Evaluation through Realistic Simulations of File Replication Strategies for Large Heterogeneous Distributed Systems

Anchen Chai[1,2], Sorina Camarasu-Pop[1], Tristan Glatard[4]
Hugues Benoit-Cattin[1], and Frédéric Suter[2,3]

[1] Université de Lyon, CREATIS
CNRS UMR5220, Inserm U1044, INSA-Lyon, Université Lyon 1, Lyon, France
[2] IN2P3 Computing Center, CNRS, Lyon-Villeurbanne, France
[3] Inria, Lyon, France
[4] Department of Computer Science and Software Engineering
Concordia University, Montreal, Canada

Abstract. File replication is widely used to reduce file transfer times and improve data availability in large distributed systems. Replication techniques are often evaluated through simulations, however, most simulation platform models are oversimplified, which questions the applicability of the findings to real systems. In this paper, we investigate how platform models influence the performance of file replication strategies on large heterogeneous distributed systems, based on common existing techniques such as prestaging and dynamic replication. The novelty of our study resides in our evaluation using a realistic simulator. We consider two platform models: a simple hierarchical model and a detailed model built from execution traces. Our results show that conclusions depend on the modeling of the platform and its capacity to capture the characteristics of the targeted production infrastructure. We also derive recommendations for the implementation of an optimized data management strategy in a scientific gateway for medical image analysis.

1 Introduction

File replication to multiple storage resources is a common technique to optimize data management in distributed systems. It reduces file transfer bottlenecks and increases file availability, with great impact on the application execution time [13]. Numerous file replication strategies were proposed and evaluated using simulations [1,9,14,16,20,21], focusing mostly on the optimization of file transfer durations (average or total duration by job). However, platform models are often oversimplified, leading to questionable accuracy of simulated transfer duration.

Two platform models are commonly used in the literature. The *homogeneous* model [2,12] uses a nominal bandwidth (e.g., 1 Gb/s) for all the network links between storage and compute resources. The *hierarchical* model [8,17] uses different theoretical bandwidths for different link categories: for instance, 1 Gb/s for local links between computing resources and their local storage resource; 100 Mb/s for national links (compute and storage resources in the same country); 10 Mb/s for inter-country links. While these models might be good approximations for large distributed systems at a coarse level, the limited number of bandwidth values can hardly capture the heterogeneity and the complexity intrinsic to real production systems. In a previous work focusing on simulation accuracy [4], we have shown that the quality of simulated file transfer duration strongly depends on the accuracy of the platform topology and on the parametrization of the simulator. In particular, the homogeneous model can hardly capture the characteristics of a large grid infrastructure and, consequently, the accuracy of the simulation is rather poor when using such a model.

In this paper, we use two different platform models to evaluate file replication strategies: (i) a three-tier hierarchical model, representing the state-of-the-art platform and (ii) a model built from real execution traces. We focus on file management in the EGI e-Infrastructure (http://egi.eu), a large distributed system with hundreds of sites spread world-wide, and in particular on applications executed by the Virtual Imaging Platform (VIP) [11], a Web portal for medical image analysis and simulation. We aim at answering the following questions:

- What is the impact of replication strategies on file transfer durations?
- Does the answer to the above question depend on the platform model?
- What would be reliable recommendations for data placement in VIP?

The remainder of this paper is organized as follows. Section 2 provides some technical background on data replication strategies in general and, more particularly, in EGI and VIP. Section 3 describes our simulation studies with focuses on platform models, studied data placement strategies, and simulation scenarios. Section 4 presents the evaluation and the analysis of the simulation results. Recommendations for the targeted production system are given in Sect. 5. Finally, Section 6 summarizes our findings and details of our future work.

2 Technical Background on File Replication

Replication management encompasses both replica creation and replica selection. The former decides where and how many times to replicate a file, while the latter defines how to choose the best replica for a given file transfer. Both components can be implemented in various ways, depending on the features to optimize, e.g., file availability, transfer time, or network usage.

Replica creation strategies can be classified in two categories: *static* and *dynamic*. In static replication, decisions are made before launching the application and not changed during the execution. In [6,15], authors demonstrated that asynchronously replicating data to several remote sites before the application execution can significantly reduce its execution time. This process is named *file prestaging*. Static replication strategies are usually simple to implement, however, they are often inefficient in a dynamic environment such as a large grid infrastructure. In dynamic replication, decisions can adapt to changes of the infrastructure characteristics, e.g., storage capacity or network bandwidth. More replicas can be created on new nodes during the execution of the application and can be deleted when they are no longer required. Dynamic replication strategies often rely on information obtained at runtime, hence adding an extra overhead to the application execution time.

The Unified Middleware Distribution [7] is an integrated set of software components packaged for deployment as production services on EGI. Among them, the data management services allow users to upload files onto a Storage Element (SE), then replicate and register them in a File Catalog. However, the decisions about where to replicate files and how many replicas to create are left to the applications (users). The replica selection algorithm of the middleware selects replicas according to their distance to the computing site, that is, first in the SE local to the computing site, then in the same country as the job execution, and in last resort, randomly among all available replicas.

The replica creation strategy implemented in VIP relies on the experience and *a priori* knowledge of its administrators. VIP files are automatically replicated to a static predefined list of 3 SEs chosen among the ones considered as stable, with a general good network connectivity, and sufficiently large amounts of available storage space (generally at least 500 GB). This list is updated when one of the SEs needs to be replaced, is in downtime, is full, or faces any other issue preventing its usage. The number of replicas may also vary depending on the type and size of the files. Files larger more than 500MB are usually replicated on the most available SEs.

3 Simulation Studies

The long-term objective of this study is to optimize data placement for scientific gateways such as VIP using large scale distributed heterogeneous infrastructures such as EGI. To this end, we propose to evaluate different simulation scenarios fed with realistic information coming from execution traces. We developed a simulator [18] based on the SimGrid toolkit [3] that is as close as possible to the actual behavior of several VIP services. Hereafter we detail the different components of these simulation scenarios.

3.1 Platform Models

We consider two platform models. First, we extend the realistic trace-based model proposed in [4]. This model determines an average bandwidth value for each network link between a SE and a computing site from file transfer logs of several application executions. This has been shown to give the best accuracy when simulating file transfers. However, some links were not used, and thus not in the logs, while they are needed to conduct the current study.

A naive solution to this issue would be to use the median of all the measured bandwidths for the missing links. However, this would neither reflect the hierarchical topology of the platform nor the overall connectivity of a site concerned by missing link(s). To address this limitation, we first define three categories of network links (local, national, and inter-country) to reflect the topology. For each category c we estimate the connectivity of a site S_i as the ratio between the median bandwidth of the known links to/from S_i and the median bandwidth of all the links: $\widetilde{B_i^c}/\widetilde{B^c}$. We weight this ratio by $|L_i^c|/|L_i|$, since the larger the number of known links, $|L|$, the more reliable the estimation. The overall connectivity of S_i with regard to the rest of the platform is then estimated by the following weighted sum:

$$C_i = \sum_c \left(\frac{|L_i^c|}{|L_i|} . \frac{\widetilde{B_i^c}}{\widetilde{B^c}} \right) . \qquad (1)$$

Finally, the bandwidth of a missing link of category c to/from S_i is computed as the median bandwidth in this category times the overall connectivity: $\widetilde{B^c} \times C_i$.

While this traced-based model is accurate, it is also complex to build. Therefore, we also consider a simpler model inspired from the state-of-the-art hierarchical model. If simulation results are consistent between the two models, then the building simplicity of this three-level hierarchical platform makes it a good candidate for further studies. To better reflect the connectivity of the production system, we enhance it by using average bandwidth values derived from logs instead of the theoretical values proposed in the literature. We use 1.3 Gb/s for local links, 255Mb/s for national links, and 100 Mb/s for inter-country links.

3.2 Replication Strategies

We study data placement strategies based on (i) file prestaging and (ii) a dynamic replication strategy. In the file prestaging strategy, files are copied on three preselected SEs before the execution of the application. This corresponds to the current replication strategy used by VIP. We evaluate the impact of different prestaging lists on the performance of file transfers, with or without *a priori* information on the sites where jobs are executed.

Given the large scale of distributed systems such as EGI, allowing thousands of independent jobs to be executed in parallel, we believe that dynamic replication could further improve data placement during the execution of an application. Our idea is inspired by the "cache hit" mechanism. The first job executed in a computing site downloads the file, then copies and registers it onto the local SE associated to this site. Then, the subsequent jobs in the same site can directly benefit of a local file transfer hence optimizing the overall file transfer duration. This strategy derives of two observations made on EGI. First, when the application consists of a large number of jobs, a given site will execute more than one job in general. Second, the queuing time from a job submission to the job execution is highly variable. It means that if subsequent jobs have a much longer queuing time compared to the first job, they can directly benefit of the local transfer without any extra delay. More details are given in [5].

3.3 Simulation Scenarios

We simulate the execution of 15 workflows, each consisting of 100 jobs, to study the performance of file transfers. Realistic information are extracted from execution traces and injected as parameters in our simulator (e.g., the queuing time of jobs, execution site, source and destination of file transfers, . . .).

To determine the impact of SE selection for each platform model, we study three categories of prestaging lists: (i) the current production setting, which corresponds to three SEs located in France, (ii) 50 randomly selected lists and (iii) four prestaging lists selected based on statistical information on the sites where the jobs of the 15 workflows were executed. These four lists contain the local SEs of the three sites hosting the largest number of jobs located in one or different countries or three sites hosting no jobs at all located in one or different countries, respectively. We always fix the number of SEs used to prestage files to three to match the number of replicas currently used in production. The impact of the number of SEs is let out of the scope of this paper.

In total, we simulate 220 scenarios (2 strategies × 2 platform models × 55 prestaging lists) for each of the 15 workflows.

4 Performance Evaluation

4.1 Impact of Dynamic Replication

We begin our evaluation by studying the cumulative distribution of the simulated durations of file transfers with and without dynamic replication. Each line in Fig. 1 corresponds to one list of 3 SEs used for file prestaging, using either the

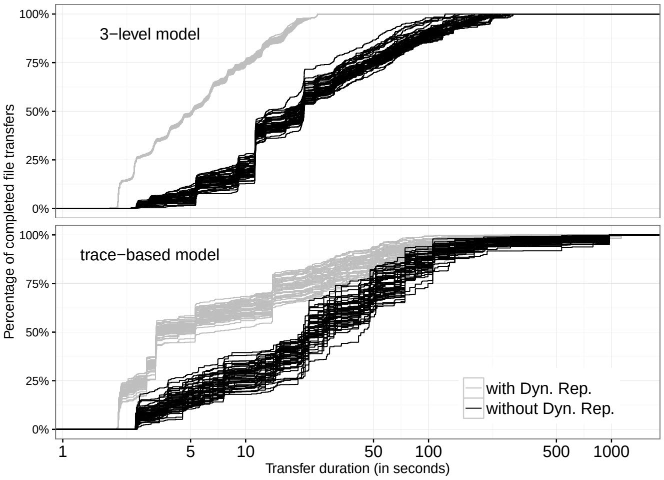

Fig. 1: Cumulative distribution of simulated file transfer durations with and without dynamic replication. Each line corresponds to a list of 3 SEs used for file pre-staging. The same 50 random prestaging lists are used in all four scenarios.

3-level (top) or the trace-based (bottom) platform model. The same 50 random prestaging lists are used in all four scenarios.

For the 3-level model, we see that dynamic replication significantly decreases file transfer durations, as more jobs can download files from a local SE. Moreover, the performance does not depend on the SEs used for prestaging with a median duration of 5.1s and a maximum value of 32.3s. Without dynamic replication, the choice of the prestaging list has a stronger impact, leading to longer and more variable transfer durations. The median varies from 13s to 21s when utilizing different lists while the maximum varies from 123s to 290s.

For the trace-based model, we also see a reduction of file transfer durations when using dynamic replication, but the gap is less clear. Contrary to the 3-level model, the performance with dynamic replication varies more significantly depending on the prestaging list. For both models, the choice of the prestaging list always has a strong impact on performance when there is no dynamic replication. Median duration varies from 20s to 44s while the maximum and the longest duration is about 975s when utilizing different lists.

4.2 Impact of Different Prestaging Lists on Static Replication

We saw that, globally, the choice of SEs used for prestaging mainly matters when there is no dynamic replication. To measure the impact of SE choice for file prestaging, we compare the 50 random prestaging lists, the 4 predefined lists and the current prestaging list used in production. The comparisons for the 3-level hierarchical (top) and trace based-model (bottom) are depicted in Fig. 2. We identify the best and the worst prestaging among these 55 lists based on

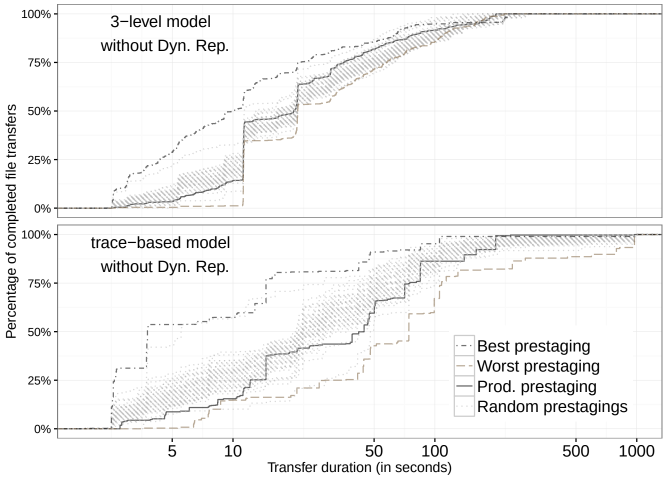

Fig. 2: Comparison of random, predefined, and the current production prestaging list without dynamic replication for two platform models

the median simulated file transfers duration. The performance corresponding to the current production prestaging list is also identified (named "prod prestaging"). It utilizes 3 SEs in France, chosen according to the criteria described in Sect. 2. Note that we only evaluate the impact of the prestaging list w.r.t. the file transfer duration. Other aspects taken into account by VIP administrators (e.g., reliability, availability and storage space of each SE) are left as future work.

For the 3-level model, the "best prestaging" corresponds to one of the predefined lists: three SEs associated to the sites hosting the largest number of jobs located in three different countries, i.e., UK, Netherlands, and France. By selecting the most used sites, most of the jobs can directly download files from their local SEs. Moreover, scattering file replicas in different countries can efficiently reduce the number of downloads from a foreign country. Conversely, the "worst prestaging" for the 3-level model is given by three SEs associated to sites that do not execute any job and are located in different countries. Thus, most of the jobs download files from a foreign country, which leads to the worst performance.

For the trace-based model, we find the exact same "best prestaging" and "worst prestaging" as for the 3-level model. It further validates the findings from the 3-level model. By collecting more historical information from the DIRAC [19] server that schedules the jobs, we find that UK, Netherlands, and France are the countries hosting the largest number of executed jobs in the Virtual Organization used by VIP. We can thus conclude that the best performance without dynamic replication is likely to be obtained by selecting the SE of the most used sites in different countries hosting the largest cumulative number of executed jobs for both models.

4.3 Impact of Platform Model on Replication Decisions

Figure 3 compares the duration of file transfers when using dynamic replication for the two models. We observe that dynamic replication leads to much more stable results in the 3-level model than in the trace-based model. In other words,

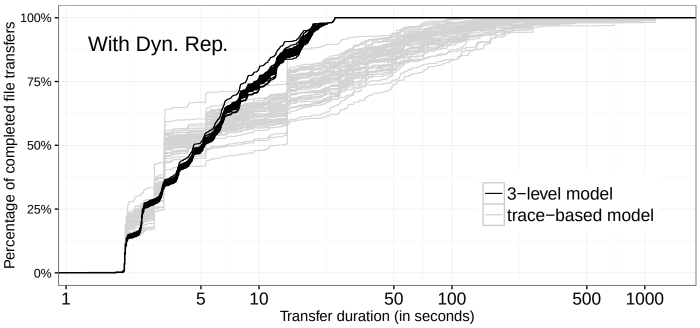

Fig. 3: Cumulative distribution of simulated file transfer durations with dynamic replication for two platform models

in the 3-level model, a random selection of SEs to prestage files is enough: no improved SE selection strategy is required. However, for the trace-based model, we observe a greater variability which can be explained by the important heterogeneity in terms of network connectivity that is better captured by this model. While a local SE may have a poor connectivity in the trace-based model, the 3-level model will always assumes a very good connectivity, which is one of its known limitations.

Figure 4 compares the best performance achieved by predefined or randomly selected lists without dynamic replication for each model. As in simulation we have the complete *a priori* information about the sites on which jobs are going to be executed, the best predefined prestaging list is always better than the best random list that we obtained. Interestingly, we see that the gain is much larger in trace-based model. The more heterogeneous the platform is, the more important *a priori* information (e.g., the distribution of executed jobs on computing sites or in countries) is to optimize file transfers.

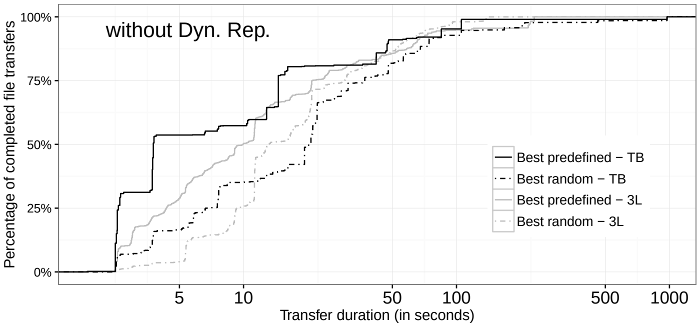

Fig. 4: Cumulative distribution of simulated file transfer durations without dynamic replication for two platform models. Best performance achieved by predefined or randomly selected lists is highlighted.

It is also interesting to note in Fig. 2 that the performance of the prestaging currently used in production is quite different between the 3-level and trace-based models. In the former, SEs are equivalent in the sense that a single bandwidth value is used for all the links in each category (i.e., local, inter-country, and intra-country). Performance will then be better for lists with SEs close to the sites executing most of the jobs. In the latter, each link is unique and the use of close SEs alone cannot ensure the best performance. The "prod prestaging" list illustrates this. It corresponds to three SEs in France, close to sites that execute more than 16% (which is more than the average sites) of the total number of jobs. However, the general connectivity for these three SEs is worse than the average. This explains why the performance of the "prod prestaging" list is better than most of the randomly selected prestaging lists in the 3-level model and worse in the trace-based model. It also shows that different platform models can lead to different qualitative assessments for similar scenarios.

5 Recommendations for File Replication in VIP on EGI

As we have seen, simulation results are not always consistent between the two models. A larger variability exists in the trace-based model even with dynamic replication. The relative performance of the current production configuration also differs from a model to another. Consequently, recommendations for VIP need to be based on the results obtained with the trace-based model.

Figure 5 compares the best and worst performance (with or without dynamic replication) to the current production setting. The performance with and without dynamic replication is depicted in black and gray, respectively.

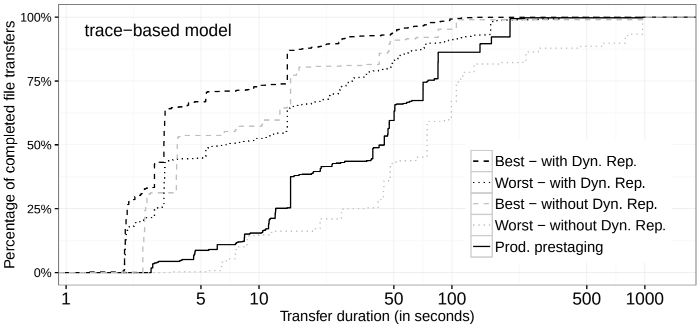

Fig. 5: Comparison of the best and the worst prestaging with the current production prestaging for trace-based model with or without dynamic replication

Without dynamic replication, a careful selection of the SEs used for file prestaging reduces file transfer times. However, this requires *a priori* information on where jobs are going to be executed. For jobs submitted independently in large distributed systems, we cannot know in advance where they will be executed. However, we could attempt to predict it by leveraging historical data on where the jobs have been running over a given period of time.

Dynamic replication always outperforms the current production configuration. To better quantify its gain, we computed in Table 1 the 95%-confidence

Table 1: 95%-confidence interval for the statistics of the simulated release transfers durations of 55 prestagings with and without dynamic replication

	1st Qu.	Median	Mean	3rd Qu.	Max
with Dyn. Rep.	[2.6;2.7]	[3.5;4.3]	[25.6;30.7]	[19.8;24.7]	[1192.1;1301.4]
without Dyn. Rep.	[8.3;11.2]	[22.9;28.23]	[60.5;71.2]	[57.8;66.9]	[974.4;974.8]

interval for the statistics on the simulated transfer durations over the 55 studied prestaging lists. We conclude that with dynamic replication, there is a 95% chance that 75% of file transfers will be 2.5 times shorther than without, regardless of the selected prestaging list.

However, the longest transfer duration seems to be worse with dynamic replication. In the proposed dynamic replication algorithm (details are given in [5]), the first job in a site tries to download the file using a timeout to reduce the impact of extremely long transfers [10]. If this timeout expires, this transfer is canceled and a new attempt is made with another SE. Then, the transfer time corresponds to the cumulative time of all transfer attempts (failed and successful). In the studied scenarios, the longest simulated transfer corresponds to a job executed on a site with poor connectivity to/from most SEs in the trace-based model. When using dynamic replication, the timeout expires 3 times, hence adding an overhead of three times the timeout value. This timeout is currently set to 110 seconds and corresponds to the third quartile of all measured transfer durations. This effect could be mitigated with a timeout value that makes a trade-off between the longest acceptable transfer duration and this extra overhead caused by retries. It is important to note that such an extreme case cannot be evaluated with the 3-level model that does not reflect the heterogeneity of the actual infrastructure.

To summarize, we can conclude from our observations that dynamic replication can globally reduce the duration of file transfers except for extreme cases where multiple transfer timeouts are hit successively. Such cases are only captured by the trace-based platform model. As the benefits of dynamic replication comes from the number of jobs that transfer files from a local SE thanks to the copy made by the first job, it may not be interesting for small applications. Finally, implementing such a dynamic replication strategy in the production environment would require non-negligible development effort for the correct handling of concurrent file access synchronization, as well as finding the optimal parameters (e.g., the timeout value and the maximum number of retries).

6 Conclusion

File replication is a widely used technique to optimize data management in distributed systems. Many replication strategies have been proposed in the literature to solve various optimization problems in which efficiency has mostly been evaluated through simulation. However, the often simplified configuration of simulators may critically question the findings derived from simulation results.

In this paper, we presented our efforts to improve the evaluation of file replication strategies by studying two platform models: a 3-level hierarchical model and a model built out of execution traces. We evaluated the impact of different strategies on file transfer durations and compared the results obtained with each model to cross-validate our findings. Last but not least, we proposed recommendations to optimize the replication management for VIP.

Simulation results show that the estimated impact of a strategy can be quite different when the platform model changes. In other words, the conclusion drawn from one model cannot be automatically transferred to another. We show that the instantiation of the two models leads to different qualitative decisions, even though they reflect a similar hierarchical topology. It emphasizes the fact that the realism of the platform model is key to the evaluation process.

By comparing the results obtained with each model, we found that selecting the sites hosting a large number of executed jobs to prestage files is a safe recommendation to optimize data management in the production system. In addition, adopting dynamic replication can further reduce the duration of file transfers except for extreme cases (very poorly connected sites) that our realistic simulations were able to capture.

All the simulation results presented in this article are available online along with all the code and data used to produce them [5]. This material allows readers and reviewers to reproduce and further investigate our results.

As future work, we plan to further improve the accuracy of our trace-based model by collecting more execution traces and evaluate different methods to fill the missing links. It would also be interesting to investigate the influence of the number of replicas and other important parameters (e.g., timeout value) for our strategy and take into account other parameters (e.g., transfer failure rate, storage space, etc) in the simulation scenarios. We also plan to build probability distributions out of the real execution traces. Integrating them into the simulator would allow us to study different "what if" scenarios.

Acknowlegments

This work is partially supported by the LABEX PRIMES (ANR-11-LABX-0063) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

References

1. Bsoul, M., Abdallah, A., Almakadmeh, K., Tahat, N.: A Round-Based Data Replication Strategy. IEEE TPDS **27**(1), 31–39 (2016)
2. Camarasu-Pop, S., Glatard, T., Benoit-Cattin, H.: Simulating Application Workflows and Services Deployed on the European Grid Infrastructure. In: Proceedings of the 13th IEEE/ACM International. Symposium on Cluster, Cloud, and Grid Computing. pp. 18–25 (2013)
3. Casanova, H., Giersch, A., Legrand, A., Quinson, M., Suter, F.: Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms. Journal of Parallel and Distributed Computing **74**(10), 2899–2917 (2014)
4. Chai, A., Bazm, M.M., Camarasu-Pop, S., Glatard, T., Benoit-Cattin, H., Suter, F.: Modeling Distributed Platforms from Application Traces for Realistic File Transfer Simulation. In: Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. pp. 54–63 (2017)

5. Chai, A., Camarasu-Pop, S., Glatard, T., Benoit-Cattin, H., Suter, F.: Companion of article "Evaluation through Realistic Simulations of File Replication Strategies for Large Heterogeneous Distributed Systems (2018), Available at: http://doi.org/10.5281/zenodo.1239677
6. Chervenak, A., et al.: Data placement for scientific applications in distributed environments. In: Proceedings of the 8th IEEE/ACM International Conference on Grid Computing. pp. 267–274 (2007)
7. David, M., et al.: Validation of Grid Middleware for the European Grid Infrastructure. Journal of Grid Computing **12**(3), 543–558 (Sep 2014)
8. Dayyani, S., Khayyambashi, M.: RDT: A New Data Replication Algorithm for Hierarchical Data Grid. International Journal of Computer Science and Engineering **3**(7), 186–197 (2015)
9. Elghirani, A., Subrata, R., Zomaya, A.: A Proactive Non-Cooperative Game-Theoretic Framework for Data Replication in Data Grids. In: Proceedings of the 8th IEEE International Symposium on Cluster Computing and the Grid. pp. 433–440 (2008)
10. Glatard, T., Montagnat, J., Pennec, X.: Optimizing Jobs Timeouts on Clusters and Production Grids. In: Proceedings of the 7th IEEE International Symposium on Cluster Computing and the Grid. pp. 100–107 (2007)
11. Glatard, T., et al.: A Virtual Imaging Platform for Multi-Modality Medical Image Simulation. IEEE Transactions on Medical Imaging **32**(1), 110–118 (2013)
12. Gupta, H., et al.: iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in The Internet of Things, Edge and Fog Computing Environments. Software: Practice and Experience **47**(9), 1275–1296 (2017)
13. Lamehamedi, H., et al.: Data Replication Strategies in Grid Environments. In: Proceedings of the 5th International Conference on Algorithms and Architectures for Parallel Processing. pp. 378–383 (2002)
14. Lei, M., Vrbsky, S., Hong, X.: An On-Line Replication Strategy to Increase Availability in Data Grids. Future Generation Computing Systems **24**(2), 85–98 (2008)
15. Ranganathan, K., Foster, I.: Simulation studies of computation and data scheduling algorithms for data grids. Journal of Grid computing **1**(1), 53–62 (2003)
16. Sato, H., Matsuoka, S., Endo, T., Maruyama, N.: Access-Pattern and Bandwidth Aware File Replication Algorithm in a Grid Environment. In: Proceedings of the 9th IEEE/ACM International Conference on Grid Computing. pp. 250–257 (2008)
17. Shorfuzzaman, M., Graham, P., Eskicioglu, R.: Adaptive Popularity-Driven Replica Placement in Hierarchical Data Grids. The Journal of Supercomputing **51**(3), 374–392 (2010)
18. Suter, F., Chai, A., Camarasu-Pop, S.: VIPSimulator: a Simulator of Gate Workflow Execution. Available at: http://github.com/frs69wq/VIPSimulator (2016)
19. Tsaregorodtsev, A., et al.: DIRAC3 – the New Generation of the LHCb Grid Software. Journal of Physics: Conference Series **219**(6), 062029 (2010)
20. Vrbsky, S., Lei, M., Smith, K., Byrd, J.: Data Replication and Power Consumption in Data Grids. In: Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science. pp. 288–295 (2010)
21. Yang, C.T., Fu, C.P., Hsu, C.H.: File Replication, Maintenance, and Consistency Management Services in Data Grids. The Journal of Supercomputing **53**(3), 411–439 (2010)