Project no. 034567

Grid4All

Specific Targeted Research Project (STREP)
Thematic Priority 2: Information Society Technologies

Grid4All White Paper

Towards Hybrid Clouds —
A Grid4All perspective on cloud computing

Due date of deliverable: June 2009.

Actual submission date: 5 August 2009.

Start date of project: 1 June 2006 | Duration: 37 months

Organisation name of lead contractor for this deliverable: Grid4All consortium

Revision: 1

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)

Dissemination Level

PU	Public	√
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Contents

Abbreviations and acronyms used in this document

Abbrev./acronym	Def.	Description
ACID		Atomic, Consistent, Isolated, Durable (the properties of a transaction)
CRDT	4.5	Commutative Replicated Data Type
SIS	4.3	Grid4All Semantic Information System
POSIX		Portable Operating System Interface for Unix (IEEE Standard 1003.1)
SLA		Service-Level Agreement
VOFS	4.4	Grid4All Virtual-Organisation-Aware File System
VO		Virtual Organisation

Grid4All list of participants

Role	Part. #	Participant name	Part. short name	Country
CO	1	France Telecom	FT	FR
CR	2	Institut National de Recherche en Informatique en Automatique	INRIA	FR
CR	3	The Royal Institute of technology	KTH	SWE
CR	4	Swedish Institute of Computer Science	SICS	SWE
CR	5	Institute of Communication and Computer Systems	ICCS	GR
CR	6	University of Piraeus Research Center	UPRC	GR
CR	7	Universitat Politècnica de Catalunya	UPC	ES
CR	8	ANTARES Produccion & Distribution S.L.	ANTARES	ES

Executive summary

Clouds promise seemingly infinite computation and storage capacity, billed on a utility model, accessible through simple Web interfaces. However, current cloud architectures are proprietary, ad-hoc, ensure vendor lock-in, deprive users of the control of their own information, and offer very stripped-down services.

We predict that users will demand a *Hybrid Cloud*, i.e., the clean and principled integration of the users' own computational resources with upcoming peer-to-peer *edge clouds* and existing vendor clouds. They will also demand new levels of service, such as an autonomic management framework, consistency guarantees for their data, or support for co-operation through the cloud. Finally, they will demand the ability to freely select the resources they use, according to various market and pricing models (from buying to rental to altruistic exchange).

The Grid4All *Democratic Grid* project aims to enable users from different organisations to share resources and data and to co-operate, to access utility resources over the network, and to set up autonomic management of these facilities. Our results are powerful technological enablers for edge and hybrid clouds. Five achievements are particularly relevant: *(i)* Autonomic management tools; *(ii)* A Resource Market for trading resources; *(iii)* Discovery of computational resources according to semantic properties; *(iv)* A file system enabling collaboration; *(v)* A collaboration-oriented middleware that supports disconnected operation and detects and repairs conflicts.

1 Introduction

The Grid4All FP6 project was started in 2006 with the goal of a "democratic grid." It aims to provide new classes of users with flexible access to large computational capacities, to enable them to pool their computational resources over the network, and to empower them with sharing and collaboration facilities.

Since then, Cloud Computing has become all the rage. Providers such as Amazon, Google or eBay offer vast amounts of CPU, storage and networking capacities. They are available over the Internet, through very simple interfaces, at small incremental cost. Users can outsource complex tasks, such as system administration, or failure detection and recovery. So, has cloud computing achieved our goal? Has Grid4All been rendered obsolete?

Indeed, beyond the hype, something substantial is happening. Clouds allow easy access to plentiful resources, in line with the Grid4All objectives. Cloud computing may be technically quite similar to grid computing, but there is also something more; for instance, an order-of-magnitude (at least) increase in scale, which creates many new problems.

However, this not the whole story. Current cloud architectures are based on big data centres owned by a single vendor and provide stripped-down functionality (e.g., mostly stateless services). This introduces a single organisational point of failure, restricts the kind of applications that are supported, causes vendor lock-in, and deprives users of control over their own data.

We believe that technological and economic forces will cause these barriers to be shattered, in the space of a few years. Smaller, more specialised providers will appear. Users will want higher levels of service (for instance, consistency guarantees for their data), will need to access a multiplicity of providers. Some classes of users will prefer to swap resources with one another rather than pay or trust a provider, except possibly for occasional needs.

Thus, the future points to a "hybrid cloud," the seamless integration of a number of providers of various sizes (including the users themselves), offering a vast, dynamic array of services and resources, under largely variable conditions and SLAs. The hybrid cloud will be large in scale and subject to highly variable latencies, churn rates, and frequent disconnection and failure events. Consistent sharing of stateful data will be a requirement.

These are precisely the focus areas of the Grid4All project. Grid4All enables open, democratic access to computing resources, supporting both grid and edge resources, and supports co-operation within distributed groups of users. As a research project, Grid4All focused on advanced technological enablers: peer-to-peer based autonomic management facilities, resource market, and federative data storage and access (and ignored other topics that are either well established or out of its scope). We argue that the Grid4All achievements are enablers for future hybrid clouds.

In this white paper, we will examine the Grid4All research agenda in light of cloud computing. In most cases, our results resonate with cloud computing requirements; in others, our approach is at odds, by design, with current cloud computing practice.

Outline of the paper is as follows. In section 2 we briefly describe cloud computing, the key differences between clouds and Grids, and the pros and cons of clouds. We end this section by analyzing trends and argue for the upcoming emergence of hybrid clouds. In section 3 we review our project, briefly describe the original project objectives, and reevaluate both our objectives and results in the light of the emergence of cloud computing. In section 4 we cover in detail the main achievements of the Grid4All project, and their relevance to hybrid clouds, and, for some, their relevance to current data-centre clouds. In section 5 we conclude.

2 Cloud computing background

2.1 Cloud computing today: isolated, proprietary clouds

Cloud computing promises easy access to scalable, virtualised resources, billed on a utility model, over a simple internet connection. Resources are diverse, and may include computing power, storage capacity, data, or software. Current clouds are proprietary. They are owned and managed by a commercial provider (e.g., Amazon or IBM), who manages the infrastructure; the user cannot control it and may ignore operational details.

Different levels of functionality are offered, such as Infrastructure-as-a-service (IaaS – CPU, network and disk), Platform-as-a-Service (PaaS – development), Software-as-a-Service (SaaS – applications), etc. Clouds bring existing distributed computing concepts to a very large scale by relying on cluster architectures, virtualisation, highly automated management, and extensive failure detection and recovery.

From the users' perspective, advantages of cloud computing include: paying only for resources effectively used; low barrier to entry; outsourced management; immediate access to virtually unlimited resources; reliability and availability; simple environments.

As for providers, they benefit from a consolidated infrastructure and shared costs, which help create new services and capture more customers. Although some providers currently specialise in renting infrastructure (e.g., IBM, HP, Sun, Dell, Akamai, or VMWare), and others in specific services (e.g., Amazon, Google, Yahoo, Microsoft, Salesforce, or Netsuite), these two models will likely soon converge.

2.2 Technical perspective, pros and cons

Cloud computing has many characteristics in common with previous styles of distributed computing, such as grids or autonomic and utility computing. Citing Ian Foster's blog [7]:

> *So is "cloud computing" just a new name for grid? [. . .] Yes: the vision is the same — to reduce the cost of computing, increase reliability, and increase flexibility by transforming computers [. . . in]to something that is operated by a third party. But no: things are different now than they were 10 years ago. [. . .] We find that those clusters are darn expensive to operate. We have low-cost virtualisation. And, above all, [. . .] we're operating at a different scale, and operating at these new, more massive scales can demand fundamentally different approaches to tackling problems. [. . .] Nevertheless, yes: the problems are mostly the same in cloud and grid. [. . .] Unfortunately, [. . .] the methods used to achieve these goals in today's commercial clouds have not been open and general purpose.*

Today's proprietary clouds, based on technologies such as virtualisation, stripped-down large-scale file systems (such as the Google File System) and data-flow computing paradigms (Map/Reduce), are distributed systems of unprecedented scale. However, the traditional trade-offs of distributed computing don't go away. These clouds trade off scalability for decreased openness and flexibility.

On the technical side, proprietary clouds have a number of drawbacks. Users lose control over their own applications and their own data, which threatens data privacy (and may harm performance). An organisational single point of failure (the connection to the remote data centre) is introduced. Although the services offered are very efficient within their narrow design parameters, these are limited and closed. Finally, proprietary clouds offer very little in terms of guarantees (e.g., consistency guarantees), being based on a "best-effort" design.

Reliance on clouds also has economic disadvantages. Although clouds offer low incremental costs, for predictable and constant needs it is generally cheaper to own than to rent. Stallman argues that proprietary clouds force their customers into locked-in systems that will cost more and more over time [9].

Current technological trends [1] is that networking costs are not decreasing as fast as cpus or storage, which indicates that data locality will become progressively more important. This may change the economics for organizations that generate massive amounts of data, move them into the cloud, analyze them, and then retrieve the results.

2.3 Future trends

Current clouds are proprietary and centralised in large data centres. This has some advantages, but it is not the only possible model. Many efforts are underway to build so-called volunteer clouds, or edge clouds, building on P2P techniques to federate available edge computing resources in a cooperative, tit-for-tat manner.

Researchers in distributed systems are already busy working on increasing scalability even further, while finding remedies to the drawbacks listed above. In particular, we expect future clouds to be considerably more open and less limited in functionality. Standardised interfaces will allow integration of the user's own infrastructure, proprietary service providers, and pooled resources, forming a *hybrid cloud*. Such hybrid clouds will combine edge resources and proprietary clouds, as well as bridging data-centre clouds with one another. In addition to the current single-user applications, the hybrid cloud will offer support for collaboration, data consistency, forming VOs that can aggregate all kinds of resources, support for general-purpose application management, more user control, etc.

On these crucial points, Grid4All proposes a number of contributions, surveyed in the following sections.

3 Grid4All objectives and results

The Grid4All project was set up to enable open, democratic access to computing resources, supporting both grid and edge resources, and to support co-operation within distributed groups of users, so-called "virtual organisations."

The name Grid4All, and its slogan "democratic Grid," highlight its focus on supporting small users, having limited financial, physical, and system administration resources. To these users, important factors are openness, the ability to utilise all available resources (including volatile ones), ease of use and self-management.

3.1 Grid4All, Grids and Clouds

The Grid4All project was formulated almost 4 years ago. At the time, grids represented the state of the art in general-purpose distributed systems, and had some of the features required for democratic scenarios, such as sharing and aggregation of resources, and forming *Virtual Organisations* (VOs) across organisational boundaries. Our vision was naturally expressed as a generalisation of grids, a movement from Grid-For-Some to Grid4All. Today, an appropriate title might be *Hybrid, Democratic Cloud*.

Recent cloud developments confirm the importance our three design principles, namely *scalability, churn tolerance and decentralisation*. Thus, according to Birman *et al.* [3], an important lesson learned from cloud computing is that "scalability emerges as cross-cutting concern affecting all building blocks used in cloud settings." Scalability figures prominently in our driving scenarios, because, even if each user provides or consumes only a small amount of resources to a VO, there may be a very large number of them. (The analogy is that, although each citizen has but one vote, democracies can be very large).

Churn tolerance is essential, even data-centre clouds, because scaling up implies a higher aggregate fault rate. Says Birman: "Cloud services must be designed under the assumption that they experience frequent and often unpredictable failures [and must be able to] recover from failures autonomously." Accordingly, Grid4All technology (e.g., Niche or VOFS) is based on peer-to-peer techniques.

Birman also teaches us that cloud providers have found it necessary to use very loosely coupled, asynchronous solutions, such as non-ACID databases and decentralised convergence mechanisms. This has also been the Grid4All approach with the decentralised "semantic" store Telex.

It is interesting to note the converging requirements of smaller systems running over volatile resources and large-scale data centres. The former are open, subject to large latencies, and uncontrolled. The latter are closed, have high-speed networks internally, and are well controlled; however, at large scale and high computational volumes, the aggregate failure rate increases and the network is a bottleneck. Thus, scalability, decentralisation and autonomic churn tolerance are equally essential in both cases. Although Grid4All never targeted data centres, it could be that some of its results are relevant even to data-centre clouds.

3.2 Project Achievements

As a research project, Grid4All focused on advanced technological enablers: autonomic management facilities, resource market, and federative data storage and access. It developed and evaluated a number of basic building blocks in the form of prototypes. It also built proof-of-concept demonstrators, while minimising non-essential engineering effort (e.g., not all parts are integrated, and in some places we used centralised implementations).

The Grid4All results are enablers for hybrid clouds. In a world that is large-scale, dynamic and multi-party, people and organisations will want to collaborate, cooperate and interact with others, sharing, pooling their resources in ventures that may be short-lived or long-lived. In this context, the Grid4All achievements can be divided into two themes.

1. Pooling resources and utilising them effectively:
    - *Niche and Management Tools:* A hybrid cloud is composed of a very large and variable number of resources, subject to unpredictable events such as failures, leaves and joins. At this scale, it is impossible to manage services, applications or resources individually or manually. The Grid4All management tools provide a self-managing infrastructure and a general-purpose programmable self-management framework, to connect and utilise effectively dynamic resources.
    - *Resource Market:* A hybrid cloud needs some mechanism for users to trade resources. The Grid4All Resource Market supports buying, selling and bartering resources, individually or in aggregate, with maximum flexibility.
    - *Semantic Discovery:* A mechanism for finding computational resources according to their semantic properties. As VOs are dynamic, and may be set up in arbitrary ways, navigating resources would be very difficult without such a tool.

2. Collaboration and sharing data:
   - *VOFS:* To enable collaboration, a minimal requirement is the ability to set up groups of collaborating users. Each such group needs a workspace for pooling resources, for federating their shared files and documents, for naming them, and security facilities to selectively protect or expose information. Grid4All developed the *Virtual-Organisation-Aware File System* (VOFS) to this effect.
   - *Telex:* Effective collaboration requires a higher level of service than just a file system. The Grid4All *Telex Semantic Store* deals with replicating data, enables disconnected operation and tolerates churn, and detects and repairs update conflicts, ensuring eventual agreement on a correct version of every shared document.

The five Grid4All achievements will be discussed in detail in the next section. We do believe that these building blocks with further polishing and integration, engineered together with other state-of-the-art mechanisms like virtualisation, policy engines, higher-level management tools, and so on would be a considerable step to the vision of hybrid clouds.

4 Project Achievements

4.1 Niche and Management Tools

Niche [13] is a general-purpose distributed component management system (DCMS) used to develop, deploy and execute self-managing distributed applications or services in all kinds of environments, including very dynamic ones with volatile resources. Niche is both a component-based programming model that includes management aspects as well as a VO-wide distributed run-time environment.

Application management in a distributed setting consists of two parts. First, there is the initial deployment and configuration, where individual components are shipped, deployed, and initialized at suitable nodes, then the components are bound to each other as dictated by the application architecture, and the application can start working. Second, there is dynamic reconfiguration where a running application needs to be reconfigured. This is usually due to environmental changes, such as change of load, the state of other applications sharing the same infrastructure, node failure, node leave (either owner rescinding the sharing of his resource, or controlled shutdown), but might also be due to software errors or policy changes. All the tasks in initial configuration occur also in dynamic reconfiguration. For instance, increasing the number of nodes in a given tier will involve discovering suitable resources, deploying and initializing components on those resources and binding them appropriately. However, dynamic reconfiguration generally involves more, because firstly, the application is running and disruption must be kept to a minimum, and secondly, management must be able to manipulate running components and existing bindings.

The advantages of self-managing applications are fairly obvious (see, for instance, the IBM Autonomic Computing Initiative), and today there is a considerable body of work in the area, most of it geared to clusters. Niche builds on this work and is Java-based and Fractal-based, where Fractal defines the management interfaces of components. The control loop paradigm is used where management logic in a continuous feedback loop senses changes in the environment and component status, reasons about those changes, and then, when needed, actuates, i.e. manipulates components and their bindings. A self-managing application can be divided into a functional part and a management part tied together by sensing and actuation.

Non-standard and novel features in Niche: Niche is a VO-wide infrastructure that loosely binds all physical resources, and provides for resource discovery by using a structured overlay. Niche provides a sensing service for application managers, as part of its programming API. Application managers can subscribe to application-specific component events (e.g. load) or platform events (e.g. failure). Niche provides an actuation service, whereby managers can control and manipulate with components and their bindings. The sensing and actuation services are robust and churn-tolerant, and Niche itself is self-managing.

Niche allows for maximum decentralization of management. Management can be divided (i.e. parallelized) by aspects (e.g. self-healing, self-tuning), spatially, and hierarchically. A single application, in general, has many loosely synchronized managers. Niche also provides the execution platform for these managers; they typically get assigned to different machines in the VO. There is some support for optimizing this placement of managers, and some support for replication of managers for fault-tolerance.

An important feature is that all elements of the architecture, components, managers, etc., have VO-wide identifiers. Niche ensures that actuation messages reach the component and sensing events reach (at least once) the subscribed manager, even if the component or manager has moved.

Although programming in Niche is on the level of Java, it is both possible and desirable to program management at a higher level (e.g. declaratively). We have done some work here. Initial configurations can be described in high-level ADL (Architecture Description Language) which is compiled to Niche code. Fpr dynamic reconfiguration, we have developed DepOz that supports navigating and querying component structures and defining reconfiguration workflows in a concise and compositional way. DepOz directly supports all common workflow patterns and is extendable. DepOz is not yet integrated with Niche.

Rationale of Approach: In the Grid4All vision non-professionals should be able to form a VO to collaborate and share their resources. In that VO, they should be able to install, deploy, use and manage non-trivial distributed applications with minimal effort. This requires that the applications are self-managing. Furthermore, in these environments, resources are at their most volatile, and therefore standard mechanisms to achieve self-management will not work. A single management node that continuously monitors the entire system and keeps an up-to-date system map introduces not only a single-point-of-failure but more importantly a bottleneck. The more dynamic and volatile the system is the more environmental events will be generated. Not only does Niche support decentralized management, but the sensing/actuation infrastructure has features to reduce messaging. For instance, a manager that is responsible for self-healing does not need to be informed about component movement due to, for instance, a node shut-down. Indeed, in a dynamic Grid4All VO, system components might be reshuffled in the continuously changing resource pool for some time before a failure occurs and the self-healing manager is triggered. Also, the Fractal model was extended to include component groups with one-to-all, one-to-any bindings for maximum efficiency.

Relevance to Clouds: We believe that Niche and the principals of how it was built, are extremely relevant for hybrid clouds, as well as democratic grids, as most of the issues are similar. In particular, it makes economic sense for users to integrate their own infrastructure with a cloud, generally running their own infrastructure at full capacity and using the absolute minimum of cloud resources for peak and excess demand. This introduces the kind of volatility with cloud resources being grabbed upon need and then being released as soon as possible, that Niche was designed to handle.

Fortuitously, our work might also be relevant for data-centre clouds. In developing Niche, our guiding principles were churn-tolerance, decentralization, and, in principle, scalability. (Though as we were not targeting large scale systems there are non-scalable aspects in the Niche implementation, but these are engineering issues). According to Birman et. al. [3] an important lesson learned by the cloud computing community is the importance of churn-tolerance and decentralization (scalability too, but that is given). Unfortunately, as was also pointed out in the article, it is virtually impossible for the research community to test mechanisms in really large scale systems (hopefully, this will be addressed in the European FIRE initiative [8]). Niche and our two application demonstrators (YASS and YACS [13]) do illustrate churn-tolerance and decentralization of management in small scale systems. We do not know what would happen on the scale of clouds. For instance, Birman also points out that in principle self-stabilizing mechanisms can exhibit unacceptable oscillations when scaled up.

4.2 Resource Market

Market places allow buyers and sellers to carry out trade using well-defined rules whose ultimate effect is to adjust prices. By adjusting prices, supply and demand are balanced. We provide tools to support building resource market places, focusing on two aspects [10]: *(a)* the allocation and pricing problem, and *(b)* scalable dissemination of price signals to shape participants' decisions. Together they provide a way for decentralised coordination and self-organising resource allocation in democratic grids. *Combinatorial auctions* offer a competitive solution to the complex resource co-allocation and planning problem; but they are computationally complex and hard to interpret.

By supporting spatial and temporal partitioning of auctions, the market place will scale with number of participants. Interpretation of market state can be improved through suitable pricing schemes. Partitioning and decentralisation should not cause loss of information leading to inefficient allocation and price instability. Accurate price signals aid buyers and sellers to plan capacity, e.g., lease resource at times that maximise their gain or reorganise internal workload and sell capacity when demand is high. We address these through commodity pricing schemes and dissemination of approximate aggregated information.

Cloud computing, allowing consumers to adapt to surges by leasing more resources when demand spikes occur, is now an oligopoly held by a few companies, which practise opaque pricing schemes. In a democratic cloud, the supply side should be democratised too. Any resource owner should also be able to sell its excess capacity. This encourages competition and innovation. Thus, the Grid4All Resource Market is an enabler for new business models, where entities that act as brokers of computing resources, will emerge. They will provide value-added services, SLA enforcement and risk management, capacity planning tools and reputation-aware market mechanisms.

4.3 Semantic Discovery

In the hybrid cloud, finding the right resource will be a challenge. This will be complex, since the hybrid cloud is a large-scale, unstructured system offering a large number of providers of various sizes and specialisations. They will offer a vast, dynamic array of services and resources under largely differing conditions and SLAs. Furthermore, any user's demand may dynamically expand or shrink, depending, for instance, on the demands of specific applications. This requires an effective matchmaking service between requests and offers for services and resources. Indeed, such a service is is already a vital requirement in grids.

The mutual discovery of clients and providers is an essential feature of the open market of the hybrid cloud, where resources are traded as goods and subject to supply and demand. Resources are made available through spontaneous agent-initiated market services. Resource matchmaking generalises to the problem of discovering those markets, each with its own auction mechanism.

The Grid4All experience suggests a declarative, semantic approach. Semantic descriptions have the potential to support automation of service retrieval, invocation, composition and monitoring tasks by providing machine-exploitable, meaningful declarative descriptions of service characteristics. In this context, the Grid4All Semantic Information System (G4A-SIS) provides:

- A semantic registry for offers and requests of services and resources in a market-oriented, democratised grid environment.
- (Semi-)automatic semantic annotation of web services, easing their semantic description and their registration in the semantic registry. This makes the best of the available information for the benefit of service providers, consumers and developers.
- Semantic matchmaking mechanisms between offers and requests for services and resources.
- A selection mechanism that ranks markets and services according to client preferences. This is crucial to maximising the satisfaction of providers (sellers) and consumers (buyers).

Summing up the above, aiming to operate at a large scale with multiple providers that have varying resources' and services' provision abilities, accentuating the role of the supply and demand law, hybrid clouds need effective matchmaking services between offers and requests for services/resources.

The Grid4All Semantic Information System provides a first step towards effective match-making in a large-scale system subject to dynamic offers and dynamic demand. For reasons of expedience, the current implementation is centralised. One of the major lessons learned is that, in order to remain effective in such a large-scale, dynamic environment, the registry must be decentralised. A distributed version of G4A-SIS has been designed to achieve this goal.

4.4 Virtual -Organisation-aware File System (VOFS)

Sharing files is a basic task of a co-operating group of users. According to our "democratic" vision, setting this up should be a casual operation and should not require any central authority or infrastructure. To that end, we designed VOFS, a virtual-organisation oriented file system [6].

VOFS provides each user with their own file system to share in the network. Users are able to federate views of their respective file systems, and to create workspaces of shared storage space and file content. Federating is straightforward and remains decentralised in the style of the world-wide web: users simply set up links to identify shared resources.

For applications, VOFS is transparent, in the sense that files are accessed via the standard POSIX interfaces. Additionally, applications may exploit the additional VOFS features, such as federation, caching, disconnected operation or notification services and generic messaging.

Users remain in strong control over their resources. The file systems of different users are independent; federation of files in shared workspaces does not compromise this. The access control architecture is flexible, as it allows users to program their own access policies. For instance, a policy might refer to a central VO security infrastructure.

VOFS is designed to maintain functional independence between users, despite failures of the network or of remote computers. Caching is used to hide network latency, to provide disconnected access, and more generally, to avoid disruption from network events. This makes the file system very responsive, and is especially valuable in loosely-coupled collaborative environments. By design, concurrent access to the same file are not synchronised, and a best-effort approach is taken. When stronger guarantees are required, applications should use the Telex Semantic Store (Section 4.5).

While users are independent from authorities and are able to create their own workspaces around their workflows, third parties may also enter the network to offer their services, by launching a VOFS peer to represent them. The peer appears as another regular peer to the users. The service provider may extend VOFS to install their own logic behind serving files or storage without creating any disruption or dependency to other users.

For example, a library might export a corpus of articles to a group of collaborating researchers, in the form of a VOFS file system. Furthermore, the library may extend the file system with "virtual paths" that instantiate library queries. For instance, virtual directory `/search/author/Turing` would contain all documents whose author is Turing. Applications access it just like any other directory. No single peer is critical: if the peer hosting the library crashes, the only disruption is the temporary loss of the library service; other services or files are not affected.

The resulting environment combines casual and independent user collaboration with potential access to big, organised third party services in the cloud.

Although cloud products such as `dropbox.com` or `box.net` provide very simple ways of sharing information, or `wuala.com` leverage social networking, users cannot create their workflows beyond what is given by the service provider. In contrast, our approach is inspired by the casual, unco-ordinated character of the World-Wide Web, where large corporate providers co-exist with individual users, and the latter retain their freedom and control. Like the Web, VOFS democratically empowers users; but where the Web only supports posting content to unlimited readers, VOFS supports true co-operation inside defined groups.

4.5 Telex Semantic Store

To support effective co-operative group work requires well-defined consistency guarantees, beyond the best-effort approach of VOFS. The Telex Semantic Store addresses this issue. Telex also addresses the larger problem of providing both scalability and well-defined consistency guarantees over shared mutable data. This will enable novel applications, that require consistency guarantees to operate over cloud-scale infrastructures.

The Telex middleware layer ensures that different network nodes may independently update their local replica of shared data, without prior synchronisation. This supports disconnected operation and co-operative work, and helps general scalability. Each user may have a different tentative view of the state of shared data; Telex provides well-defined guarantees that programs may rely upon: replicas eventually converge, and application invariants are never violated, despite concurrency, disconnection and reconciliation [2]. Example applications include a consistent, peer-to-peer, shared calendar system, Sakura and a co-operative Wiki editing system [5].

To scale up, clouds must avoid strong consistency mechanisms and promote decoupling [4]. Indeed, the Telex consistency mechanisms are parametrised with the semantics of shared data items, and ensure scalability by relaxing consistency mechanisms and moving them to the background.

A follow-up of this research is the design of Commutative Replicated Data Types (CRDTs), i.e., data types that support completely decoupled updates and arbitrary divergence yet ensure eventual consistency without any concurrency control [12]. We conducted a preliminary study of techniques supporting CRDTs in cloud and edge-computing scenarios, with large, dynamic numbers of sites [11].

5 Conclusion

We predict that, in the future, users will demand a Hybrid Cloud, i.e., the clean and principled integration of the users' own computational resources with upcoming peer-to-peer edge clouds and existing data-centre clouds. They will also demand new levels of service, such as autonomic management framework, consistency guarantees for their data, or support for co-operation through the cloud. Finally, they will demand the ability to freely select the resources they use, according to various market and pricing models (from buying to rental to altruistic exchange), and having done so, integrate them for maximum utilization.

The five main achievements of the Grid4All project, are all in line with this vision. Niche is a programming environement and a runtime application management platform designed for systems with volatile resources, whether due to resource churn or user-directed actions such as moving running applications, wholly or partly, to cheaper resources. The resource market provides hybrid clouds with a mechanism to support buying, selling or bartering of resources. Semantic discovery enables effective navigation in VOs that are dynamic and heterogenous. The VOFS enable collaboration by providing a means for sharing and federating files and documents, with security facilities to selectively protect and expose data. Telex offers consistency guarantees, beyond best-effort, but without strong coupling.

Recent developments, in Cloud computing and elsewhere, only confirm the importance of our three main guiding priciples, scalability, decentralization and churn-tolerance. We saw these principles as the key to effectively support dynamic VOs with volatile resources, and today they are acknowledged as the key for large data-centre clouds as well.

References

[1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, U. of California, Berkeley, Berkeley, CA, USA, February 2009.

[2] Lamia Benmouffok, Jean-Michel Busca, Joan Manuel Marquès, Marc Shapiro, Pierre Sutra, and Georgios Tsoukalas. Telex: A semantic platform for cooperative application development. In *Conférence Française de Systèmes d'Exploitation (CFSE)*, Toulouse, France, September 2009.

[3] Ken Birman, Gregory Chockler, and Robbert van Renesse. Toward a cloud computing research agenda. In *ACM Sigact News*, pages 68–80. ACM, June 2009. Report on Workshop on Large Scale Distributed Systems (LADIS 2008).

[4] Ken Birman, Gregory Chockler, and Robbert van Renesse. Toward a Cloud Computing research agenda. *ACM SIGACT News*, 40(2):68–80, June 2009.

[5] Gérôme Canals, Pascal Molli, Julien Maire, Stéphane Laurière, Esther Pacitti, and Mounir Tlili. XWiki Concerto: A P2P wiki system supporting disconnected work. In *Cooperative Design, Visualization, and Engineering (CDVE)*, volume 5220 of *lncs*, pages 98–106, Calvià, Mallorca, Spain, September 2008. Springer.

[6] Antony Chazapis, Georgios Tsoukalas, Georgios Verigakis, Kornilios Kourtis, Aristidis Sotiropoulos, and Nectarios Koziris. Global-scale peer-to-peer file services with DFS. In *Internation Conference on Grid Computing*, pages 251–258, Austin, TX, USA, September 2007. IEEE/ACM.

[7] Ian Foster. There's Grid in them thar Clouds. Blog http://ianfoster.typepad.com/blog/2008/01/theres-grid-in.html, January 2008.

[8] Objective 1.6 ICT Work Programme. Future internet research & experimentation, June 2009. See http://cordis.europa.eu/fp7/ict/fire/.

[9] Bobbie Johnson. Cloud computing is a trap, warns GNU founder Richard Stallman. *The Guardian*, September 2008.

[10] Ruby Krishnaswamy, Leandro Navarro, René Brunner, Xavier León, and Xavier Vilajosana. Grid4All: Open market places for democratic grids. In *International Workshop on Grid Economics and Business Models (GECON)*, volume 5206 of *lncs*, pages 197–207, Las Palmas de Gran Canaria, August 2008. Springer.

[11] Mihai Leția, Nuno Preguiça, and Marc Shapiro. CRDTs: Consistency without concurrency control. Rapport de recherche RR-6956, Institut National de la Recherche en Informatique et Automatique, Rocquencourt, France, June 2009. Sumbitted for publication.

[12] Nuno Preguiça, Joan Manuel Marquès, Marc Shapiro, and Mihai Leția. A commutative replicated data type for cooperative editing. In *Int. Conf. on Distributed Comp. Sys. (ICDCS)*, pages 395–403, Montréal, Canada, June 2009.

[13] SICS/KTH. Niche, a distributed component management system, June 2009. Available at http://niche.sics.se/.