IN PARTNERSHIP WITH:
CNRS

Ecole normale supérieure de Cachan

Activity Report 2011

Team KERDATA

Scalable Storage for Clouds and Beyond

IN COLLABORATION WITH: Institut de recherche en informatique et systèmes aléatoires (IRISA)

RESEARCH CENTER
Rennes - Bretagne-Atlantique

THEME
Distributed and High Performance Computing

Table of contents

Team KERDATA

Keywords: High Performance Computing, Cloud Computing, Middleware, Data Management, Data Storage

The KerData Team has been officially created on July 1st, 2009. It is a spinoff of the Paris Project-Team. It corresponds to the former "Data management" activity of the Paris Project-Team. Project-Team.

1. Members

Research Scientist

Gabriel Antoniu [Team leader, Junior Researcher (CR1) INRIA, HdR]

Faculty Member

Luc Bougé [Professor, ENS CACHAN Brittany Campus, HdR]

PhD Students

Alexandra Carpen-Amarie [INRIA CORDI-S Grant until September 30, 2010. Then, on a temporary ACET research position until December 31, 2011. PhD thesis defended on December 15, 2011.]

Diana Moise [INRIA and Brittany Regional Council Grant until September 30, 2010. Then, on a temporary ACET research position until December 31, 2011. PhD thesis defended on December 16, 2011.]

Viet-Trung Tran [MESR Grant]

Houssem-Eddine Chihoub [European Marie-Curie Scalus Project Grant. PhD thesis started on 1 September 2010.]

Radu Tudoran [MESR Grant, starting October 1, 2011]

Matthieu Dorier [4th-year ENS student, starting October 1, 2011]

Post-Doctoral Fellows

Alexandru Costan [ANR MapReduce Project, started February 14, 2011.]

Louis-Claude Canon [A-Brain MSR-INRIA Project, started on September 1, 2011.]

Shadi Ibrahim [HEMERA Large-Wingspan Project, started on November 2, 2011.]

Visiting Scientists

Bunjamin Memishi [Visiting PhD student, Universidad Politecnica de Madrid (UPM), 1 month (April 2011), funded by Universidad Politecnica de Madrid through the SCALUS Marie-Curie Initial Training Network. Thesis co-advised by Mariá Pérez (UPM) and Gabriel Antoniu (KerData).]

Florin Pop [Visiting Postdoc Fellow, Polytechnic University of Bucharest, 1 month (June 2011), funded by the DataCloud@work Associate Team]

Ciprian Dobre [Visiting Postdoc Fellow, Polytechnic University of Bucharest, 1 month (June 2011), funded by the DataCloud@work Associate Team]

Daniel Higuero [Visiting PhD student, Carlos III University, Madrid, 3 months (September-November 2011), funded by Carlos III University, Madrid]

Elena Apostol [Visiting PhD student, Polytechnic University of Bucharest, 3 months to start in June 2011, funded by the DataCloud@work Associate Team]

Administrative Assistants

Maryse Fouché [Team Administrative Assistant (TR) INRIA, until November 13, 2011.]

Céline Gharsalli [Team Administrative Assistant CNRS, since November 14, 2011.]

2. Overall Objectives

2.1. General context and our focus

We are witnessing a rapidly increasing number of application areas generating and processing very large volumes of data on a regular basis. Such applications are called *data-intensive*. Governmental and commercial statistics, climate modeling, cosmology, genetics, bio-informatics, high-energy physics are just a few examples. In these fields, it becomes crucial to efficiently store and manipulate massive data, which are typically *shared* at a large scale and *concurrently accessed*. In all these examples, the overall application performance is highly dependent on the properties of the underlying data management service. With the emergence of recent infrastructures such as cloud computing platforms and post-Petascale architectures, achieving highly scalable data management has become a critical challenge.

Our research activities focus on data-intensive high-performance applications that exhibit the need to handle:

- massive unstructured data, BLOBs (Binary Large OBjects), in the order of Terabytes,
- stored in a large number of nodes, thousands to tens of thousands,
- accessed under heavy concurrency by a large number of processes, thousands to tens of thousands at a time,
- with a relatively fine access grain, in the order of Megabytes.

Examples of such applications are:

- Massively parallel cloud data-mining applications (e.g., MapReduce-based data analysis).
- Advanced Platform-as-a-Service (PaaS) cloud data services requiring efficient data sharing under heavy concurrency.
- Advanced concurrency-optimized, versioning-oriented cloud services for virtual machine image storage and management at IaaS (Infrastructure-as-a-Service) level.
- Scalable storage solutions for I/O-intensive HPC simulations for post-Petascale architectures.

2.2. Highlights

Gilles Kahn/SPECIF PhD Thesis Award 2011. Bogdan Nicolae, former PhD student in the KerData Team (defense on November 30, 2010) won the 2nd Prize at the 2011 Gilles Kahn/SPECIF PhD Thesis Award for his thesis entitled BlobSeer*: Towards efficient data storage management for large-scale, distributed systems*.

ACM Student Research Competition at ICS 2011. Matthieu Dorier, Master intern at KerData in Summer 2011 (and now a PhD student there), was awarded the 2nd Prize at the ACM Student Research Competition organized in the framework of the ICS 2011 Conference (http://ics-conference.org/ , Tucson, Arizona, May 2011), for his poster entitled *Damaris - Using Dedicated I/O Cores for Scalable Post-petascale HPC Simulations* [16].

IEEE TCPP Best Poster Award at IPDPS 2011. Alexandra Carpen-Amarie, 3rd-year PhD student at KerData, won one of the 3 IEEE TCPP Best Poster Awards at IPDPS 2011 (http://www.ipdps.org/ipdps2011/, Anchorage, Alaska, USA, May 2011) for her poster entitled *Towards a Self-Adaptive Data Management System for Cloud Environments* [15].

3. Scientific Foundations

3.1. Our goals and methodology

Managing data at large scales is paramount nowadays and many application areas exhibit a need for efficient scaling to huge data sizes: data mining applications [55], multimedia applications [46], database-oriented applications ([49], [67], [62]), bioinformatic applications, etc. In such contexts, one important goal is to provide mechanisms allowing to transparently manage massive data blocks (e.g., of several terabytes), while providing efficient, fine-grain access to small parts of the data.

The overall goal of the KerData team is to bring a substantial contribution to the effort of the research community to address the above challenges. More specifically, to support the large-scale execution of the applications we described, KerData aims to design and implement distributed algorithms for scalable data storage and input/output management for efficient large-scale data processing. We target two main execution infrastructures: cloud platforms and post-Petascale HPC supercomputers. We are also looking at other kinds of infrastructures (that we are considering as secondary), e.g. hybrid platforms combining enterprise desktop grids extended to cloud platforms.

Our approach relies on building prototypes and on their large-scale experimental validation on real testbeds and experimental platforms. In our current projects, our target platforms include: the Grid'5000 testbed, Amazon and Microsoft's Azure commercial clouds, public clouds based on open-source IaaS toolkits such as Nimbus and OpenNebula. In the HPC area we have access to the Jaguar and Kraken supercomputers (ranked 3rd and 11th respectively in the Top 500 supercomputer list). Last but not least, our methodology includes large-scale validations of our solutions with real-life applications, such as the ones described in Section 4.1. To this purpose, we have started to build partnerships with the application communities that can potentially benefit from our contributions and we will continue to do so in future collaborative projects.

.

3.2. Transparent, distributed data sharing

The management of massive data blocks naturally requires the use of data fragmentation and of distributed storage. Grid infrastructures, typically built by aggregating distributed resources that may belong to different administration domains, were built during the last years with the goal of providing an appropriate solution. When considering the existing approaches to grid data management, we can notice that most of them heavily rely on *explicit* data localization and on *explicit* transfers of large amounts of data across the distributed architecture: GridFTP [40], LDR [35], Chirp [29], IBP [41], NeST [42], etc. Managing huge amounts of data in such an explicit way at a very large scale makes the design of grid applications much more complex. One key issue to be addressed is therefore the *transparency* with respect to data localization and data movements. Such a transparency is highly suitable, as it alleviates the user of the need to handle data localization and transfers.

3.3. Managing massive unstructured data under heavy concurrency on large-scale distributed infrastructures

3.3.1. Massive unstructured data: BLOBs

Studies show more than 80% [53] of data globally in circulation is unstructured. On the other hand, data sizes increase at a dramatic level with more than 1 TB of data gathered per week in common scenarios for some production applications (e.g., medical experiments [65]). Finally, on Post-Petascale HPC machines, the use of huge storage objects is also currently being considered as a promising alternative to today's dominant approaches to data management. Indeed, these approaches rely on very large numbers of small files, and using huge storage objects reduces the corresponding metadata overhead of the file system. Such huge unstructured data are stored as *binary large objects (BLOBs)* that may continuously be updated by applications. However, traditional databases or file systems can hardly cope in an efficient way with BLOBs which grow to huge sizes.

3.3.2. Scalable processing of massive data: heavy access concurrency

To address the scalability issue, specialized abstractions like MapReduce [47] and Pig-Latin [63] propose high-level data processing frameworks intended to hide the details of parallelization from the user. Such platforms are implemented on top of huge object storage platforms. They target high performance by optimizing the parallel execution of the computation. This leads to *heavy access concurrency* to the BLOBs, thus the need for the storage layer to offer support in this regard. Parallel and distributed file systems also consider using objects for low-level storage (see next subsection [48], [69], [51]). In other application areas, huge BLOBs need to be used concurrently at the highest level layers of applications directly: high-energy physics, multimedia processing [46] or astronomy.

3.3.3. Versioning

When addressing the problem of storing and efficiently accessing very large unstructured data objects [60], [65] in a distributed environment, a challenging case is the one where data is *mutable* and potentially accessed by a very large number of concurrent, distributed processes. In this context, *versioning* is an important feature. Not only it allows to roll back data changes when desired, but it also enables cheap branching (possibly recursively): the same computation may proceed independently on different versions of the BLOB. Versioning should obviously not impact access performance to the object significantly, given that objects are under constant heavy access concurrency. On the other hand, versioning leads to increased storage space usage and becomes a major concern when the data size itself is huge. Versioning efficiency thus refers to both access performance under heavy load and reasonably acceptable overhead of storage space.

3.4. Towards scalable, BLOB-based distributed file systems

Recent research [50] emphasizes a clear move currently in progress from a block-based interface to a object-based interface in storage architectures. The goal is to enable scalable, self-managed storage networks by moving low-level functionalities such as space management to storage devices or to storage server, accessed through a standard object interface. This move has a direct impact on the design of today's distributed file systems: object-based file system would then store data rather as objects than as unstructured data blocks. According to [50], this move may eliminate nearly 90% of management workload which was the major obstacle limiting file systems' scalability and performance.

Two approaches exploit this idea. In the first approach, the data objects are stored and manipulated directly by a new type of storage device called *object-based storage device* (OSD). This approach requires an evolution of the hardware, in order to allow high-level object operations to be delegated to the storage device. Examples of parallel/distributed file systems following this approach are Lustre [66] and Ceph [69]. Recently, research efforts [48] have explored the feasibility and the possible benefits of integrating OSDs into parallel file systems, such as PVFS [45].

The second approach does not rely on the presence of OSDs, but still tries to benefit from an object-based approach to improve performance and scalability: files are structured as a set of objects that are stored on storage servers. Google File System [51], and HDFS (*Hadoop File System*) [33] illustrate this approach.

3.5. Emerging large-scale infrastructures for distributed applications

During the last few years, research and development in the area of large-scale distributed computing led to the clear emergence of several types of physical execution infrastructures for large-scale distributed applications.

3.5.1. Cloud computing infrastructures

The cloud computing model [68], [59], [44] is gaining serious interest from both industry and academia in the area of large-scale distributed computing. It provides a new paradigm for managing computing resources: instead of buying and managing hardware, users rent virtual machines and storage space.

Various cloud software stacks have been proposed by leading industry companies, like Google, Amazon or Yahoo!. They aim at providing fully configurable virtual machines or virtual storage (IaaS: *Infrastructure-as-a-Service*), higher-level services including programming environments such as MapReduce [47] (*PaaS: Platform-as-a-Service* [31], [36]) or community-specific applications (*SaaS: Software-as-a-Service* [32], [37]). On the academic side, two of the most visible projects in this area are Nimbus [38], [57] from the Argonne National Lab (USA) and OpenNebula [39], which aim at providing a reference implementation for a IaaS.

In the context of the emerging cloud infrastructures, some of the most critical open issues relate to data management. Providing the users with the possibility to store and process data on externalized, virtual resources from the cloud requires simultaneously investigating important aspects related to security, efficiency and quality of service. To this purpose, it clearly becomes necessary to create mechanisms able to provide feedback about the state of the storage system along with the underlying physical infrastructure. The information thus monitored, can further be fed back into the storage system and used by self-managing engines, in order to enable an autonomic behavior [58], [64], [54], possibly with several goals such as self-configuration, self-optimization, or self-healing. Exploring ways to address the main challenges raised by data storage and management on cloud infrastructures is the major factor that motivated the creation of the KerData research team INRIA RENNES – BRETAGNE ATLANTIQUE. These topics are at the heart of our involvement in several projects that we are leading in the area of cloud storage: MapReduce (see Section 6.1), AzureBrain (see Section 6.1), DataCloud@work (see Section 6.3).

3.5.2. Petascale infrastructures

In 2011, a new NSF-funded Petascale computing system, Blue Waters, will go online at the University of Illinois. Blue Waters is expected to be the most powerful supercomputer in the world for open scientific research when it comes online. It will be the first system of its kind to sustain one-Petaflop performance on a range of science and engineering applications. The goal of this facility is to open up new possibilities in science and engineering. It provides unheard computational capability. It makes it possible for investigators to tackle much larger and more complex research challenges across a wide spectrum of domains: predict the behavior of complex biological systems, understand how the cosmos evolved after the Big Bang, design new materials at the atomic level, predict the behavior of hurricanes and tornadoes, and simulate complex engineered systems like the power distribution system and airplanes and automobiles.

To reach sustained-Petascale performance, machines like Blue Waters relies on advanced, dedicated technologies at several levels: processor, memory subsystem, interconnect, operating system, programming environment, system administration tools. In this context, data management is again a critical issue that highly impacts the application behavior and its overall performance. Petascale supercomputers exhibit specific architectural features (e.g., a multi-level memory hierarchy scalable to tens to hundreds of thousands of codes) that needs to be specifically taken into account. Providing scalable data throughput on such unprecedented scales is clearly an open challenge today. In this context, we are investigating techniques to achieve concurrency-optimized I/O in collaboration with teams from the National Center for Supercomputing Applications (NCSA/UIUC) in the framework of the Joint INRIA-UIUC for Petascale Computing (see Section 6.6).

3.6. Emerging programming models for scalable data-management

MapReduce is a parallel programming paradigm successfully used by large Internet service providers to perform computations on massive amounts of data. A computation takes a set of input key/value pairs, and produces a set of output key/value pairs. The user of a MapReduce library expresses the computation as two functions: *map*, that processes a key/value pair to generate a set of intermediate key/value pairs, and *reduce*, that merges all intermediate values associated with the same intermediate key. The framework takes care of splitting the input data, scheduling the jobs' component tasks, monitoring them and re-executing the failed ones. After being strongly promoted by Google, it has also been implemented by the open source community through the Hadoop project, maintained by the Apache Foundation and supported by Yahoo! and even by Google itself. This model is currently getting more and more popular as a solution for rapid implementation of distributed data-intensive applications. The key strength of the MapReduce model is its inherently high degree of potential parallelism that should enable processing of Petabytes of data in a couple of hours on large clusters consisting of several thousand nodes.

At the core of the MapReduce frameworks stays a key component: the storage layer. To enable massively parallel data processing to a high degree over a large number of nodes, the storage layer must meet a series of specific requirements. Firstly, since data is stored in huge files, the computation will have to efficiently process small parts of these huge files concurrently. Thus, the storage layer is expected to provide efficient *fine-grain* *access* to the files. Secondly, the storage layer must be able to sustain a *high throughput* in spite of *heavy access concurrency* to the same file, as thousands of clients simultaneously access data.

These critical needs of data-intensive distributed applications have not been addressed by classical, POSIX-compliant distributed file systems. Therefore, specialized file systems have been designed, such as HDFS, the default storage layer of Hadoop. HDFS has however some difficulties in sustaining a high throughput in the case of concurrent accesses to the same file. Amazon's cloud computing initiative, Elastic MapReduce, employs Hadoop on their Elastic Compute Cloud infrastructure (EC2) and inherits these limitations. The storage back-end used by Hadoop is Amazon's Simple Storage Service (S3), which provides limited support for concurrent accesses to shared data. Moreover, many desirable features are missing altogether, such as the support for versioning and for concurrent updates to the same file. Finally, another important requirement for the storage layer is its ability to expose an interface that enables the application to be *data-location aware*. This is critical in order to allow the scheduler to use this information to place computation tasks close to the data and thus reduce network traffic, contributing to a better global data throughput. These topics are at the core of KerData's contribution to the MapReduce ANR project and to the Hemera large wingspan project, both started in 2010, see Section 8.2.

4. Application Domains

4.1. Motivating applications

Below are three examples which illustrate the needs of large-scale data-intensive applications with respect to storage, I/O and data analysis. We have begun working on such applications in the context of our current projects started in 2010. They illustrate the classes of applications that can benefit from our research activities.

4.1.1. Joint genetic and neuroimaging data analysis on Azure clouds

Joint acquisition of neuroimaging and genetic data on large cohorts of subjects is a new approach used to assess and understand the variability that exists between individuals, and that has remained poorly understood so far. As both neuroimaging- and genetic-domain observations represent a huge amount of variables (of the order of millions), performing statistically rigorous analyses on such amounts of data is a major computational challenge that cannot be addressed with conventional computational techniques only. On the one hand, sophisticated regression techniques need to be used in order to perform significant analysis on these large datasets; on the other hand, the cost entailed by parameter optimization and statistical validation procedures (e.g. permutation tests) is very high.

The A-Brain (AzureBrain) Project started in October 2010 within the Microsoft Research-INRIA Joint Research Center. It is co-led by the KerData (Rennes) and Parietal (Saclay) INRIA teams. They jointly address this computational problem using cloud related techniques on Microsoft Azure cloud infrastructure. The two teams bring together their complementary expertise: KerData in the area of scalable cloud data management, and Parietal in the field of neuroimaging and genetics data analysis.

In particular, KerData brings its expertise in designing solutions for optimized data storage and management for the Map-Reduce programming model. This model has recently arisen as a very effective approach to develop high-performance applications over very large distributed systems such as grids and now clouds. The computations involved in the statistical analysis designed by the Parietal team fit particularly well with this model.

4.1.2. Structural protein analysis on Nimbus clouds

Proteins are major components of the life. They are involved in lots of biochemical reactions and vital mechanisms for the living organisms. The three-dimensional (3D) structure of a protein is essential for its function and for its participation to the whole metabolism of a living organism. However, due to experimental limitations, only few protein structures (roughly, 60,000) have been experimentally determined, compared to the millions of proteins sequences which are known. In the case of structural genomics, the knowledge of the

3D structure may be not sufficient to infer the function. Thus, an usual way to make a structural analysis of a protein or to infer its function is to compare its known, or potential, structure to the whole set of structures referenced in the *Protein Data Bank* (PDB).

In the framework of the MapReduce ANR project led by KerData, we focus on the SuMo application (*Surf the Molecules*) proposed by Institute for Biology and Chemistry of the Proteins from Lyon (IBCP, a partner in the Map-Reduce project). This application performs structural protein analysis by comparing a set of protein structures against a very large set of structures stored in a huge database. This is a typical data-intensive application that can leverage the Map-Reduce model for a scalable execution on large-scale distributed platforms. Our goal is to explore storage-level concurrency-oriented optimizations to make the SuMo application scalable for large-scale experiments of protein structures comparison on cloud infrastructures managed using the Nimbus IaaS toolkit developed at Argonne National Lab (USA).

If the results are convincing, then they can immediately be applied to the derived version of this application for drug design in an industrial context, called MED-SuMo, a software managed by the MEDIT SME (also a partner in this project). For pharmaceutical and biotech industries, such an implementation run over a cloud computing facility opens several new applications for drug design. Rather than searching for 3D similarity into biostructural data, it will become possible to classify the entire biostructural space and to periodically update all derivative predictive models with new experimental data. The applications in that complete chemo-proteomic vision concern the identification of new druggable protein targets and thereby the generation of new drug candidates.

4.1.3. I/O intensive climate simulations for the Blue Waters post-Petascale machine

A major research topic in the context of HPC simulations running on post-Petascale supercomputers is to explore how to efficiently record and visualize data during the simulation without impacting the performance of the computation generating that data. Conventional practice consists in storing data on disk, moving it off-site, reading it into a workflow, and analyzing it. It becomes increasingly harder to use because of the large data volumes generated at fast rates, in contrast to limited back-end speeds. Scalable approaches to deal with these I/O limitations are thus of utmost importance. This is one of the main challenges explicitly stated in the roadmap of the Blue Waters Project (http://www.ncsa.illinois.edu/BlueWaters/), which aims to build one of the most powerful supercomputers in the world when it comes online in 2012.

In this context, the KerData team started to explore ways to remove the limitations mentioned above through a collaborative work in the framework of the Joint INRIA-UIUC Lab for Petascale Computing (JLPC, Urbana-Champaign, Illinois, USA), whose research activity focuses on the Blue Waters project. As a starting point, we are focusing on a particular tornado simulation code called CM1 (Cloud Model 1), which is intended to be run on the Blue Waters machine. Preliminary investigation demonstrated the inefficiency of the current I/O approaches, which typically consists in periodically writing a very large number of small files. This causes burst of I/O in the parallel file system, leading to poor performance and extreme variability (jitter) compared to what could be expected from the underlying hardware. The challenge here is to investigate how to make an efficient use of the underlying file system by avoiding synchronization and contention as much as possible. In collaboration with the JLPC, we started to address those challenges through an approach based on dedicated I/O cores.

4.2. Core challenges for scalable data-intensive storage and processing

Although they are issued from different application areas, the above three examples of data-intensive applications illustrate common requirements with respect to the need for data storage and I/O processing. These requirements lead to several core challenges discussed below.

4.2.1. Challenges related to cloud storage

In the area of cloud data management, a significant milestone is the emergence of the Map-Reduce [47] parallel programming paradigm. Today, it is successfully used on most cloud platforms, following the trend set up by Amazon [28]. The key strength of this model is its inherently high degree of potential parallelism.

Actually, it has been demonstrated that it enables processing Petabytes of data in a couple of hours, on large clusters consisting of several thousand nodes. At the core of the Map-Reduce frameworks stays a key component, which must meet a series of specific requirements.

First, since data is stored in huge files, the computation must efficiently process small parts of these huge files concurrently. Thus, the storage layer is expected to provide efficient *fine-grain access* to the files. Second, the storage layer must be able to sustain a *high throughput* in spite of *heavy access concurrency* to the same file, as thousands of clients simultaneously access data, while preserving *fault-tolerance* and *security* requirements.

Our goal is precisely to address these challenges by proposing scalable data management techniques meeting those properties to support Map-Reduce-based, data-intensive applications. Thanks to partnerships with leading teams in the area of cloud computing, both in the Academia (the Nimbus team at Argonne National Lab) and Industry (the Microsoft Azure team), we anticipate our contributions can have a high potential impact.

4.2.2. Challenges related to data-intensive HPC applications

The requirements exhibited by the climate simulation described above specifically highlights a major, more general research topic. It has been clearly identified by international panels of experts like IESP [34] and EESI [30], in the context of HPC simulations running on post-Petascale supercomputers. It aims to explore how to store and analyze massive outputs of data during and after the simulation without impacting the overall performance.

A jump of one order of magnitude in the size of numerical simulations is required to address some of the fundamental questions in several communities such as climate modeling, solid earth sciences or astrophysics. Scientists, codes and computing infrastructure are in an advanced stage for this, but the lack of data-intensive infrastructure and methodology to analyze huge simulations is a growing limiting factor. Our goal is to contribute to the removal of this bottleneck through innovative data storage techniques.

5. Software

5.1. BlobSeer

Contact: Gabriel Antoniu, gabriel.antoniu@inria.fr.

Participants from the KerData team: Alexandra Carpen-Amarie, Diana Moise, Viet-Trung Tran, Alexandu Costan, Gabriel Antoniu, Luc Bougé.

Presentation: BlobSeer is the core software platform for most current projects of the KerData team. It is a data storage service specifically designed to deal with the requirements of large-scale data-intensive distributed applications that abstract data as huge sequences of bytes, called BLOBs (Binary Large OBjects). It provides a versatile versioning interface for manipulating BLOBs that enables reading, writing and appending to them.

BlobSeer offers both scalability and performance with respect to a series of issues typically associated with the data-intensive context: *scalable aggregation of storage space* from the participating nodes with minimal overhead, ability to store *huge data objects*, *efficient fine-grain access* to data subsets, *high throughput in spite of heavy access concurrency*, as well as *fault-tolerance*.

Users: Work is currently in progress in several formalized projects (see previous section) to integrate and leverage BlobSeer as a data storage back-end in the reference cloud environments: a) Microsoft Azure; b) the Nimbus cloud toolkit developed at Argonne National Lab (USA); and c) in the OpenNebula IaaS cloud environment developed at UCM (Madrid).

URL: http://blobseer.gforge.inria.fr/

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on INRIA's forge. Version 1.0 (released late 2010) registered with APP: IDDN.FR.001.310009.000.S.P.000.10700.

5.2. Damaris

Contact: Gabriel Antoniu, gabriel.antoniu@inria.fr.

Participants from the KerData team: Matthieu Dorier, Gabriel Antoniu.

Presentation: Damaris is a middleware for multicore SMP nodes enabling them to efficiently handle data transfers for storage and visualization. The key idea is to dedicate one or a few cores of each SMP node to the application I/O. It is developed within the framework of a collaboration between KerData and the Joint Laboratory for Petascale Computing (JLPC). The current version enables efficient asynchronous I/O, hiding all I/O related overheads such as data compression and post-processing. On-going work is targeting fast direct access to the data from running simulations, and efficient I/O scheduling.

Users: Damaris has been preliminarily evaluated at NCSA (Urbana-Champaign) with the CM1 tornado simulation code. CM1 is one of the target applications of the Blue Waters supercomputer developed by at NCSA/UIUC (USA), in the framework of the INRIA-UIUC Joint Lab (JLPC). Work is currently in progress to use Damaris as an intermediate data layer optimizing simulation/visualization coupling for several HPC scientific applications intended to run on the Blue Waters.

URL: http://damaris.gforge.inria.fr/

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on INRIA's forge. Registration with APP is in progress.

5.3. Derived software

Derived from BlobSeer, two additional platforms are currently being developed within KerData: 1) Pyramid, a software service for array-oriented active storage developed within the framework of the PhD thesis of Viet-Trung Tran (see Section 6.7); and 2) TomusBlobs, a PaaS-level storage service for Azure clouds developed within the framework of the thesis of Radu Tudoran in relation to the A-Brain project. These platforms have not been publicly released yet.

6. New Results

6.1. BlobSeer and Map-Reduce programming

6.1.1. BlobSeer-based cloud storage

Participants: Alexandra Carpen-Amarie, Alexandru Costan, Gabriel Antoniu, Luc Bougé.

As data volumes generated and processed by such applications increase, a key requirement that directly impacts the adoption rate of the Cloud paradigm is efficient and reliable data management. In this context, we investigate the requirements of Cloud data services in terms of data-transfer performance and access patterns and we explore the ways to leverage and adapt existing data-management solutions for Cloud workloads. We aim at building a Cloud data service both compatible with state-of-the-art Cloud interfaces and able to deliver high-throughput data storage.

To achieve this goal, we developed a file system layer on top of BlobSeer, which exposes a hierarchical file namespace enhanced with the concurrency-optimized BlobSeer primitives. Furthermore, we integrated the BlobSeer file system as a backend for Cumulus, an efficient open-source Cloud storage service. We validated our approach through extensive evaluations performed on Grid'5000. We devised a set of synthetic benchmarks to measure the performance and scalability of the Cumulus system backed by BlobSeer, showing it can sustain high-throughput data transfers for up to 200 concurrent clients.

Next, we explored the advantages and drawbacks of employing Cloud storage services for distributed applications that manage massive amounts of data. We investigated two types of applications. We relied on an atmospheric phenomena modeling application to conduct a set of evaluations in a Nimbus Cloud environment. This application is representative for a large class of simulators that compute the evolution in time set of parameters corresponding to specific points in a spatial domain. As a consequence, such applications generate important amounts of output data. We evaluated an S3-compliant Cloud storage service as a storage solution for the generated data. To this end, we employed distributed Cumulus services backed by various storage systems. The reason for targeting this approach is that storing output data directly into the Cloud as the application progresses can benefit higher-level applications that further process such simulation data. As an example, visualization tools need to have real-time access to output data for analysis and filtering purposes.

We built an interfacing module to enable the application to run unmodified in a Cloud environment and to send output data to an S3-based Cloud service. Our experiments show that distributed Cumulus backends, such as BlobSeer or PVFS, sustain a constant throughput even when the number of application processes that concurrently generate data becomes 3 times higher than the number of storage nodes.

6.1.2. Optimizing Intermediate Data Management in MapReduce Computations

Participants: Diana Moise, Gabriel Antoniu, Luc Bougé.

MapReduce applications, as well as other cloud data flows, consist of multiple stages of computations that process the input data and output the result. At each stage, the computation produces *intermediate* data that is to be processed by the next computing stage. We studied the characteristics of intermediate data in general, and we focused on the way it is handled in MapReduce frameworks. Our work addressed intermediate data at two levels: inside the same MapReduce job, and during the execution of pipeline applications.

We focused first on efficiently managing intermediate data generated between the “map” and “reduce” phases of MapReduce computations. In this context, we proposed to store the intermediate data in the distributed file system used as underlying backend. In this direction, we investigated the features of intermediate data in MapReduce computations and we proposed a new approach consisting in storing this kind of data in a DFS. The major benefit of this approach is better illustrated when considering failures. Existing MapReduce frameworks store intermediate data on nodes local disk. In case of failures, intermediate data produced by mappers can no longer be retrieved and processed further by reducers. the solution of most frameworks is to reschedule the failed tasks and to re-generate all the intermediate data that was lost because of failures. This solution is costly in terms of additional execution time. With our approach of storing intermediate data in a DFS, we avoid the re-execution of tasks in case of failures that lead to data loss. As storage for intermediate data, we considered BSFS as being a suitable candidate for providing for the requirements of intermediate data: availability and high I/O access. The tests we performed in this context, measured the impact of using a DFS as storage for intermediate data instead of the local-disk approach. We then assessed the performance of BSFS and HDFS when serving as storage for intermediate data produced by several MapReduce applications.

We then considered another type of intermediate data that appears in the context of *pipeline MapReduce applications*. In order to speed-up the execution of pipeline MapReduce applications (applications that consist of multiple jobs executed in a pipeline) and also, to improve cluster utilization, we proposed an optimized Hadoop MapReduce framework, in which the scheduling is done in a dynamic manner. We introduced several optimizations in the Hadoop MapReduce framework in order to improve its performance when executing pipelines. Our proposal consisted mainly in a new mechanism for creating tasks along the pipeline, as soon as the tasks’ input data becomes available. As our evaluation showed, this dynamic task scheduling leads to an improved performance of the framework, in terms of job completion time. In addition, our approach ensures a more efficient cluster utilizations, with respect to the amount of resources that are involved in the computation.

We evaluated both approaches for intermediate data through a set of experiments on the Grid’5000 [56] testbed. Preliminary results [17] show the scalability and efficiency of our proposals, as well as additional benefits brought forward by our approach.

6.1.3. A-Brain: Perform genetic and neuroimaging data analysis in Azure clouds

Participants: Radu Tudoran, Alexandru Costan, Gabriel Antoniu, Louis-Claude Canon.

Joint genetic and neuroimaging data analysis on large cohorts of subjects is a new approach used to assess and understand the variability that exists between individuals. This approach has remained poorly understood so far and brings forward very significant challenges, as progress in this field can open pioneering directions in biology and medicine. As both neuroimaging- and genetic-domain observations represent a huge amount of variables (of the order of 10^6), performing statistically rigorous analyses on such amounts of data represents a computational challenge that cannot be addressed with conventional computational techniques.

In order to perform an accurate analysis we need to provide a programming platform and a high throughput storage. The target infrastructure is the Azure clouds. Hence we have adapted the BlobSeer storage approach for Azure, thus providing a new way to store data in clouds, that federates the local storage space from computational nodes into a uniform shared storage, called TomusBlobs. Using this storage system as a storage backend, we have built a MapReduce prototype for Azure clouds. This MapReduce system, called TomusMapReduce -TMR, is used to perform the simulation of the joint genetic and neuroimaging application. For validating the framework, a toy application that simulate the data access and computation patterns of the real application, was used. The next step, after the evaluation of the framework, that has just began, consists in replacing the toy application with the real one and the scaling of the framework in the limit allowed by the cloud provider. In addition a demo for this project is in progress, that will consists in providing a visualization tool for the framework. This will be used to intuitively represent the results for the simulation of the scientific application, this being useful both for better presenting the project to interested parties and for the researchers from bioinformatics.

6.2. Efficient VM management in clouds

Participants: Alexandru Costan, Alexandra Carpen-Amarie, Gabriel Antoniu.

Infrastructure as a Service (IaaS) cloud computing allows users to lease computational resources from the cloud provider's datacenter for a short time by deploying virtual machines (VMs) on these resources. This model raises new challenges in the design and development of IaaS middleware. One of those challenges is the need to deploy a large number (hundreds or even thousands) of VM instances simultaneously. Once the VM instances are deployed, another challenge is to simultaneously take a snapshot of many images and transfer them to persistent storage to support fault tolerance and management tasks, such as suspend-resume and migration. With datacenters growing rapidly and configurations becoming heterogeneous, it is important to enable efficient concurrent deployment and snapshotting that are at the same time hypervisor independent and ensure a maximum compatibility with different configurations.

We addressed these challenges by proposing a virtual file system specifically optimized for virtual machine image storage [19]. It is based on a lazy transfer scheme coupled with object versioning that handles snapshotting transparently in a hypervisor-independent fashion, ensuring high portability for different configurations. Large-scale experiments on hundreds of nodes demonstrate excellent performance results: speedup for concurrent VM deployments ranges from a factor of 2 up to 25, with a reduction in bandwidth utilization of as much as 90 % [18]. We implemented this deployment scheme in the Nimbus cloud and presented a demo illustrating it at the Grid'5000 School [26].

Given the dynamic nature of IaaS clouds and the long runtime and resource utilization of scientific applications, an interesting use-case for the multi-snapshotting techniques is for efficient checkpoint-restart. We introduced an approach that leverages VM disk-image multi-snapshotting and multi-deployment inside checkpoint-restart protocols running at guest level in order to efficiently capture and potentially roll back the complete state of the application, including file system modifications. This framework is specifically optimized for tightly-coupled scientific applications that were written using a message passing system (in particular MPI) and need to be ported to IaaS clouds. Our solution introduces a dedicated checkpoint repository that is able to efficiently take incremental snapshots of the whole disk attached to the virtual machine instances, thus offering support to use any checkpointing protocol that can save the state of processes into files, including application level mechanisms, where the process state is managed by the application itself, and process-level mechanisms, where the process state is managed transparently at the level of the message passing library. Experiments on the G5K testbed show substantial improvement for MPI applications over existing approaches, both for the case when customized checkpointing is available at application level and the case when it needs to be handled at process level.

We integrated this checkpointing scheme inside the Nimbus cloud with some promising preliminary results. We plan to complement the existing solution with live incremental snapshotting using asynchronous background transfers for high checkpointing efficiency and with adaptive prefetching to achieve high restart efficiency.

6.3. Cloud data storage management

6.3.1. Autonomic storage for cloud services

Participants: Alexandru Costan, Alexandra Carpen-Amarie, Gabriel Antoniu, Florin Pop, Ciprian Dobre, Elena Apostol.

A means to achieve performance improvement and resource-usage optimization in cloud storage systems consists in enabling an autonomic behavior based on introspection. Self-adaptation incurs a high degree of complexity in the configuration and tuning of the system, with possible repercussions on its availability and reliability. To address these challenges we introduced in BlobSeer in [11] a three-layered architecture designed to identify and generate relevant information related to the state and the behavior of the system, based on the MonALISA monitoring framework. Such information is then expected to serve as an input to a higher-level self adaptation engine. These data are yielded by an (1) *introspection* layer, which processes the raw data collected by a (2) *monitoring* layer. The lowest layer is represented by the (3) *instrumentation* code that enables BlobSeer to send monitoring data to the upper layers.

A first approach to leverage the introspection framework aims at enhancing BlobSeer with *self-configuration* capabilities, as a means to support storage elasticity trough dynamic deployment of data providers. This solution enables the data providers to scale up and down depending on the detected system's needs. The component we designed adapts the storage system to the environment by contracting and expanding the pool of storage providers based on the system's load. The key idea of this component is the automatic decision that has to be made on how many resources the system needs to operate normally while keeping the resources utilization down to a minimum. This problem is addressed by using a test-decided heuristic based on the monitoring data. The introspective architecture has been evaluated on the Grid'5000 testbed, with experiments that prove the feasibility of generating relevant information related to the state and the behavior of the system.

We plan to use the introspective BlobSeer to develop a distributed data aggregation service. Its primary goal will be to serve as a repository backend for complex analysis and automatic mining of scientific data. Another direction that will be explored is to use the introspective BlobSeer as a cloud-based storage layer for sensitive context data, collected from a vast amount of sources: from smartphones to sensors located in the environment. Clouds are perfect candidates to handle the storage and aggregation of such data for even larger context-aware applications. Such solutions rely on more relaxed storage capabilities than traditional relational databases (eventual consistency suffices for example). This, combined with the high concurrency support and the flexible storage schema make BlobSeer a suitable candidate for the storage layer. We plan to develop a new layer on top on BlobSeer targeting context aware applications. At the logical level, this layer will provide transparency, mobility, real-time guarantees and access based on meta-information. At the physical layer, the most important capability will rely on BlobSeer's elasticity to scale up and down according to real-time usage, in order to reduce the costs within the Cloud.

6.3.2. Managing data access on Clouds through security policies

Participants: Alexandru Costan, Alexandra Carpen-Amarie, Gabriel Antoniu.

With the emergence of Cloud computing, there has been a great need to provide an adequate security level in such environments, as they are vulnerable to various attacks. Malicious behaviors such as Denial of Service attacks, especially when targeting large-scale data management systems, cannot be detected by typical authentication mechanisms and are responsible for drastically degrading the overall performance of such systems.

In [14] we proposed a generic security management framework allowing providers of Cloud data management systems to define and enforce complex security policies. The generality of this approach comes from the flexibility both in terms of supporting custom security scenarios and interfacing with different Cloud storage systems. This security framework is designed to detect and stop a large array of attacks defined through an expressive policy description language and to be easily interfaced with various data management systems. We introduced a modular architecture consisting of three components. The *Policy Management* module represents the core of the framework, where security policies definition and enforcement takes place. This module is completely independent of the Cloud system, as its input only consists in user activity events monitored from the system. The *User Activity History* module is a container for monitoring information describing users' actions. It collects data by employing monitoring mechanisms specific to each storage system and makes them available for the Policy Management module. The *Trust Management* module incorporates data about the state of the Cloud system and provides a trust value for each user based on his past actions. The trust value identifies a user as a fair or a malicious one. Furthermore, the trust values enable the system to take custom actions for each detected policy violation, by taking into account the history of each user.

As a case study, we applied the proposed framework to BlobSeer. We defined a specific policy to detect DoS attacks in BlobSeer and we evaluated the performance of our framework through large scale experiments on the Grid'5000 testbed.. The results show that the Policy Management module meets the requirements of a data storage system in a large-scale deployment: it was able to deal with a large number of simultaneous attacks and to restore and preserve the performance of the target system.

As a next step we will focus on more in-depth experiments involving the detection of various types of attacks in the same time. Moreover, we will investigate the limitations of our Security Management framework, with respect to the accuracy of the detection in the case of more complex policies, as well as the probability and the impact of obtaining false positive or false negative results. Another research direction is to further develop the Trust Management component of the security management framework and study the impact it has on the Policy Enforcement decisions for complex scenarios.

6.4. Storage architecture and adaptive consistency for clouds

Participants: Houssem-Eddine Chihoub, Gabriel Antoniu.

As more and more applications are becoming data-intensive, the design of a scalable storage architecture providing a huge file sharing and fine grain access with high throughput under heavy concurrency is a timely and relevant challenge.

In [24], we introduce a storage architecture for Cloud computing. This architecture proposes efficient and scalable storage support for both VM images and application data. Our architecture relies on BlobSeer [12], a data sharing platform optimized for concurrent accesses, as a basic storage backend enhanced in term of quality of service and efficiency by GloBeM tool [61] that rely on behavior modeling and monitoring to avoid bad case scenarios. The architecture uses this approach to have a better and efficient storage VM images, that allow a faster image deployment and efficient versioning and snapshotting. Furthermore, the architecture provides a platform to store, manage, and share cloud application data allowing several key features to clouds such as storage elasticity.

The main aim is to provide high availability and good scalable performance at low cost. This is justified by the growing need of data-intensive applications for managing huge sets of data replicated over several data centers. In order to provide good performance and high availability, data replication is mandatory. But this generates the issue of replicas consistency as shown by the CAP theorem [52]. To achieve the aforementioned goals, relaxing consistency rules is unavoidable. On the other hand, opting for weaker consistency, all the time, can be too costly.

In current work we leverage the trade-off between consistency and availability and performance. We are investigating an approach that changes the consistency level at runtime considering system and application needs. In order to choose the most suitable consistency level, our approach monitor the storage system and collect useful information, such as network load and applications access patterns, that enable the system to estimate the amount of expected stale reads.

6.5. Modelling cloud storage performance

Participants: Daniel Higuero, Louis-Claude Canon, Alexandru Costan, Gabriel Antoniu.

The objective of this research direction is to provide comprehensive performance models for storage systems. Their role is to capture how the system components interact for different usage patterns (number of reads or writes). The objective is to determine the incurred costs in terms of storage space and efficiency for a given workload.

One application of this model consists in dynamically adjusting the parameters of the storage system as required in an autonomic approach. For this purpose, it is necessary to identify the characteristics of the storage system for meeting a given level of requirements. Progress has been made on this part during the 3-month visit of Daniel Higuero (University Carlos III, Madrid). A preliminary performance model currently predicts the available bandwidth when multiple concurrent transfers occur. This model serves as a basis for a dimensioning strategy that is formulated through a linear program.

This approach has further been complemented with an offline analysis of several traces of the BlobSeer storage system when it is used as a backend for MapReduce applications. Mining this information in an automated fashion allowed to detect the different trade-offs that influence a BlobSeer deployment: time required to execute the application vs. number of machines used by the storage system, communication costs vs. space usage. The final goal is to tune BlobSeer for specific applications. The proposed strategy is currently being evaluated.

Future directions are directed towards refining the proposed model. Several parameters significantly impact the performance of storage systems such as the redundancy mechanism, the data placement strategy or disk-related effects. As a first step, experiments for assessing the quality of finer models will be designed. Ultimately, we aim to capture the I/O variability of storage systems, in particular in the context of the cloud.

This work will enable new collaborations. It is planned to work on the models mentioned above with the Mescal INRIA team in the context of a collaboration between the MapReduce ANR project and the Songs ANR project. Moreover, in the framework of the MapReduce project, we expect to work on a performance model for designing decision algorithms that are required by the component-based MapReduce framework that is developed in the GRAAL/Avalon INRIA team. Finally, the GRAAL/AVALON team works on scheduling algorithms that could beneficially profit from a storage performance model.

6.6. Scalable I/O and visualization for post-petascale HPC simulations

Participants: Matthieu Dorier, Gabriel Antoniu.

In the context of the Joint INRIA/UIUC Laboratory for Petascale computing (JLCP), we are addressing the new challenges related to I/O, data analysis and visualization for extreme-scale simulations. As HPC resources approaching millions of cores become a reality, a growing challenge in maintaining high performance is the presence of high variability in the effective throughput of codes performing I/O operations. Since I/O is mainly performed for the purpose of subsequent data analysis and visualization, another way to limit the impact of I/O performance on scientific discovery consists in enabling in-situ visualization. This brings again new challenges such as how to efficiently couple large-scale simulations with visualization software.

We started the development of Damaris, a middleware targeting multicore SMP nodes to efficiently address the problems mentioned above. Damaris has been evaluated with the CM1 atmospheric simulation [43], one of the targeted application for the BlueWaters project. To show the capability of our approach to efficiently hide I/O jitter and related costs, experiments have been carried on the French Grid'5000, on a Power5 cluster at NCSA and on the Kraken Cray XT5 supercomputer (currently 11th in the Top500) with up to 9K cores. By gathering data into large files while avoiding synchronization between processes, our solution brings several benefits:

1. it increases the sustained write throughput of the simulation by a factor of almost 15;
2. it provides almost 70 % overall application speedup on 9,000 cores;

3. it fully hides I/O-related costs;
4. it enables a 600 % compression ratio without any additional overhead, leading to a major reduction of storage requirements.

A poster [16] presenting some of these results has been accepted at ICS'11 and awarded the second price at the ACM Student Research Competition. All these results are presented in a research report [25] pending for publication.

Current work addresses the efficient coupling of large-scale simulations and visualization tools through Damaris. We have been able to get access to the Jaguar supercomputer hosted at ORNL and we are planning very-large scale experiments on up to 100,000 cores to show the benefits of our Damaris approach.

6.7. Scalable array-oriented active storage

Participants: Viet-Trung Tran, Gabriel Antoniu, Luc Bougé.

The recent explosion in data sizes manipulated by distributed scientific applications has prompted the need to develop specialized storage systems capable to deal with specific access patterns in a scalable fashion. In this context, a large class of applications focuses on parallel array processing: small parts of huge multi-dimensional arrays are concurrently accessed by a large number of clients, both for reading and writing. However, many established storage solutions such as parallel file systems and database management systems expose data access models (e.g., file systems, structured databases) that are too general and do not exactly match the nature requirements of the application. This forces the application developer to either adapt to the exposed data access model or to use an intermediate layer that performs a translation. In either case, the mismatch leads to suboptimal data management: the one-storage-solution-fits-all-needs has reached its limits.

Thus, there is an increasing need to specialize the I/O stack to match the requirements of the application. The objective of this research is to design Pyramid: an array-oriented active storage system optimizing for applications that represent and manipulate data as huge multi-dimensional arrays. However, a specialized storage system that deals with such an access pattern faces several challenges at the level of data/metadata management, we carefully design the system with the following principles: (1) we introduce a dedicated array-oriented data access model that offers support for active storage and versioning; (2) we enrich striping techniques specifically optimized for multi-dimensional arrays with a distributed metadata management scheme that avoids potential I/O bottlenecks observed with centralized approaches.

We evaluated Pyramid through a set of experiments on the Grid'5000 [56] testbed that aims to evaluate both the performance and the scalability of our approach under concurrent accesses. Preliminary evaluation in our recent papers [22], [13] shows promising results: our prototype demonstrates good performance and scalability under concurrency, both for read and write workloads.

7. Contracts and Grants with Industry

7.1. Contracts with Industry

Microsoft: A-Brain (2010–2013). In the framework of the Joint INRIA-Microsoft Research Center. See details in Section 4.1. To support this project, Microsoft provides 2 million computation hours on the Azure platform and 10 TB of storage per year. The project is funding Louis-Claude Canon as a postdoc fellow (18 months since September 2011) and to complete the PhD MESR grant of Radu Tudoran (*Mission complémentaire d'expertise*, 3 years, started in October 2011).

IBM: MapReduce ANR Project (2010–2014). IBM is a partner of the MapReduce ANR Project: see 8.2.

8. Partnerships and Cooperations

8.1. Regional Initiatives

Participant: Diana Moise.

The Brittany Regional Council provides half of the financial support for the PhD thesis of D. Moise (GRID5000BD project). This support amounts to a total of around 14,000 Euros/year. This support ended on September 30, 2011.

8.2. National Initiatives

MapReduce (2010–2014). An ANR project (ARPEGE 2010) with international partners on optimized Map-Reduce data processing on cloud platforms. This project started in October 2010 in collaboration with Argonne National Lab, the University of Illinois at Urbana Champaign, the UIUC/INRIA Joint Lab on Petascale Computing, IBM, IBCP, MEDIT and the GRAAL INRIA Project-Team. URL: http://mapreduce.inria.fr/

Grid'5000. We are members of the Grid'5000 community: we make experiments on the Grid'5000 platform on an everyday basis.

HEMERA (2010–2014). An INRIA Large Wingspan Project, started in 2010. Within Hemera, G. Antoniu (KerData INRIA Team) and Gilles Fedak (GRAAL INRIA Project-Team) co-lead the Map-Reduce scientific challenge. KerData also co-initiated a working group called “Efficient management of very large volumes of information for data-intensive applications”, co-led by G. Antoniu and Jean-Marc Pierson (IRIT, Toulouse).

EquipEx projects (submitted in 2011). We participated to the submission of two EquipEx projects in 2011: DIP-HPC in the HPC area (leader: GENCI; KerData stands for the INRIA partner); and VIRTEXP in the cloud area (leader: Christian Pérez, INRIA-GRAAL).

8.3. European Initiatives

The SCALUS FP7 Marie Curie Initial Training Network (2009–2013). Partners: Universidad Politécnica de Madrid (UPM), Barcelona Supercomputing Center, University of Paderborn, Ruprecht-Karls-Universität Heidelberg, Durham University, FORTH, École des Mines de Nantes, XLAB, CERN, NEC, Microsoft Research, Fujitsu, Sun Microsystems. Topic: scalable distributed storage. We mainly collaborate with UPM (2 co-advised PhD theses).

8.3.1. Major European Organizations with which you have followed Collaborations

CoreGRID ERCIM Working Group, since 2009. The CoreGRID Symposium held in Las Palmas de Gran Canaria, Spain, 25-26 August 2008 marked the end of the ERCIM-managed CoreGRID Network of Excellence funded by the European Commission. There, it was decided to re-launch CoreGRID as a self-sustained ERCIM Working Group covering research activities on both Grid and Service Computing while maintaining the momentum of the European collaboration on Grid research.

8.4. International Initiatives

F3PC: ANR-JST project (2010–2013) In this project we mainly collaborate with Tsukuba University, Japan (Gfarm Team). This project is a follow up to several previous collaborations: NEGST (2006–2009): CNRS-JST project. Bilateral PHC (ex-PAI) Sakura project (2006–2007).

8.4.1. INRIA Associate Teams

8.4.1.1. DataCloud@Work

Title: Distributed data management for cloud services

INRIA principal investigator: Gabriel Antoniu

International Partner:

Institution: Politehnica University of Bucharest (UPB, Romania)

Laboratory: Distributed Systems Software Laboratory, National Center for Information Technology (NCIT, http://cs.pub.ro/)

Researcher: Valentin Cristea, Professor at UPB

Duration: 2010–2012

See also: http://www.irisa.fr/kerdata/doku.php?id=cloud_at_work:start

Our research topics address the area of distributed data management for cloud services. We aim at investigating several open issues related to autonomic storage in the context of cloud services. The goal is explore how to build an efficient, secure and reliable storage IaaS for data-intensive distributed applications running in cloud environments by enabling an autonomic behavior, while leveraging the advantages of the grid operating system approach.

Our research activities involve the design and implementation of experimental prototypes based on the following software platforms:

The BlobSeer data-sharing platform (designed by the KerData Team)

The XtreemOS grid operation system (designed under the leadership of the Myriads Team)

The MonALISA monitoring framework (using the expertise of the PUB Team).

8.4.2. Visits of International Scientists

Bunjamin Memishi, Visiting PhD student Universidad Politecnica de Madrid (UPM), 1 month (April 2011), funded by Universidad Politecnica de Madrid through the SCALUS Marie-Curie Initial Training Network. His thesis is co-advised by Mariá Pérez (UPM) and Gabriel Antoniu (KerData).

Florin Pop, Visiting Postdoc Fellow Polytechnic University of Bucharest, 1 month (June 2011), funded by the DataCloud@work Associate Team.

Ciprian Dobre, Visiting Postdoc Fellow Polytechnic University of Bucharest, 1 month (June 2011), funded by the DataCloud@work Associate Team.

Elena Apostol, Visiting PhD student Polytechnic University of Bucharest, 3 months (June - August) 2011, funded by the DataCloud@work Associate Team.

Daniel Higuero, Visiting PhD student Carlos III University, Madrid, 3 months (September-November 2011), funded by Carlos III University, Madrid.

8.4.3. Participation In International Programs

Joint INRIA-UIUC Lab for Petascale Computing (JLPC), since 2009. Collaboration on concurrency-optimized I/O for post-Petascale platforms (see details in Section 4.1). A joint project proposal with the team of Rob Ross (Argonne National Lab) has been submitted in 2011 to the FACCTS call for projects (evaluation pending).

9. Dissemination

9.1. Animation of the scientific community

9.1.1. Leaderships, Steering Committees and community service

Euro-Par Conference Series. L. Bougé serves as a Vice-Chair of the *Steering Committee* of the *Euro-Par* annual conference series on parallel computing. G. Antoniu served as a Local Topic Chair for the *Parallel and Distributed Data Management* topic of *Euro-Par 2011*, held in Bordeaux.

IEEE CloudCom 2011 Conference. G. Antoniu served as a Track-Chair for the MapReduce track of the *IEEE CloudCom 2011* International Conference on Cloud Computing Technology and Science.

IEEE HPCS 2012 Conference. G. Antoniu serves as a Special Session and Workshop Chair for the 2012 International Conference on High Performance Computing and Simulation to be held in Madrid, Spain, in July 2012.

ACM HPDC 2012 Conference. G. Antoniu serves as a Publicity Co-Chair for the 21st International ACM Symposium on High-Performance Parallel and Distributed Computing to be held in Delft, the Netherlands in June 2012.

ScienceCloud 2012 Workshop. G. Antoniu serves as a Co-Chair for the ScienceCloud international workshop to be held in conjunction with the ACM HPDC 2012 international conference.

MapReduce ANR Project. G. Antoniu serves as a coordinator for the MapReduce ANR project (see Section 8.2).

A-Brain Microsoft-INRIA Project. G. Antoniu and B. Thirion (PARIETAL Project-Team, INRIA SACLAY – ÎLE-DE-FRANCE) co-lead the AzureBrain Microsoft-INRIA Project started in October 2010 in the framework of the Microsoft Research - INRIA Joint Center (2010-2013).

DataCloud@work Associate Team. G. Antoniu serves as a coordinator for the DataCloud@work Associate Team, a project involving the KerData and MYRIADS INRIA Teams in Rennes and the Distributed Systems Group from Politehnica University of Bucharest (2010–2012).

SCALUS Marie-Curie Initial Training Networks project. G. Antoniu coordinates the involvement of the INRIA RENNES – BRETAGNE ATLANTIQUE Research Center in the SCALUS Project of the Marie-Curie Initial Training Networks Programme (ITN), call FP7-PEOPLE-ITN-2008 (2009-2013).

CoreGRID ERCIM Working Group. G. Antoniu coordinates the involvement of the INRIA RENNES – BRETAGNE ATLANTIQUE Research Center in the CoreGRID ERCIM Working Group.

Agrégation of Mathematics. L. Bougé serves as a Vice-Chair of the National Selection Committee for High-School Mathematics Teachers, Informatics Track.

9.1.2. Editorial boards, direction of program committees

L. Bougé is a member of the *Editorial Advisory Board* of the *Scientific Programming* Journal.

9.1.3. Program Committees

G. Antoniu served in the Program Committees for the following conferences and workshops: ACM HPDC 2012, MAPREDUCE 2012 (workshop held in conjunction with HPDC 2012), AINA 2012, IEEE CloudCom 2011, Euro-Par 2011, AINA 2011, MAPREDUCE 2011 (workshop held in conjunction with HPDC 2011).

L. Bougé served in the Program Committee for the following conferences: NPC 2011, IEEE Cluster 2011.

9.1.4. Evaluation committees, consulting

L. Bougé was the chair of the national evaluation committee for the 2011 Scientific Excellence Award (*Prime d'excellence scientifique*, PES) targeted to the researchers on an academic teaching position in France (around 420 applications.)

G. Antoniu is a member of INRIA's Evaluation Committee.

G. Antoniu is a member of INRIA's Committee for Science and Technology Orientation - International Relations Working Group (COST-GTRI).

9.2. Invited talks

9.2.1. Keynotes and special invited talks

G. Antoniu was invited to give a keynote talk entitled *Scalable cloud storage for data-intensive applications: the BlobSeer approach* at the MapReduce workshop held in San José (USA) in June 2011 in conjunction with the ACM HPDC 2011 international conference.

G. Antoniu was invited to give an invited talk at Microsoft TechDays 2011 (with Pierre-Louis Xech and Thierry Priol) and was interviewed (with Thierry Priol) for a live Web TV broadcast on MS TechDays TV (Palais des Congrès, Paris, February 2011).

9.2.2. Invited talks at international workshops

G. Antoniu gave an invited talk entitled *BlobSeer: Efficient, Versioning-Based Storage for Massive Data under Heavy Access Concurrency on Clouds* at Microsoft Research, Cambridge in February 2011.

G. Antoniu gave an invited talk entitled *Update on Damaris: How to Make CM1 Scale Linearly up to 10K Cores And What Comes Next* at the 6th workshop of the Joint Laboratory for Petascale Computing held in November 2011 at NCSA/UIUC, Urbana-Champaign, IL, USA.

G. Antoniu gave a a talk entitled *A-Brain: Using the Cloud for Neuroimaging and Genetics Research* at the Microsoft Research - INRIA Joint Research Center in Saclay, December 2011.

9.3. Administrative responsibilities

G. Antoniu serves as the Scientific Correspondent for the International Relations Office of the INRIA RENNES – BRETAGNE ATLANTIQUE Research Center.

G. Antoniu serves as the Scientific Leader of the KerData research team.

L. Bougé chairs the Computer Science and Telecommunication Department (*Département Informatique et Télécommunications, DIT*) of the Brittany Extension of ENS CACHAN. He leads the Master Program (*Magistère*) in Computer Science at the Brittany Extension of ENS CACHAN.

9.4. Miscellaneous

L. Bougé is a member of Scientific Committee of INRIA RENNES – BRETAGNE ATLANTIQUE (*Comité des projets*), standing for the ENS CACHAN partner.

G. Antoniu is a member of Scientific Committee of INRIA RENNES – BRETAGNE ATLANTIQUE (*Comité des projets*), standing for the KerData research team.

9.5. Teaching

9.5.1. Academic courses

Gabriel Antoniu:

Master (Engineering Degree, 5th year): Grid and cloud computing, 18 hours (lectures), M2 level, Ecole Supérieure d'Informatique, Electronique, Automatique, Paris, France.

Master: Grid, P2P and cloud data management, 18 hours (lectures), M2 level, University of Nantes, ALMA Master, Distributed Architectures module, France.

Master: Peer-to-Peer Applications and Systems, 10 hours (lectures), M2 level, ENS Cachan - Brittany, M2RI Master Program, PAP Module, France.

Doctorat: MapReduce processing, 2 hours (lectures), SCALUS Spring School on Distributed Data Management held in Barcelona, Spain, February 2011.

Luc Bougé:

Licence *(Magistère)*: *Introduction to programming*, 36 hours, level L3, Magistère of Informatics and Telecommunication (MIT), ENS Cachan/Rennes, France

Licence *(Magistère)*: *Object-oriented programming in Java*, 18 hours, level L3, Magistère of Informatics and Telecommunication (MIT), ENS Cachan/Rennes, France

Licence *(Magistère)*:*Introduction to research*, 18 hours, L3, Magistère of Informatics and Telecommunication (MIT), ENS Cachan/Rennes, France

Master *(Magistère)*: *Object-oriented programming in C++*, 36 hours, level M1, Magistère of Mathematics and Applications, ENS Cachan/Rennes, France

9.5.2. HDR and PhD theses supervision

PhD: Alexandra Carpen-Amarie, BlobSeer *as a data-storage facility for Clouds: self-adaptation, integration, evaluation*, ENS Cachan/Rennes, defended on December 15, 2011, supervisors: Luc Bougé (30 %) and Gabriel Antoniu (70 %). Repository: [7].

PhD: Diana Moise, *Optimizing data management for MapReduce applications on large-scale distributed infrastructures*, ENS Cachan/Rennes, defended on December 16, 2011, supervisors: Luc Bougé (30 %) and Gabriel Antoniu (70 %). Repository: [8].

PhD in progress: Viet-Trung Tran, *Scalable array-oriented active storage*, ENS Cachan/Rennes, started October 2009, supervisors: Luc Bougé (30 %) and Gabriel Antoniu (70 %).

PhD in progress: Houssem-Eddine Chihoub, *Storage architecture and adaptive consistency for clouds*, ENS Cachan/Rennes, started October 2010, supervisors: Gabriel Antoniu (70 %) and Mariá Pérez (Universidad Politecnica de Madrid, 30 %).

PhD in progress: Bunjamin Memishi, *Fault-Tolerance in Scalable File Systems*, Universidad Politecnica de Madrid, started October 2010, supervisors: Gabriel Antoniu (30 %) and Mariá Pérez (Universidad Politecnica de Madrid, 70 %).

PhD in progress: Radu Tudoran, *AzureBrain: Perform genetic and neuroimaging data analysis in Azure clouds*, ENS Cachan/Rennes, Started October 2011, supervisors: Luc Bougé (30 %) and Gabriel Antoniu (70 %).

PhD in progress: Matthieu Dorier, *Scalable I/O and visualization for post-petascale HPC simulations*, ENS Cachan/Rennes, started October 2011, supervisors: Luc Bougé (30 %) and Gabriel Antoniu (70 %).

10. Bibliography

Major publications by the team in recent years

[1] G. Antoniu, L. Cudennec, M. Jan, M. Duigou. *Performance scalability of the JXTA P2P framework*, in "Proc. IEEE International Parallel and Distributed Processing Symposium (IPDPS 2007)", Long Beach, USA, 2007, 108, http://hal.inria.fr/inria-00178653/en/.

[2] G. Antoniu, J.-F. Deverge, S. Monnet. *How to bring together fault tolerance and data consistency to enable grid data sharing*, in "Concurrency and Computation: Practice and Experience", 2006, n^o 17, p. 1-19, http://hal.inria.fr/inria-00000987/en/.

[3] R. Morales, S. Monnet, I. Gupta, G. Antoniu. *MOve:Design and Evaluation of A Malleable Overlay for Group-Based Applications*, in "IEEE Transactions on Network and Service Management, Special Issue on Self-Management", 2007, vol. 4, p. 107-116 [*DOI :* 10.1109/TNSM.2007.070903], http://hal.inria.fr/inria-00446067/en/.

[4] B. NICOLAE, G. ANTONIU, L. BOUGÉ, D. MOISE, A. CARPEN-AMARIE. *BlobSeer: Next Generation Data Management for Large Scale Infrastructures*, in "Journal of Parallel and Distributed Computing", February 2011, vol. 71, n^{o} 2, p. 169-184, Special issue on data intensive computing. To appear, http://hal.inria.fr/inria-00511414/en/.

[5] B. NICOLAE, J. BRESNAHAN, K. KEAHEY, G. ANTONIU. *Going Back and Forth: Efficient Multi-Deployment and Multi-Snapshotting on Clouds*, in "The 20th International ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC 2011)", San José, CA, United States, June 2011, Selection rate: 12.9%., http://hal.inria.fr/inria-00570682/en.

[6] B. NICOLAE, D. MOISE, G. ANTONIU, L. BOUGÉ, M. DORIER. *BlobSeer: Bringing High Throughput under Heavy Concurrency to Hadoop Map-Reduce Applications*, in "24th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2010)", Atlanta, IEEE and ACM, Apr 2010, A preliminary version of this paper has been published as INRIA Research Report RR-7140., http://hal.inria.fr/inria-00456801.

Publications of the year

Doctoral Dissertations and Habilitation Theses

[7] A. CARPEN-AMARIE. *Utilisation de BlobSeer pour le stockage de données dans les Clouds: auto-adaptation, intégration, évaluation*, École normale supérieure de Cachan - ENS Cachan, December 2011, http://tel.archives-ouvertes.fr/tel-00653623/en/.

[8] D. MOISE. *Optimisation de la gestion des données pour les applications MapReduce sur des infrastructures distribuées à grande échelle*, École normale supérieure de Cachan - ENS Cachan, December 2011, http://tel.archives-ouvertes.fr/tel-00653622/en/.

Articles in International Peer-Reviewed Journal

[9] L. BOUGÉ, C. LENGAUER. *Preface for the special issue on Euro-Par 2009*, in "Concurrency and Computation: Practice and Experience", February 2011, vol. 23, n^{o} 2, p. 143-144 [*DOI :* 10.1002/CPE.1649], http://hal.inria.fr/inria-00555602/en.

[10] L. BOUGÉ, C. LENGAUER. *Preface for the special issue on Euro-Par 2010*, in "Concurrency and Computation: Practice and Experience", December 2011, vol. 23, n^{o} 17, p. 2137-2139 [*DOI :* 10.1002/CPE.1800], http://hal.inria.fr/hal-00654198/en/.

[11] A. CARPEN-AMARIE, A. COSTAN, J. CAI, G. ANTONIU, L. BOUGÉ. *Bringing Introspection into BlobSeer: Towards a Self-Adaptive Distributed Data Management System*, in "International Journal of Applied Mathematics & Computer Science", 2011, vol. 21, n^{o} 2, http://hal.inria.fr/inria-00555610/en.

[12] B. NICOLAE, G. ANTONIU, L. BOUGÉ, D. MOISE, A. CARPEN-AMARIE. *BlobSeer: Next Generation Data Management for Large Scale Infrastructures*, in "Journal of Parallel and Distributed Computing", February 2011, vol. 71, n^{o} 2, p. 168-184 [*DOI :* 10.1016/J.JPDC.2010.08.004], http://hal.inria.fr/inria-00511414/en.

[13] V.-T. TRAN, B. NICOLAE, G. ANTONIU. *Towards Scalable Array-Oriented Active Storage: the Pyramid Approach*, in "ACM Operating Systems Review (OSR)", 2012, Special issue on LADIS 2011, the 5th Workshop on Large Scale Distributed Systems and Middleware., http://hal.inria.fr/hal-00640900/en.

International Conferences with Proceedings

[14] C. Basescu, A. Carpen-Amarie, C. Leordeanu, A. Costan, G. Antoniu. *Managing Data Access on Clouds: A Generic Framework for Enforcing Security Policies*, in "The 25th International Conference on Advanced Information Networking and Applications (AINA-2011)", Singapore, Singapore, Institute for Infocomm Research (I2R) in cooperation with the Singapore Chapter of ACM, March 2011, http://hal.inria.fr/inria-00536603/en.

[15] A. Carpen-Amarie. *Towards a Self-Adaptive Data Management System for Cloud Environments*, in "IPDPS PhD Forum", Anchorage, United States, 2011, http://hal.inria.fr/inria-00575511/en.

[16] M. Dorier. *Poster: Damaris - Using Dedicated I/O Cores for Scalable Post-petascale HPC Simulations*, in "International Conference on Supercomputing (ICS)", Tucson, Arizona, USA, ACM, May 2011 [*DOI :* 10.1145/1995896.1995953], http://hal.inria.fr/hal-00639157/en/.

[17] D. Moise, T.-T.-L. Trieu, G. Antoniu, L. Bougé. *Optimizing Intermediate Data Management in MapReduce Computations*, in "CloudCP 2011 – 1st International Workshop on Cloud Computing Platforms, Held in conjunction with the ACM SIGOPS Eurosys 11 conference", Salzburg, Austria, 2011, http://hal.inria.fr/inria-00574351/en.

[18] B. Nicolae, J. Bresnahan, K. Keahey, G. Antoniu. *Going Back and Forth: Efficient Multi-Deployment and Multi-Snapshotting on Clouds*, in "The 20th International ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC 2011)", San José, CA, United States, June 2011, Selection rate: 12.9%., http://hal.inria.fr/inria-00570682/en.

[19] B. Nicolae, F. Cappello, G. Antoniu. *Optimizing multi-deployment on clouds by means of self-adaptive prefetching*, in "Euro-Par '11: Proc. 17th International Euro-Par Conference on Parallel Processing", Bordeaux, France, February 2011, p. 503-513 [*DOI :* 10.1007/978-3-642-23400-2_46], http://hal.inria.fr/inria-00594406/en.

[20] S. Orlando, G. Antoniu, A. Ghoting, M. S. Pérez-Hernández. *Introduction*, in "Euro-Par (1)", Springer, 2011, p. 351-352.

[21] V.-T. Tran, B. Nicolae, G. Antoniu, L. Bougé. *Efficient support for MPI-IO atomicity based on versioning*, in "11th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid 2011)", Newport Beach, CA, United States, IEEE CS Press, 2011, A preliminary version was published as INRIA Research Report RR-7487, http://hal.inria.fr/inria-00565358/en.

[22] V.-T. Tran, B. Nicolae, G. Antoniu, L. Bougé. *Pyramid: A large-scale array-oriented active storage system*, in "LADIS 2011: The 5th Workshop on Large Scale Distributed Systems and Middleware", Seattle, United States, September 2011, http://hal.inria.fr/inria-00627665/en.

[23] V.-T. Tran. *Towards a storage backend optimized for atomic MPI-I/O for parallel scientific applications*, in "Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW)", Anchorage, Alaska, United States, May 2011, http://hal.inria.fr/inria-00627667/en.

Conferences without Proceedings

[24] H.-E. Chihoub. *Towards a scalable, fault-tolerant, self-adaptive storage for the clouds*, in "EuroSys'11 Doctoral Workshop", Salzburg, Austria, April 2011, http://hal.inria.fr/inria-00637766/en.

Research Reports

[25] M. Dorier, G. Antoniu, F. Cappello, M. Snir, L. Orf. *Damaris: Leveraging Multicore Parallelism to Mask I/O Jitter*, INRIA, August 2011, n° RR-7706, http://hal.inria.fr/inria-00614597/en.

Other Publications

[26] A. Carpen-Amarie, D. Moise, B. Nicolae. *Leveraging BlobSeer to boost up the deployment and execution of Hadoop applications in Nimbus cloud environments on Grid'5000*, 2011, https://www.grid5000.fr/school2011/Grid5000-2011-Challenge1.pdf.

[27] R. Tudoran. *Optimizing data storage for MapReduce applications in the Azure Clouds*, École Normale Supérieure de Cachan/Rennes, Rennes, June 2011, http://hal.inria.fr/hal-00643336/en.

References in notes

[28] *Amazon Elastic MapReduce*, http://aws.amazon.com/elasticmapreduce/.

[29] *Chirp protocol specification*, http://www.cs.wisc.edu/condor/chirp/.

[30] *European Exascale Software Initiative*, http://www.eesi-project.eu.

[31] *Google App Engine*, http://code.google.com/appengine/.

[32] *Google Docs*, http://www.google.com/google-d-s/tour1.html.

[33] *HadoopFS*, 2009, http://hadoop.apache.org/hdfs/docs/current/.

[34] *International Exascale Software Program*, http://www.exascale.org/iesp/Main_Page.

[35] *Lightweight Data Replicator*, http://www.lsc-group.phys.uwm.edu/LDR/.

[36] *Microsoft Azure*, 2009, http://www.microsoft.com/azure/.

[37] *Microsoft Office Live*, 2009, http://www.officelive.com/.

[38] *The Nimbus project*, 2009, http://workspace.globus.org/.

[39] *OpenNebula*, 2010, http://www.opennebula.org/.

[40] B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, S. Tuecke. *Data management and transfer in high-performance computational grid environments*, in "Parallel Comput.", 2002, vol. 28, n° 5, p. 749–771, http://dx.doi.org/10.1016/S0167-8191(02)00094-7.

[41] A. BASSI, M. BECK, G. FAGG, T. MOORE, J. S. PLANK, M. SWANY, R. WOLSKI. *The Internet Backplane Protocol: A Study in Resource Sharing*, in "Proc. 2nd IEEE/ACM Intl. Symp. on Cluster Computing and the Grid (CCGRID '02)", Washington, DC, USA, IEEE Computer Society, 2002, 194.

[42] J. BENT, V. VENKATARAMANI, N. LEROY, A. ROY, J. STANLEY, A. ARPACI-DUSSEAU, R. ARPACI-DUSSEAU, M. LIVNY. *Flexibility, Manageability, and Performance in a Grid Storage Appliance*, in "Proc. 11th IEEE Symposium on High Performance Distributed Computing (HPDC 11)", 2002.

[43] G. H. BRYAN, J. M. FRITSCH. *A Benchmark Simulation for Moist Nonhydrostatic Numerical Models*, in "Monthly Weather Review", 2002, vol. 130, n^{o} 12, p. 2917–2928 [*DOI :* 10.1175/1520-0493(2002)130<2917:ABSFMN>2.0.CO;2], http://journals.ametsoc.org/doi/abs/10.1175/1520-0493%282002%29130%3C2917%3AABSFMN%3E2.0.CO%3B2.

[44] R. BUYYA, C. S. YEO, S. VENUGOPAL. *Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities*, in "HPCC '08: Proceedings of the 2008 10th IEEE International Conference on High Performance Computing and Communications", Washington, DC, USA, IEEE Computer Society, 2008, p. 5–13, http://dx.doi.org/10.1109/HPCC.2008.172.

[45] P. H. CARNS, W. B. LIGON, R. B. ROSS, R. THAKUR. *PVFS: A Parallel File System for Linux Clusters*, in "ALS '00: Proceedings of the 4th Annual Linux Showcase and Conference", Atlanta, GA, USA, USENIX Association, 2000, p. 317–327.

[46] M. A. CASEY, F. KURTH. *Large data methods for multimedia*, in "Proc. 15th Intl. Conf. on Multimedia (Multimedia '07)", New York, NY, USA, ACM, 2007, p. 6–7, http://doi.acm.org/10.1145/1291233.1291238.

[47] J. DEAN, S. GHEMAWAT. *MapReduce: simplified data processing on large clusters*, in "Communications of the ACM", 2008, vol. 51, n^{o} 1, p. 107–113.

[48] A. DEVULAPALLI, D. DALESSANDRO, P. WYCKOFF, N. ALI, P. SADAYAPPAN. *Integrating parallel file systems with object-based storage devices*, in "SC '07: Proceedings of the 2007 ACM/IEEE conference on Supercomputing", New York, NY, USA, ACM, 2007, p. 1–10, http://dx.doi.org/10.1145/1362622.1362659.

[49] K. DOUGLAS, S. DOUGLAS. *PostgreSQL*, New Riders Publishing, Thousand Oaks, CA, USA, 2003.

[50] M. FACTOR, K. METH, D. NAOR, O. RODEH, J. SATRAN. *Object storage: the future building block for storage systems*, in "Local to Global Data Interoperability - Challenges and Technologies, 2005", 2005, p. 119–123, http://dx.doi.org/10.1109/LGDI.2005.1612479.

[51] S. GHEMAWAT, H. GOBIOFF, S.-T. LEUNG. *The Google file system*, in "SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles", New York, NY, USA, ACM Press, 2003, p. 29–43, http://dx.doi.org/10.1145/945445.945450.

[52] S. GILBERT, N. LYNCH. *Brewer's Conjecture and the Feasibility of Consistent Available Partition-Tolerant Web Services*, in "In ACM SIGACT News", 2002, 2002.

[53] S. GRIMES. *Unstructured Data and the 80 Percent Rule*, 2008, Carabridge Bridgepoints.

[54] M. IBRAHIM, R. ANTHONY, T. EYMANN, A. TALEB-BENDIAB, L. GRUENWALD. *Exploring Adaptation & Self-Adaptation in Autonomic Computing Systems*, in "Database and Expert Systems Applications, International Workshop on", 2006, vol. 0, p. 129-138, http://dx.doi.org/10.1109/DEXA.2006.57.

[55] R. JIN, G. YANG. *Shared Memory Parallelization of Data Mining Algorithms: Techniques, Programming Interface, and Performance*, in "IEEE Trans. on Knowl. and Data Eng.", 2005, vol. 17, n^{o} 1, p. 71–89, http://dx.doi.org/10.1109/TKDE.2005.18.

[56] Y. JÉGOU, S. LANTÉRI, J. LEDUC, M. NOREDINE, G. MORNET, R. NAMYST, P. PRIMET, B. QUETIER, O. RICHARD, E.-G. TALBI, T. IRÉA. *Grid'5000: a large scale and highly reconfigurable experimental Grid testbed.*, in "International Journal of High Performance Computing Applications", November 2006, vol. 20, n^{o} 4, p. 481-494.

[57] K. KEAHEY, T. FREEMAN. *Science Clouds: Early Experiences in Cloud Computing for Scientific Applications*, in "Cloud Computing and Its Applications 2008 (CCA-08)", Chicago, IL, 2008.

[58] J. O. KEPHART, D. M. CHESS. *The Vision of Autonomic Computing*, in "Computer", 2003, vol. 36, n^{o} 1, p. 41–50, http://dx.doi.org/10.1109/MC.2003.1160055.

[59] A. LENK, M. KLEMS, J. NIMIS, S. TAI, T. SANDHOLM. *What's inside the Cloud? An architectural map of the Cloud landscape*, in "Software Engineering Challenges of Cloud Computing (CLOUD '09)", 2009, p. 23 - 31, ICSE Workshop.

[60] M. MESNIER, G. R. GANGER, E. RIEDEL. *Object-based storage*, in "Communications Magazine, IEEE", 2003, vol. 41, n^{o} 8, p. 84–90, http://dx.doi.org/10.1109/MCOM.2003.1222722.

[61] J. MONTES, B. NICOLAE, G. ANTONIU, A. SÁNCHEZ, M. S. PÉREZ-HERNÁNDEZ. *Using Global Behavior Modeling to Improve QoS in Cloud Data Storage Services*, in "CloudCom '10: Proc. 2nd IEEE International Conference on Cloud Computing Technology and Science", Indianapolis, United States, October 2010.

[62] M. NICOLA, M. JARKE. *Performance Modeling of Distributed and Replicated Databases*, in "IEEE Trans. on Knowl. and Data Eng.", 2000, vol. 12, n^{o} 4, p. 645–672, http://dx.doi.org/10.1109/69.868912.

[63] C. OLSTON, B. REED, U. SRIVASTAVA, R. KUMAR, A. TOMKINS. *Pig latin: a not-so-foreign language for data processing*, in "SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data", New York, NY, USA, ACM, 2008, p. 1099–1110, http://doi.acm.org/10.1145/1376616.1376726.

[64] M. PARASHAR, S. HARIRI. *Autonomic computing: An overview*, in "Unconventional Programming Paradigms", Springer Verlag, 2005, p. 247–259.

[65] A. RAGHUVEER, M. JINDAL, M. F. MOKBEL, B. DEBNATH, D. DU. *Towards efficient search on unstructured data: an intelligent-storage approach*, in "CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management", New York, NY, USA, ACM, 2007, p. 951–954, http://doi.acm.org/10.1145/1321440.1321583.

[66] P. SCHWAN. *Lustre: Building a file system for 1000-node clusters*, in "Proceedings of the Linux Symposium", 2003, http://citeseer.ist.psu.edu/schwan03lustre.html.

[67] A. THOMASIAN. *Concurrency control: methods, performance, and analysis*, in "ACM Computing Survey", 1998, vol. 30, n^{o} 1, p. 70–119, http://doi.acm.org/10.1145/274440.274443.

[68] L. M. VAQUERO, L. RODERO-MERINO, J. CACERES, M. LINDNER. *A break in the clouds: towards a cloud definition*, in "SIGCOMM Comput. Commun. Rev.", 2009, vol. 39, n^{o} 1, p. 50–55, http://doi.acm.org/10.1145/1496091.1496100.

[69] S. A. WEIL, S. A. BRANDT, E. L. MILLER, D. D. E. LONG, C. MALTZAHN. *Ceph: a scalable, high-performance distributed file system*, in "OSDI '06: Proceedings of the 7th symposium on Operating systems design and implementation", Berkeley, CA, USA, USENIX Association, 2006, p. 307–320, http://portal.acm.org/citation.cfm?id=1298455.1298485.