Project no. 004758

GORDA

Open Replication of Databases

Specific Targeted Research Project

Software and Services

Dissemination and Exploitaion Plan

GORDA Deliverable D6.5

Due date of deliverable: 2007/07/01
Actual submission date: 2007/10/01
Resubmission date: 2008/05/25

Start date of project: 1 October 2004 Duration: 42 Months

UMINHO

Revision 1.1

Project co-funded by the European Commission within the Sixth Framework		
Dissemination Level		
PU	Public	X
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)	

Contributors

Fernando Pedone, U. Lugano
José Pereira, U. Minho
Luís Rodrigues, U. Lisboa
Luís Soares, U. Minho
Robert Hodges, Continuent
Nuno Carvalho, U. Lisboa
Rui Oliveira, U. Minho
Sara Bouchenak, INRIA

Abstract

This report presents the main dissemination and exploitation activities carried during the GORDA project and plans for future exploitation.

Contents

Chapter 1

Introduction

The GORDA project joined four academic partners with strong expertise on database management systems, dependable distributed systems and autonomic management, and an industrial partner prominent on the database clustering market.

While the project's goals, research and development strategies were, in the essential, shared by all partners, the relevance of the various results and the opportunities to exploit them varied and vary among the partners.

For academic partners advancing the state-of-the-art in strong consistent database replication, strengthen their position among the research communities, build proof-of-concept prototypes to support pedagogical and consultancy activities were the main goals. As detailed in the following sections, these were entirely achieved. More than 50 research papers have been published, partners organized and participated in several highly reputed scientific meetings, gave invited talks in several universities, developed system prototypes and reinforced their universities' courses with database replication topics.

Continuent, GORDA's industrial partner, joined the project as one of the very few companies commercially exploiting mission-critical database clustering and with the vision of the need for a standard DBMS interface allowing efficient strong consistent database replication. Continuent's offering was pretty limited in October 2004 (it was exclusively applicable to MySQL databases, supported a single active replication protocol, and was monolithically tightened to a specific total order multicast protocol). The company's main goal was to rapidly enlarge its commercial offer by diversifying its applicability and judiciously incorporating proven solutions from the distributed systems communities. As detailed later on, Continuent sports today a richer product line and has clear plans for the continued integration of GORDA results.

In the following, each partner presents the main dissemination and exploitation activities carried during the project's lifespan and plans for future exploitation.

Chapter 2

UMINHO

During the project UMinho greatly reinforced its team expertise on fundamental and practical aspects of distributed database management systems, database replication protocols, simulation and benchmarking. The knowledge and experience amassed in the large code-base developed during the period of the project constitutes an immensely competitive business position that can be exploited by all the involved. Three PhD. students that have been key to UMinho's contribution will be graduating soon.

A major investment near the distributed systems and databases communities on database replication challenges and benefits resulted in a very important asset that promises continued leveraging.

UMINHO carried several initiatives in order to disseminate and apply GORDA results and experience. At the university itself, database replication topics, in particular its practical aspects, has been emphasized in the current master course of computer science and a dedicated course is being offered in UMinho's doctoral programme.

Recent contacts and collaborations with the national industry have been triggered by the work and the vision of GORDA. Currently, two projects are in place to evaluate the requirements and to propose dependable database solutions to a large real-estate and automotive online database, and a telecommunications fraud detection system.

Near our academic and industrial peers we carried several initiatives with the following objectives: a) make researchers and developers aware of the GORDA Architecture and API relevance, completeness and usefulness, and b) bring together a heterogeneous but consistent community around the broad topic of "Dependable Distributed Data Management". These initiatives comprehended several invited talks in highly reputed universities: U. Nantes, CWI, U. P. Navarra, B. U. Hong Kong, Tokyo Institute of Technology, U.F Bahia, and demonstrations in companies: SUN Microsystems, Telbit Tecnologias de Informação, Lda. and Multivector - TI - Tecnologias de Informação S.A.. In the context of GORDA, UMINHO has organized

two workshops (in VLDB 2005 and Eurosys 2008) and, co-organized with other IST FP6 projects, three tracks in the ACM Symposium on Applied Computing in 2006, 2007 and 2008,[1] and tutorials on Database Replication and Clustering in SAC 2008 and DSN 2008.

Finally, two major actions are planned for the near future: a spin-off of Universidade do Minho, Dependableware S.A., exclusively dedicated to research and development of Dependable Data Systems is planned for the beginning of 2009; a proposal for networking and community consolidation in forthcoming opportunities, most likely FP7 Call 4 or a COST action.

[1] A fourth one has already been accepted for the 2009 edition.

Chapter 3

USI

The main exploitable results for USI from the GORDA project are the suite of protocols for database replication in both local- and wide-area networks. Database replication is an active area of research at USI and the expertise acquired by USI researchers while working on this project and interacting with the other academic and industrial partners has significantly contributed to advance our research agenda. One PhD student deeply involved in GORDA is expected to graduate soon.

The activities in the GORDA project helped to shape a Master course in Computer Systems at USI. The practical perspective provided by the project led to a complete course, involving both theory and practice. The Master-level course was offered to about 20 students in the area of Computer Systems. Part of the course covered advanced topics in database systems. This part greatly benefited the activities of USI researchers in the GORDA project.

We intend to offer other courses on Computer Systems and Distributed Systems as part of our Master and PhD programs. These courses will use part of the results obtained from the GORDA project.

USI will be organizing a Summer School on Dependable Computer Systems together with the School of Computer Science, Carnegie Mellon University (US). Some of the modules offered as part of this school will cover distributed systems and database systems and will benefit from the knowledge that researchers at USI acquired as part of the GORDA project.

Chapter 4

FCUL

FCUL identifies several exploitable results developed during the project, namely with respect to group communication services and the compliance of the Apache Derby DBMS to the GORDA API. Group communication has been for long the main area of research of the FCUL team and GORDA resulted in advancing the state-of-art in self-adaptive and large scale total order protocols as well as consolidating the implementation of the Appia communication toolkit and promoting its usability by means of the definition of a generic interface to group communication services, jGCS.

From a research point of view, the system produced by the GORDA project can serve as a testbed to new communication and replication protocols. New protocols can be implemented in the Appia and ESCADA toolkits and tested in a real environment, avoiding the use of simulators.

The FCUL team is currently working on a new protocol that exploits both technologies of atomic broadcast and quorum systems to optimize read accesses while maintaining the linearizability consistency model. This protocol has been accepted for publication as a short paper and is being tested in the GORDA framework, with the TPC-W benchmark. The modularity of the GORDA framework allows to build new database replication protocols and reuse the several components such as the Communication Service to disseminate transaction information and the GAPI reflection interfaces to extract write sets from the database engine.

Parts of the GORDA framework are already being used in FCUL and Technical University of Lisbon in new research projects. The Pastramy (Persistent and highly Available Software TRansactional MemorY) project aims to augment the basic Software Transactional Memory (STM) model with persistence, to provide ACID properties to web application functionalities, and replication for high availability and throughput. The STM system that will be improved is being used in a web based multi-tier application that handles the logic of the Technical University of Lisbon. Not just parts of the GORDA framework, but also the knowledge on replication protocols pro-

duced in this project, will be used to feed the Pastramy project. Finally, the Peervibes project will reuse the Appia protocol composition framework to build and compose new Peer-to-Peer based communication systems.

Chapter 5

INRIA

A first exploitable result for INRIA from the GORDA project consists in an autonomic management framework for the administration of distributed systems. This framework proposes a novel approach that allows to tackle legacy (i.e. black-box) database systems as well as open GORDA-compliant database systems. This framework builds on the Jade prototype, and integrates it to the GORDA Management Service, in cooperation with replication protocols of the GORDA Replication Service and communication protocols of the GORDA Communication Service. Another exploitable result for INRIA from the GORDA project is the set of management policies, such as self-recovery and self-optimization, that were implemented and deployed on real distributed database systems. The novelty here is twofold: first, proposing a self-self-management approach and second, proposing new models that allow to calculate the optimal configuration of a distributed system. With the self-self-management approach, the management system is itself self-managed (e.g. self-recovered) through reification. While the proposed new models allow to calculate the optimal configuration of a distributed system, e.g. a configuration that provides best performance with minimal resource consumption. Finally, GORDA provides INRIA with interesting distributed system case studies for self-management, where distributed systems may be legacy or GORDA-compliant database systems, including different replication protocols and different communication protocols. Two PhD students involved in the GORDA project are expected to graduate soon. Several initiatives were carried to present and demonstrate GORDA results, such as at the INRIA's 40 years anniversary celebration in December 2007 (*http://www.inria.fr/40ans/*), the "Forum 4i" in April 2007 (*http://www.forum4i.fr/*), and at "Fête de la Science" in 2006 (*http://www.fetedelascience.fr/*). The research activities in the GORDA project helped us to enrich the "Distributed Systems" Master's course with recent state-of-the-art results in self-management of distributed systems and replicated database systems. This also applies to the international Mas-

ter of Computer Science at Grenoble that will open in September 2008 (*http://mosig.imag.fr/*), in which several professors from INRIA are involved. Finally, INRIA's research activities in the context of GORDA result in several publications, among which the most recent ones will appear in the Springer Encyclopedia of Database Systems.

Chapter 6

Continuent

Overview

Continuent became a member of the GORDA project because as a company we shared the vision proposed by GORDA of a clustering architecture capable of accommodating multiple clustering approaches. In addition, GORDA is based on the insight that replication is key to all clustering designs. This is also a viewpoint that we share very strongly and that we have validated through experience with hundreds of customers and prospects.

By contrast, most previous work in on scale-out clustering has focused on particular solutions such as certification models or state-machine based replication, each of which addresses a limited set of use cases. The lack of generality has prevented any one of these individual approaches from providing a dominating solution. We thus believe that the GORDA perspective is unique in the research community and is an important intellectual achievement. It is a good basis for a new generation of database availability and performance solutions.

We briefly explain Continuent's plans for utilizing GORDA project deliverables in commercial offerings for clustering. We start with a brief overview and background for commercialization, then describe Continuent's current offerings and the company's exploitation Roadmap through the integration and commercialization of GORDA work products.

Current Product Overview

This section provides a brief description of the current and future commercial products of Continuent. Continuent specializes in software to raise database availability and throughput. Our solutions have consistently been based on shared-nothing approaches in which a set of database nodes linked together across a network function as a cluster that appears to be a single database to applications.

m/cluster

Continuent's original clustering product was an in-core database cluster using group communication for replication between nodes. The theoretical underpinnings of this model were based on the concept of extended virtual synchrony, which was first described and applied to replication by Yair Amir PhD. thesis. To our knowledge Continuent developed the first commercial cluster based on this design.

The m/cluster product implements eager replication by modifying the MySQL lock manager directly. This design requires delicate changes at the source code level, which is difficult for open source databases and impossible for commercial databases. m/cluster offered excellent read scaling and a unique update capability that allowed users to make schema changes with minimal downtime. The product is in use in a number of sites across the world, the largest of which is the US National Weather Service.

One of the key lessons of this effort was the need to adopt solutions that did not require alteration of the database, especially delicate internals like the lock manager. The difficulty of maintaining the approach across MySQL versions, let alone porting to new databases, led Continuent to seek less intrusive solutions.

uni/cluster

Continuent's current product offering is uni/cluster, which uses middleware to create clusters using unaltered, off-the-shelf databases. Uni/cluster is built on the Sequoia open source clustering project, which in turn evolved from C-JDBC, a project first developed at INRIA in Grenoble, France. The core technology is based on the state-machine replication approach, which was actively investigated in the late 1990s.

Uni/cluster is based on a set of open source technologies that are owned or sponsored by Continuent and comprise a stack for database scale-out. The following table summarizes some of the most important stack components released to-date.

Component	**Description**
Sequoia	State-machine based middleware replication, load balancing, and failover
Myosotis	Native wire-protocol proxying for MySQL and PostgreSQL
Hedera	Generic adapters for group communications
Bristlecone	Testing tools and benchmarks for scale-out
Duocomm	Optimized group communications (commercial only)
Management Tools	Cluster management tools (commercial only)

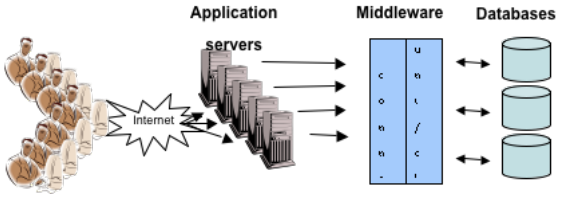

Figure 6.1: uni/cluster Architecture.

Uni/cluster adds a number of features and extensions to the open source stack in order to create a product that is stable and performant for commercial use. These include improved group communications, upgraded support for SQL dialects of MySQL and PostgreSQL, and better management tools. The uni/cluster architecture is illustrated below.

The uni/cluster architecture has a number of strengths that are characteristic of middleware approaches. Middleware offers particularly good support for failover and load balancing, since the logic for these is hidden from applications, which do not know which database(s) they are actually using. This "virtualization" capability is a key tool for building large clusters, and Continuent has implemented in such a way that both MySQL and PostgreSQL clients can connect without library or code changes.

On the other hand, the middleware approach is only suitable for certain types of applications. The biggest problem we have encountered is that middleware replication is not as quick or as flexible as other replication designs. For example, middleware replication does not work across WAN links, as communications failures require databases to be resynchronized. Given that such communications failures are common in many environments, other replication approaches are needed.

Tungsten

Tungsten is Continuent's latest design effort for clustering. Tungsten builds on the lessons of the previous efforts, including the need for off-board solutions as well as the need to allow different replication approaches.

Much like GORDA, Tungsten is a middleware framework that enables the use of a variety of replication methods. Tungsten extends the scale-out stack created by uni/cluster to add distributed management and new forms of replication. It also strengthens the middleware virtualization capabilities of Sequoia and the Myosotis connector (used to provide access for native client applications).

The main improvement to Myosotis is to provide clear application semantics that trade-off traditional database properties such as read consistency against improved performance and/or availability. This idea has been im-

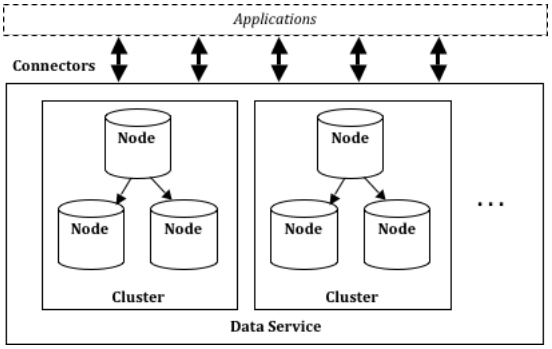

Figure 6.2: Tungsten Architecture.

plemented in an ad-hoc fashion on open source databases, such as MySQL. It is a goal of Tungsten to make such trade-offs explicit and easy to manage.

The following schematic diagram illustrates the Tungsten architecture in use.

The Tungsten architecture creates "data services", which are distributed databases composed of multiple underlying databases. Data may be partitioned into separate clusters. The connectors maintain consistent application semantics when working across copies of data. For example, a connector offers policies to applications that determine when it is safe to load balance operations onto replicas.

The Tungsten architecture shares the design approach of GORDA in that multiple replication mechanisms are supported within a single framework. For high transparency as well as WAN support, database replication is the best approach. For building local clusters with applications are willing to use a subset of SQL operations, the middleware state-machine approach is a good choice. Finally, the architecture is designed to integrate certification-based approaches when these become available.

Exploitation Roadmap

Continuent engineering efforts are focused on extending the basic technology stack for scale-out and using that stack to construct innovative new products for users. We have identified a number of locations in which the GORDA deliverables appears to be a good fit within the stack and offers interesting commercial possibilities.

Exploitation Philosophy

GORDA work products can be integrated into Continuent products in at least three different ways.

Direct integration. In some cases deliverables can be integrated as is. This is particularly true for the performance testing tools (D5.3), which were developed with the GORDA architecture in mind but with the goal of immediate reuse.

Re-implementation and reuse. In this case we consider the GORDA work to be a prototype and re-implement with a greater or lesser amount of recoding to ensure robustness and full compatibility with other features. For example, this is the best approach for re-using the GORDA reflector interfaces in Sequoia.

Reuse of designs and architecture. In this case we reuse concepts with a new implementation. This is effectively the case with the Tungsten architecture, which follows the overall approach of GORDA to provide a very generic data service that can handle different replication types.

The following sections provide additional details on specific areas of exploitation.

Overall Technology Stack Design

GORDA's key contribution as mentioned at the beginning of this document is the definition of an overall architecture that integrates different replication approaches within a single framework. Our own commercial experience as well as work on specific GORDA deliverables such as performance testing, plus examination of the reflector interfaces implemented in Sequoia and PostgreSQL convince us that this idea is a profound insight. It is a fundamental organizing principle that allows us to build a broad-based stack for database scale-out.

The Continuent technology stack will increasingly conform to the GORDA approach in which a generic framework supports a variety of clustering implementations. For instance, Tungsten Connector services can provide access to clusters composed of nodes that are linked by different types of replication mechanisms. This implies that the connector will implement different sets of policies that take into account the consistency semantics of each model. For example, primary/backup models do not guarantee read consistency between replicas, whereas certification approaches do. The Connector implementation much therefore be extensible to support both.

Other components of the stack will apply this same notion, particularly the management and testing frameworks.

GORDA Reflector Interfaces in Sequoia

GORDA reflector interfaces are currently checked into the Sequoia codeline as part of Sequoia 3.0. The GAPI implementation will be integrated fully into Sequoia as a long term feature. This is useful for a number of reasons.

GAPI interfaces allow quick testing and exploitation of new replication techniques. While GORDA performance analysis (D5.3) showed that this is not a particularly high performance approach to replication, it is a useful way to experiment with new techniques. Providing the interfaces is consistent with the philosophy of making Sequoia available as a vehicle for experimentation and research, which in turn should lead to further commercial applications.

GAPI interface call-outs facilitate integration with other replication mechanisms, such as MySQL replication. We have implemented simple interceptors in Sequoia that add comments to SQL statements so that we can integrate with MySQL replication in order to transmit cluster updates over a WAN. However, the GAPI interface model is more consistent and more flexible than the current Sequoia approach. Integrating it long-term therefore supports commercial engagements for uni/cluster.

Finally, having a generic interceptor model allows other types of SQL transformations and integrations. For example, it is conceivable that we can extend Sequoia more easily to perform heterogeneous replication, which requires systematic transformation of SQL. This so far has not been very easy in the existing implementation.

Full integration of GAPI interfaces into Sequoia requires first of all some clean-up of Sequoia code itself as well as corresponding redesign of the GAPI reflector implementation in gorda.db.* packages to ensure acceptable reliability and performance. We plan to do the Sequoia clean-up as part of the upcoming Sequoia 4.0 release and full GAPI integration in Sequoia 5.0. These changes are currently undergoing review in the Sequoia community.

GORDA Reflector Interfaces in PostgreSQL/G

PostgreSQL is an important database for Continuent's business, as it is more widely used for enterprise applications than MySQL, the other main open source database. However, PostgreSQL support for scale-out has been relatively weak. This is a problem for commercial applications, as we need highly performant database replication for PostgreSQL in order to build clusters.

Part of the reason for the weakness is that the PostgreSQL community itself has been reluctant to commit to a particular replication implementation. This has led to something of a stalemate on replication. The main open source replication, SLONY, is based on triggers and runs as an off-board application. This is in marked contrast to the MySQL, which implemented built-in and easy to configure replication relatively early in the product history.

The GAPI approach offers is the first attempt of which we are aware to create replication interfaces rather than implementations of replication. The GAPI design meets many of the requirements of the PostgreSQL community. We plan to encourage the community to look at this approach and plan to

fund development to implement GAPI-like hooks in PostgreSQL in order to support our own replication mechanism(s) for clustering. Due to the mechanics of getting features added to PostgreSQL (mostly having to do with how the community accepts changes), it is likely that such hooks will be quite different in the final implementation from GAPI, but the general idea will be the same. It is in the meantime quite useful to be able to point to the GAPI interface as an example of how to attack the problem of supporting multiple replication mechanisms.

GORDA Reflector Interfaces in Myosotis Connector

The Myosotis connector provides translation of native client wire protocols into JDBC calls. As part of Tungsten we plan to upgrade this component to a much more capable service that will deliver specific connection semantics that may differ depending on the type of replication mechanism used implement the database cluster. It will also handle failover as well as load balancing of reads, in effect taking over a number of the functions currently implemented in Sequoia. (Replication capabilities will remain in Sequoia.)

We are considering two specific ways of utilizing GORDA work within Myosotis.

Partial implementation of GAPI within Myosotis. To implement failover and other operations, Myosotis will require a state model that accurately tracks request and session state. We can copy parts of this model from GAPI. Also, many of the GAPI listeners are relevant to request processing. Change set extraction is not relevant and would not be included, since replication capabilities will be handled at a lower level in the stack.

In additional we are evaluating use of jGCS as a replacement for Hedera, which are the group communication adapters developed for Sequoia.

Bristlecone Test Tools

Bristlecone test tools are designed to provide a full set of easy-to-use scale out benchmarks ranging from mixed load tests to micro-benchmarks. Bristlecone was scaled out considerably during the GORDA time-frame with the goal of being able to test primary/backup, certification, and state-model clusters effectively using a common set of tests. It benefits from the generic design approach to replication of GORDA.

We plan to continue Bristlecone work by adding in more GORDA use cases. Implementing a TPC-W approximation would be a very useful contribution , as the full implementation is quite cumbersome to set up. Also, we would like extend micro-benchmarks to cover group communications, an area in which GORDA spent considerable effort.

While Bristlecone tools are open source, they are in fact an important part Continuent's open source strategy, which is to develop a highly perfor-

mant stack that works across a variety of scale-out implementation architectures. We use results from these tools to drive development of further features as well as completely new products.

Other Plans and Possibilities

There are additional possibilities for commercialization of work coming out of the GORDA project. One area that we will investigate more thoroughly in the future is the Derby replication support. We have not had much involvement in this up until the current time because there were limited commercial opportunities offered by it. However, with Sun Microsystem's commanding position in the open source database market, this integration is more interesting. We will pull in GORDA work as it seems relevant, in much the same way we are doing with Sequoia and PostgreSQL.