR, ARRA and BHR. The overall result in this experiment infers that FDDRA (Fuzzy based dynamic replication algorithm)algorithm is better.




NO OF REPLICATIONS
6
5
4
3
2
1
0
WMAQ00240
WMVR00010
WMTT00030
WNAA00051
WNTG00620
WNVI00052
WOVE00300
WORC00051
WPBE00272
WPHI00251
WPML00260
WRUT00020
BHR
FIRE
PHFS
PDDRA
DPUDE
FDDRA


MJET
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
WAAA00040
WAAC00195
WADA00410
WAEA00010
WAAE00055
WBUA00040
WAVO00030
WBAA00030
WBMA00030
WBOA00051
WBBU00220
WCBI00252
BHR
FIRE
PHFS
PDDRA
DPUDE
FDDRA

VI. CONCLUSION

In this research work, a new replication strategy based is presented, based on Fuzzy based dynamic replication algorithm. As the demand for automated analysis of large and distributed data grows, new data mining challenges in distributed computing environments emerge. It is mainly addressed with the key problems related to the impact of the replicas distribution within sites on the performances of replication strategies. An overview of the parameters used in replication strategies as well as a survey of the main evaluation metrics is clearly proposed. These metrics are mainly quantitative. A new criterion is proposed having a direct impact on the results of quantitative evaluations, namely Number of replications, scalability, stress, average connectivity, eccentricity. To prove the efficiency of FDDRA (Fuzzy based dynamic replication algorithm)and its distribution affects the evaluation results of replication strategies and analysis of these results with existing algorithms like BHR, FIRE, PHFS, PDDRA, DPUDE.

REFERENCES

[1] I. Foster, C. Kesselman and S. Tuecke. 2001. The Anatomy of the Grid: Enabling Scalable Virtual Organizations, IJSA.
[2] KavithaRanganathan and Ian Foster.: Identifying Dynamic Replication Strategies for a High Performance Data Grid. International Workshop on Grid Computing, Denver, November 2001.
[3] William H. Bell, David G. Cameron, Ruben Carvajal-Schiaffino, A. Paul Millar, Kurt Stockinger, and FlorianoZini.: Evaluation of an Economy-Based File Replication Strategy for a Data Grid. In International Workshop on Agent based Cluster and Grid Computing at CCGrid 2003, Tokyo, Japan, May 2003. IEEE Computer Society Press.
[4] Mark Carman, FlorianoZini, Luciano Serafini, and Kurt Stockinger.: Towards an Economy-Based Optimisation of File Access and Replication on a Data Grid. In International Workshop on Agent based Cluster and Grid Computing at International Symposium on Cluster Computing and the Grid (CCGrid'2002), Berlin, Germany, May 2002. IEEE Computer Society Press.
[5] M. Tang, B.S. Lee, C.K. Yeo, X. Tang, Dynamic replication algorithms for the multi-tier data grid, Future Generation Computer Systems 21 (5) (2005), pp. 775– 790.
[6] Y. Yuan, Y. Wu, G. Yang, F. Yu, Dynamic data replication based on local optimization principle in data grid, (2007).
[7] A. Abdullah, M. Othman, H. Ibrahim, M.N. Sulaiman, A.T. Othman, Decentralized replication strategies for P2P based scientific data grid, in: Information Technology, ITSim, International Symposium on, (2008), pp. 1–8.
[8] Y. Ding, Y. Lu, Automatic data placement and replication in grids, in: High Performance Computing, HiPC, International Conference on, (2009), pp. 30– 39.
[9] Neeraj Nehra, R.B.Patel, V.K.Bhat, Distributed Parallel Resource Co-Allocation with Load Balancing in Grid Computing, IJCSNS International Journal of Computer Science and Network Security, January (2007), pp. 282-291.
[10] A. Horri, R. Sepahvand, Gh. Dastghaibyfard, A Hierarchical Scheduling and Replication Strategy, IJCSNS International Journal of Computer Science and Network 30 Security, August (2008), pp. 30-35.
[11] S. M. Park, J. H. Kim, Y. B. Ko, W. S. Yoon, "Dynamic Data Replication Strategy Based on Internet Hierarchy BHR", in: Lecture notes in Computer Science Publisher, 2004, pp. 838-846.
[12] K. Sashi, A.S. Thanamani, Dynamic replication in a data grid using a modified BHR region based algorithm, Future Generation Computer Systems 27 (2), (2011), pp. 202–210.
[13] Q. Rasool, J. Li, S. Zhang, Replica placement in multi-tier data grid, in: 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, (2009), pp. 103–108.
[14] Y.F. Lin, J.J. Wu, P. Liu, A list-based strategy for optimal replica placement in data grid systems, in: 37th International Conference on Parallel Processing, (2008), pp. 198–205.
[15] D. G. Cameron, R. C. Schiaffino, J. Ferguson et al., "OptorSim v2.0 installation and user guide,"2004,http://read.pudn.com/downloads74/doc/fileformat/270400/Optorsim%20v2.0%20Installation%20and%20User%20Guide.pdf.
[16] S.-M. Park, J.-H. Kim, Y.-B. Ko, and W.-S. Yoon, "Dynamic data grid replication strategy based on Internet hierarchy," in Grid and Cooperative Computing, vol. 3033 of Lecture Notes in Computer Science, pp. 838–846, Springer, Berlin, Germany, 2004. View at Google Scholar.

[17] B. A. Forouzan, TCP/IP Protocol Suite, Tata McGRAW-Hill, Noida, India, 3rd edition, 2006.
[18] The European Data Grid Project, http://www.gridpp.ac.uk/papers/chep04_optorsim.pdf.
[19] D. G. Cameron, R. C. Schiaffino, J. Ferguson et al., "OptorSim v2.0 installation and user guide,".