SIXTH FRAMEWORK PROGRAMME

INFORMATION SOCIETY TECHNOLOGIES

IST-FP6-004758 GORDA

GORDA Final Report

Date:	31 October 2008
Author	Rui Oliveira
Distribution	Public
Version	1.0
Status	Final

Executive Summary

This document comprises the final report for the IST FP6-004758 STREP Project GORDA “Open Replication of Databases”.

GORDA addresses the lack of a standard, or even a commonly adopted, architecture and interface for the replication of database management systems, fundamental to increase the dependability and scalability of current information systems.

GORDA proposes an open architecture and a set of flexible interfaces to promote the interoperability and to allow plugging virtually any replication protocol into compliant database management systems. GORDA provides a collection of protocols and tools for database replication, reliable group communication and cluster management offering a comprehensive environment ready to be deployed.

All specifications and software resulting from GORDA are publicly available as Open Source on the project’s website. The scientific results were presented in more than fifty published research papers and pilots have been set up by direct integrators and two potential final users.

The project started in October 2004 and concluded in March 2008.

The Challenge

Database management systems are at the core of information systems supporting a wide range of economic, social, and public administration activities. The current thrust for an information society translates at a technological level in additional demand for database management systems. Specifically, it calls for high availability, reliability and prompt disaster recovery. This makes database replication a key technology for the long-term competitiveness of today's businesses.

Replication is however challenged by the integration of legacy systems, by the performance in wide-area and large systems and by the cost of database management systems supporting replication, especially, when many businesses are using cost-effective open-source databases.

The innovation in replication is still stifled by proprietary or inadequate interfaces for developing replication middleware. Developing new replication protocols still often requires that a custom interface with the database management system be also implemented. Replication solutions are therefore either implemented within the database core itself, leading to lack of portability and inter-operability, or are based on client access middleware, which makes it difficult to support advanced database features and hinders performance.

The lack of innovation in database replication, visible in current commercial offers, is per se a major challenge. Most depend on lazy propagation of updates with serious limitations on consistency and resilience to failure, and on a distinguished site to handle all the work, impairing scalability. Consistent and balanced replication protocols are scarce and targeted at niche applications that can afford expensive hardware solutions or have restricted requirements.

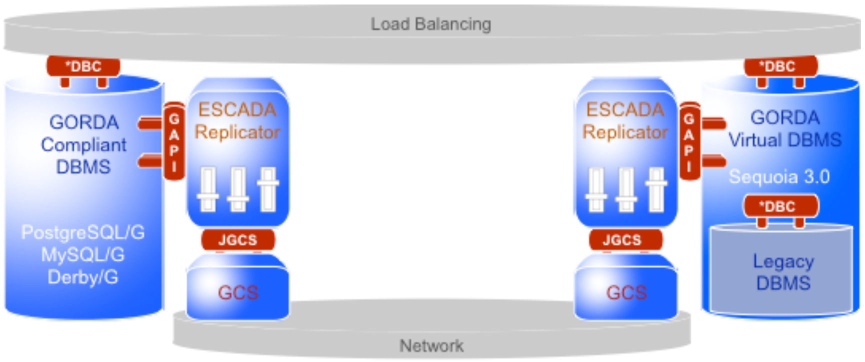
Addressing the Challenge: the GORDA proposition

Database replication is a key technology for the long-term competitiveness of today's businesses. Replication technology is faced with new requirements. First, enterprise-wide availability should take into account legacy information systems. Open markets and a global economy also mean that wide-area support is required not only by large businesses but also by small and medium enterprises, which inevitably raises security issues. Finally, wider applicability of database technology calls for highly scalable database management systems.

The involved heterogeneity, geographical dispersion, and scale have a profound impact on the applicability of replication techniques. The wider applicability of database management systems also calls for an increased concern with costs.

The GORDA project fosters database replication as a means to address these challenges. This is supported by standardising the architecture and a set of interfaces, and by sparking their usage with a comprehensive set of components ready to be deployed.

A high level perspective of the approach is depicted in the next figure.



GORDA proposes detailed specifications for GORDA Architecture and Programming Interfaces (GAPI) – the interface between the database management system (DBMS) and the replication middleware –, and jGCS – the interface between the replication middleware and reliable communication systems (GCS).

The ESCADA replicator middleware is the project's approach to a pluggable replication solution. A comprehensive set of previously developed protocols as well as new ones invented in the scope of the project can be seamlessly integrated and combined to address specific requirements of the target information system.

The consortium provided reference implementations for the proposed architecture, made all interfaces open and actively works to foster the adoption of the interfaces, and of the resulting generic replication middleware, by different database and middleware providers.

Who can benefit from GORDA

Database replication is an enabling technology to ensure availability, performance, and geographic distribution of databases.

The project's proposals and achievements are directly available to database vendors and software developers. The former should find clear advantage in implementing a replication API such as the one GORDA specifies. They are now given the means to address concrete requirements of specific markets. The latter, given the availability of replication aware DBMSs implementing the GORDA API can leverage many existing replication protocols focusing on tuning and specializing them to specific environments as scenarios.

Higher in the chain, system integrators and service providers have now a comprehensive set of alternatives of DBMS and replication protocols to choose from.

End-users are given flexible choices to pick from by not having products tied to a single vendor or proprietary interfaces. Taken alone, the foregoing capabilities allow users to construct better database applications. However, GORDA has a more fundamental role. By enabling and improving replication, GORDA also enables scale-out designs, which offer a different economic model from other more capital-intensive approaches to database systems. Scale-out designs work by spreading copies of data and load across many database hosts. This simple mechanism enables users to create highly available and very highly performing systems at a fraction of the cost of other approaches. As requirements increase, users can incrementally add more hardware, thereby scaling the system efficiently.

Scale-out is thus a design with important economic properties that particularly benefit small as well as growing companies. The low initial cost makes such systems accessible to a wide variety of businesses. It is no accident that countless web businesses use scale-out designs to get started and to provide efficient growth as usage increases. GORDA's focus on replication is thus a root cause for such benefits.

Highlights of Achievements

GORDA key achievements are given below and detailed in the following sections.

Architecture The overall achievement of GORDA is the detailed definition and reference implementations of an open architecture for the construction and management of interoperable clustered databases.

Programming Interfaces Key to the overall result is the specification of standard programming interfaces that abstract the coordination between the DBMS and the replication middleware, and the general requirements on reliable group communication systems.

Replication Protocols The project carried an unusual deep assessment of existing database replication protocols. As a result, and besides providing complete and open-source implementations of them, GORDA proposes two novel protocols targeted at high-performance clustered databases as well as wide area inter-cluster settings.

Monitoring, Management and Benchmarking As part of the project's comprehensive proof of concept a whole set of tools for the deployment, monitoring, management and benchmarking has been made available.

Technology Assessment During the whole project lifespan the consortium studied, assessed and adopted several technologies in the several axes of action. A worthwhile achievement is the current understanding of their value and articulation.

Building a user community The GORDA consortium has ever since been deeply involved both in the scientific communities cross-cutting the research areas of database, dependability, distributed systems and autonomic systems, as well as cooperating, and often coordinating, with several systems development forums where the project's industrial partners are major players.

Current and Future Synergies Many collaborations have been established during the project's lifespan. Some with research groups and companies involved in FP6 projects were very fruitful and are being sustained by several common efforts in the creation of a so-called community around the challenges of Dependable Distributed Data Management. The recent contribution of GORDA to emerging FP7 initiatives such as the NEXOF-RA project is a very good example of ongoing collaborations.

The GORDA Results

GORDA Architecture and Programming Interfaces

Result at a glance

Motivation

- Define an open and comprehensive architecture to enable the construction of flexible and interoperable replicated database systems
- Define a set of interfaces capable of reflecting the requirements of existing database replication protocols

Main characteristics

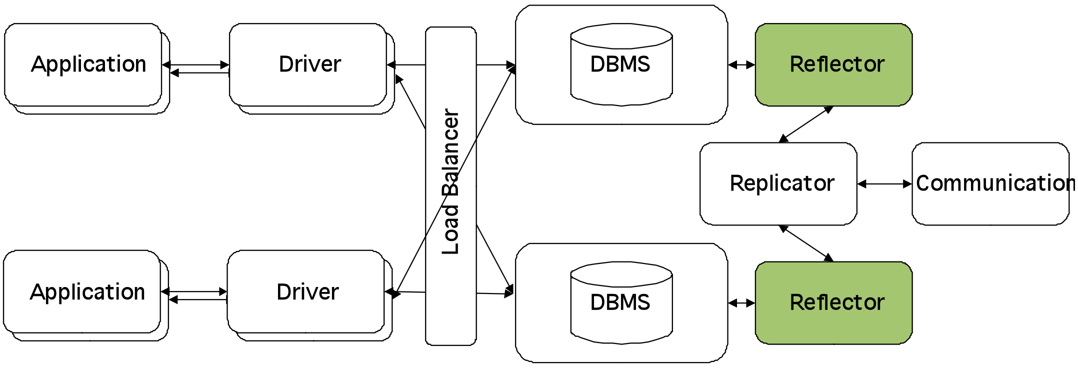
- Comprehensive architecture capable of accommodating all kinds of replication protocols
- Variable geometry interfaces. Each implementer is free to provide only a subset of GORDA Programming Interfaces that is adequate for each situation
- Façade interfaces allowing the efficient manipulation of the internal state of the DBMS
- Modular and technology agnostic

Main outcome

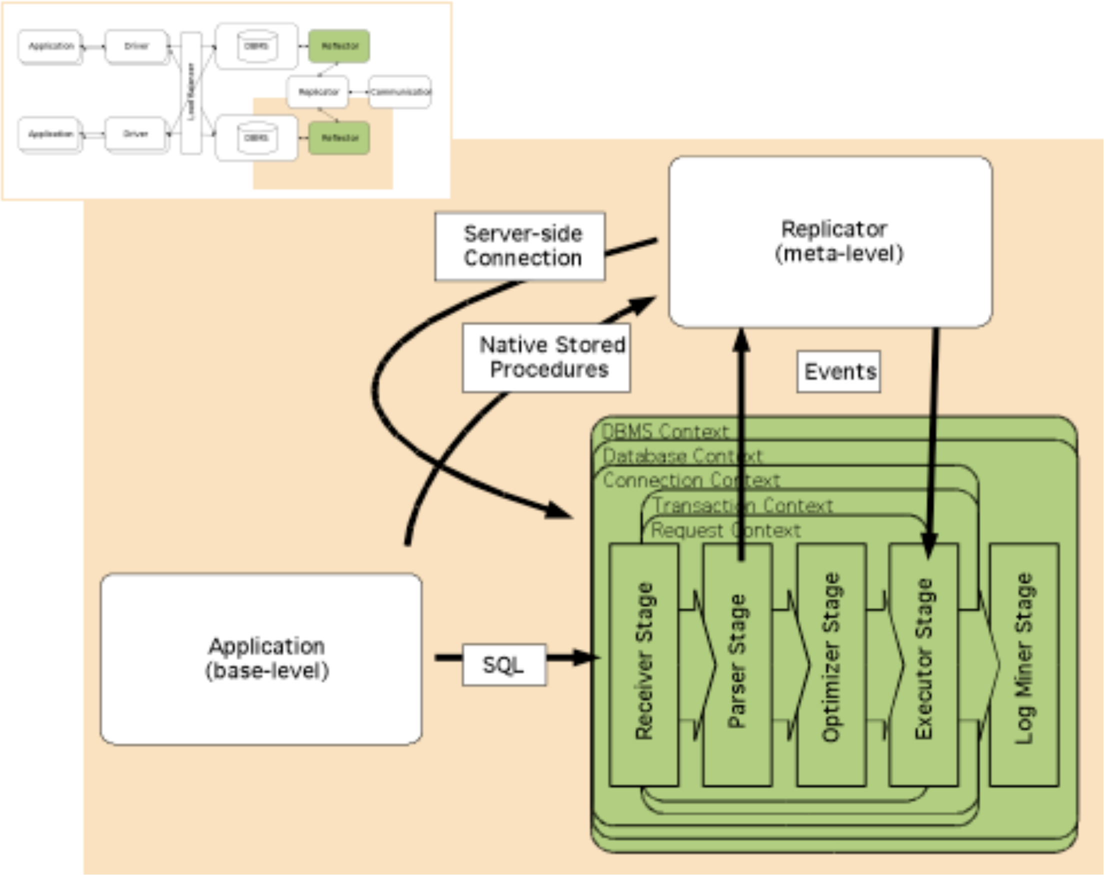
- Detailed specification
- Several reference instantiations

Result Description

The following architectural pattern defines the abstract components and interactions of a replicated database management system.



The main focus is on a reflective view of transaction processing that interfaces data processing and replication. This view combines the transaction processing pipeline with the several contexts relevant to the interface with the DBMS, to the specific database and to each client connection.



The jGCS Application Interface

Result at a glance

Motivation

- To create a generic interface for group communication that should be used by the GORDA replication protocols

Main Characteristics

- Integrated service of data and membership in a group of communicating processes
- Interface with variable geometry and a common semantics
- Clear decoupling between the application code and the specific implementations of group communication
- Support for multiple group based programming paradigms

Main Outcome

- Detailed specification
- Reference implementation for three state of the art group communication toolkits, namely Appia, JGroups and Spread
- Performance assessment reports

Result description

jGCS has been defined as a generic interface that may be used to wrap multiple reliable group communication toolkits. jGCS has been designed for the Java programming language and leverages on several design patterns that have recently become common ground of Java-based middleware. The interface specifies not only the API but also the (minimum) semantics that allows application portability. jGCS owns a number of novel features that makes it quite distinct from previous attempts to define standard group communication interfaces, namely:

- jGCS aggregates the service in several complementary interfaces, namely a configuration interface, a message passing interface, and a set of membership interfaces.
- jGCS provides support for recent research results that improve the performance of group communication systems.
- jGCS introduces negligible overhead, even when implemented as a wrapper layer and is not supported natively by the underlying toolkit.

Innovative Database Replication Protocols

Result at a glance

Motivation

- To create a novel database replication protocolos targeting high-performance cluster settings and wide area networks, namely serving the replication of geographically dispersed data centers

Main Characteristics

- Leverage on the experience of existing consistent database replication protocols
- Resource efficient and adaptive protocols
- Flexible consistency criteria protocols

Main Outcome

- Provably correct protocols
- Reference implementations
- Evaluation results

Result description

Major improvements have been made to the so-called *certification based* protocols for database replication. Conflict-aware load balancing protocols have been proposed that take into account the conflicting characteristic of the workload to mitigate the major drawback of this kind of protocols, transactions aborts due to concurrent uncoordinated execution.

Devoted to the cluster setting, where balancing dependability and performance represents the major challenge, a novel protocol called Akara has been devised. Akara is a hybrid approach of several well-know database replication techniques and manages to ensure the fairness characteristics of *conservative execution* protocols while achieving the desirable performance of the *certification based* family. Akara may well become the protocol of election for consistent database clusters.

On the other front, that is, the wide area scenarios, GORDA developed WICE, a highly efficient, certification based protocol for the interconnection of database clusters. While being extremely frugal in the use of wide area links, WICE still provides full consistent full replicated database clusters and the capability of seamless fail-over in the case of total failure of a cluster.

Autonomic Management System

Result at a glance

Motivation

- Adjust system accordingly to the dynamic nature of requirements and the environment

Main Characteristics

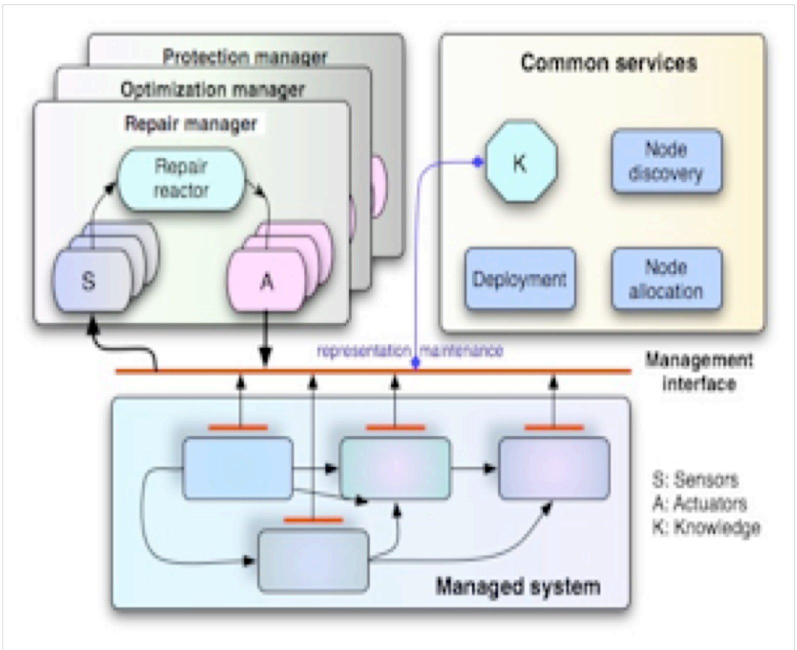
- Stateless architecture
- Hierarchic capabilities
- Compliant to the standard Java Management eXtensions
- Pluggable policy scripts written in a scripting language

Main Outcome

- Generic software module to manage complex distributed systems

Result description

The GORDA autonomic module provides an abstraction to manage the components of a complex, potentially distributed, system. This process (figure below) can be split in four main phases: gathering the state of the managed components (through the use of Sensors), building the overview of the system based on the previously acquired Knowledge (combination of real-time data, system configuration data and pre-defined policies), decide based on business policies and, finally, apply those decisions on the components to achieve the desired results (through system Actuators).

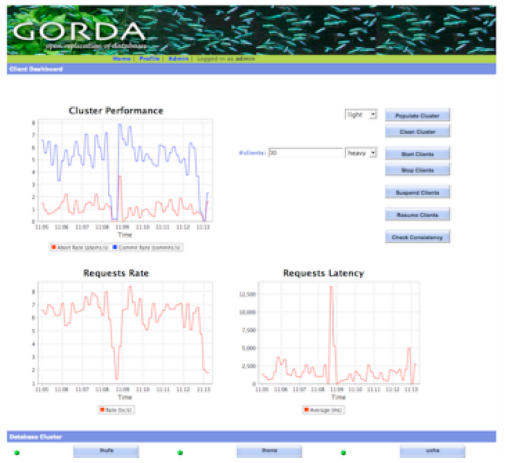


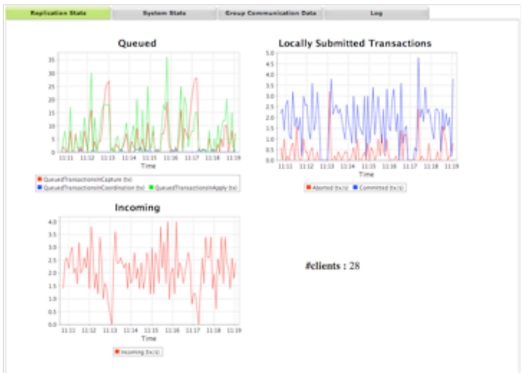
Monitoring Console

Result at a glance
Motivation • To provide a global management tool for the system Main characteristics • Based on JMX technology • Web-based interface • Highly customized Main outcome: • Open-source software package based on the jManage tool (http://www.jmanage.org) • Global overview of the system • Detailed view of each replica • Simple user interface to perform cluster-wide operations

Result Description

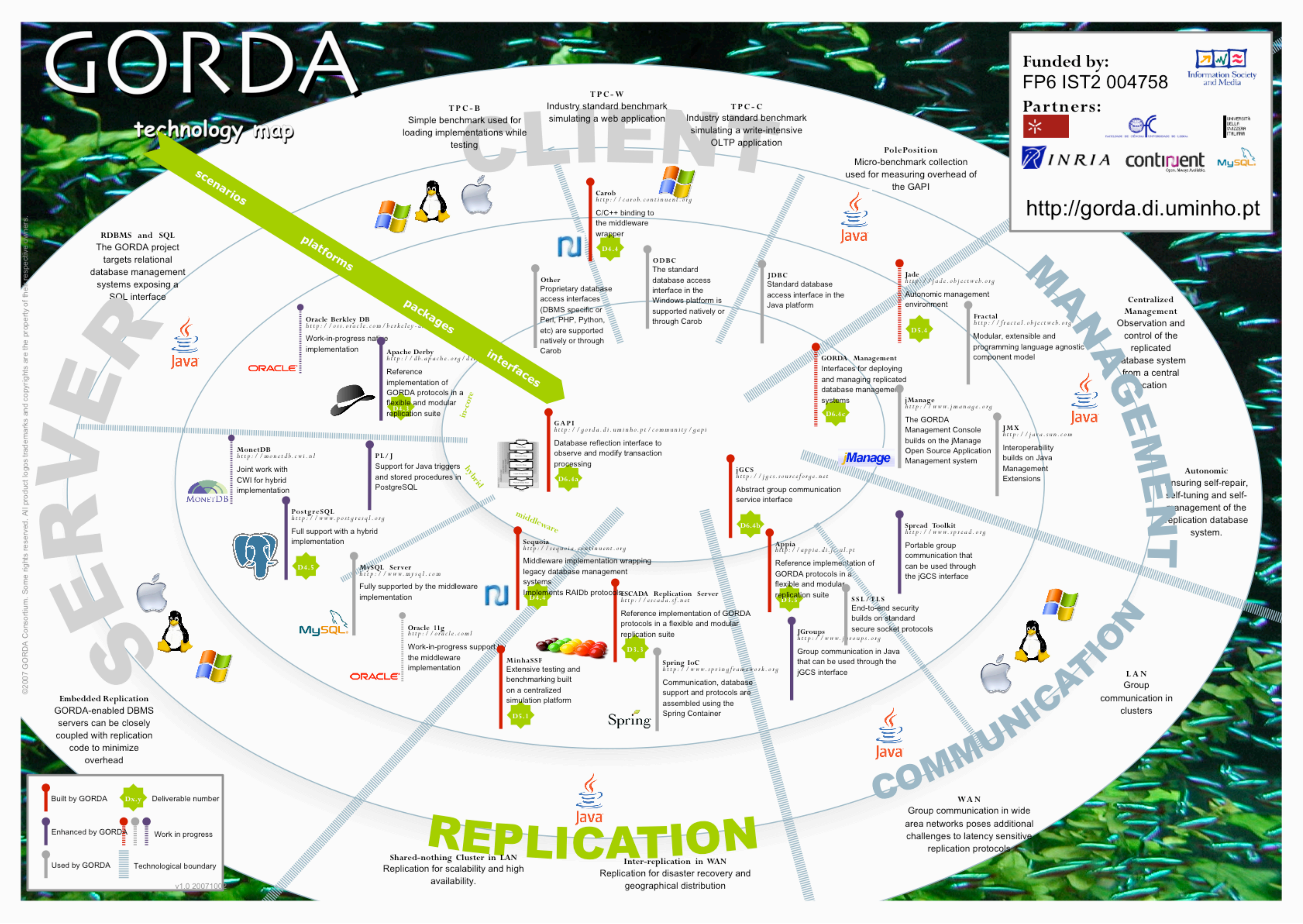
The supervision console is the frontend to the management tool set and provides a comfortable, easy to use, web-based interface to monitor and manage the cluster and is based on a highly customized version of the jManage 2.0 platform. In a nutshell, the tool allows to reset the cluster, start, stop, suspend and resume clients, and check for the consistency of all replicas. It also allows to monitor the input transaction rate, average latency, abort and commit rates, and per replica, analyze its queues, systems statistics, and usage of the communication substrate. An example screenshot of the console is presented next.





Technology Assessment

The GORDA technology map:



GORDA technology map

Funded by:
FP6 IST2 004758

Partners:

INRIA continuent MySQL

http://gorda.di.uminho.pt

CLIENT

TPC-B
Simple benchmark used for loading implementations while testing

TPC-W
Industry standard benchmark simulating a web application

TPC-C
Industry standard benchmark simulating a write-intensive OLTP application

PolePosition
Micro-benchmark collection used for measuring overhead of the GAPI

Carob
http://carob.continuent.org
C/C++ binding to the middleware wrapper
D4.4

Other
Proprietary database access interfaces (DBMS specific or Perl, PHP, Python, etc) are supported natively or through Carob

ODBC
The standard database access interface in the Windows platform is supported natively or through Carob

JDBC
Standard database access interface in the Java platform

scenarios platforms packages interfaces

in-core hybrid middleware

GAPI
http://gorda.di.uminho.pt/community/gapi
Database reflection interface to observe and modify transaction processing
D6.4a

MANAGEMENT

Jade
http://jade.objectweb.org
Autonomic management environment
D5.4

Fractal
http://fractal.objectweb.org
Modular, extensible and programming language agnostic component model

Centralized Management
Observation and control of the replicated atabase system from a central cation

GORDA Management
Interfaces for deploying and managing replicated database manageme systems
D6.4c

jManage
http://www.jmanage.org
The GORDA Management Console builds on the jManage Open Source Application Management system

JMX
http://java.sun.com
Interoperability builds on Java Management Extensions

Autonomic
nsuring self-repair, elf-tuning and self- anagement of the eplication database system.

COMMUNICATION

jGCS
http://jgcs.sourceforge.net
Abstract group communication service interface
D6.4b

Spread Toolkit
http://www.spread.org
Portable group communication that can be used through the jGCS interface

Appia
http://appia.di.fc.ul.pt
Reference implementation of GORDA protocols in a flexible and modular replication suite
D3.5

SSL/TLS
End-to-end security builds on standard secure socket protocols

JGroups
http://www.jgroups.org
Group communication in Java that can be used through the jGCS interface

LAN
Group communication in clusters

WAN
Group communication in wide area networks poses additional challenges to latency sensitive replication protocols

REPLICATION

Sequoia
http://sequoia.continuent.org
Middleware implementation wrapping legacy database management systems
Implements RAIDb protocols
D4.4

ESCADA Replication Server
http://escada.sf.net
Reference implementation of GORDA protocols in a flexible and modular replication suite
D3.3

MinhaSSF
Extensive testing and benchmarking built on a centralized simulation platform
D5.1

Spring IoC
http://www.springframework.org
Communication, database support and protocols are assembled using the Spring Container

Shared-nothing Cluster in LAN
Replication for scalability and high availability.

Inter-replication in WAN
Replication for disaster recovery and geographical distribution

SERVER

RDBMS and SQL
The GORDA project targets relational database management systems exposing a SQL interface

Oracle Berkley DB
http://oss.oracle.com/berkeley-db
Work-in-progress native implementation

Apache Derby
http://db.apache.org/derby
Reference implementation of GORDA protocols in a flexible and modular replication suite
D4.3

MonetDB
http://monetdb.cwi.nl
Joint work with CWI for hybrid implementation

PL/J
Support for Java triggers and stored procedures in PostgreSQL

PostgreSQL
http://www.postgresql.org
Full support with a hybrid implementation
D4.5

MySQL Server
http://www.mysql.com
Fully supported by the middleware implementation

Oracle 11g
http://oracle.com
Work-in-progress support by the middleware implementation

Embedded Replication
GORDA-enabled DBMS servers can be closely coupled with replication code to minimize overhead

Built by GORDA
Dx.y Deliverable number
Enhanced by GORDA
Work in progress
Used by GORDA
Technological boundary

v1.0 2007100

Pilots

Most of the results and technology generated by the project have been widely disseminated to the database developer forums and are being steadily and progressively integrated in the industrial partners product lines. In the following, two concrete examples are described. The first is the complete integration of GORDA's Sequoia middleware solution into Continuent's current state-of-the-art clustering and replications offer. The second is a pilot carried within a small Portuguese company with a specific and very demanding use case is described.

Continuent Tungsten

Overview

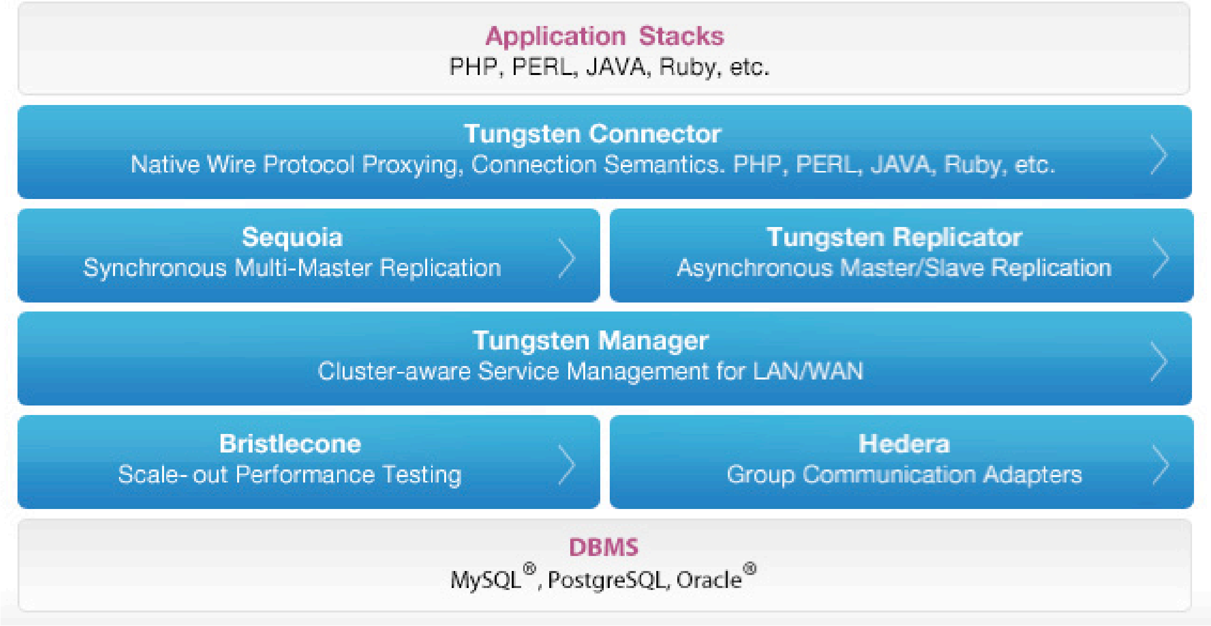
Sequoia provides middleware-based replication and provides the ability to wrap databases that do not support GORDA replication APIs. As a result of the GORDA research projects we have made a number of improvements to Sequoia that have resulted in a much broader architecture that subsumes a wide range of clustering and replication problems.

From an industrial perspective the major contribution of GORDA is pluggable replication. Pluggable replication allows users to select the method of replication that is most applicable to the clustering problem at hand, since each type of replication has trade-offs. State machine replication, for example, avoids single points of failure but has difficulty serializing transactions efficiently. Master/slave (primary/backup) replication on the other hand is very fast and handles a wide range of SQL but introduces a single point of failure.

Sequoia/Tungsten Architecture

The initial work in GORDA led to changes to Sequoia multi-master replication to enable Sequoia to “wrap” databases with middleware to make them compliant with GORDA. These changes are in the Sequoia codeline now. However, the basic concepts have led us to a much broader architecture that is quite general and takes advantage of pluggable replication concepts as well as other components that were improved over the course of the GORDA project.

The following diagram shows the resulting architecture, which is known as Tungsten.


Application Stacks
PHP, PERL, JAVA, Ruby, etc.
Tungsten Connector
Native Wire Protocol Proxying, Connection Semantics. PHP, PERL, JAVA, Ruby, etc.
Sequoia
Synchronous Multi-Master Replication
Tungsten Replicator
Asynchronous Master/Slave Replication
Tungsten Manager
Cluster-aware Service Management for LAN/WAN
Bristlecone
Scale- out Performance Testing
Hedera
Group Communication Adapters
DBMS
MySQL®, PostgreSQL, Oracle®

The Tungsten architecture contains Sequoia multi-master replication as well as a new master/slave replication product known as the Tungsten Replicator. It also makes used of the following additional components that were developed or improved over the course of GORDA:

- Tungsten Connector (formerly known as Myosotis) – A flexible native-client to JDBC proxy. This is the "front door" to clusters built on replication and provides location transparency for database replicas.
- Bristlecone performance test tools. These include load and performance tests for clusters, for example to check replication latency.
- Hedera group communications adapters and Applia group communications. These have both been upgraded over the course of the project.

The most important characteristic of the combined architecture is that it works as a stack. The Tungsten Connector, for example, works with any type of database. The components within the stack are flexible and highly capable. Sequoia, for example, supports a broad range of capabilities for state machine replication.

Industrial Results

All components describe here are either in industrial use or in beta with selected customers. All components in the architecture are also available as open source, which means that they are accessible not just to enterprises but to any organization that wishes to explore replication and implement new types of systems. Here are a few results from current use.

A. The Tungsten Connector (Myosotis) continues to be a critical part of the architecture. We are extending this to become multi-purpose gateway that can route SQL transactions to databases using different replication and partitioning patterns.
B. Sequoia replication development is continuing with Sequoia 4.0, which started toward the end of the GORDA effort.
C. Tungsten Replicator. While not directly part of GORDA, we are using GORDA API ideas to help make databases like PostgreSQL more amenable to replication. Continuent is funding design work in this area.
D. Bristlecone tools. These are in everyday use for cluster and replication testing.
E. Hedera group communications. Hedera group communications wrappers are part of every Continuent product. Appia is fully supported.

The ability to offer multiple replication forms through a stack approach is fundamental to Continuent's business strategy and from our perspective is the most important and most tangible result of our work on the GORDA project.

Software for these and other projects are freely available at the Continuent community website, which is hosted at the following URL:

http://www.continuent.com/community.

Telbit Pilot

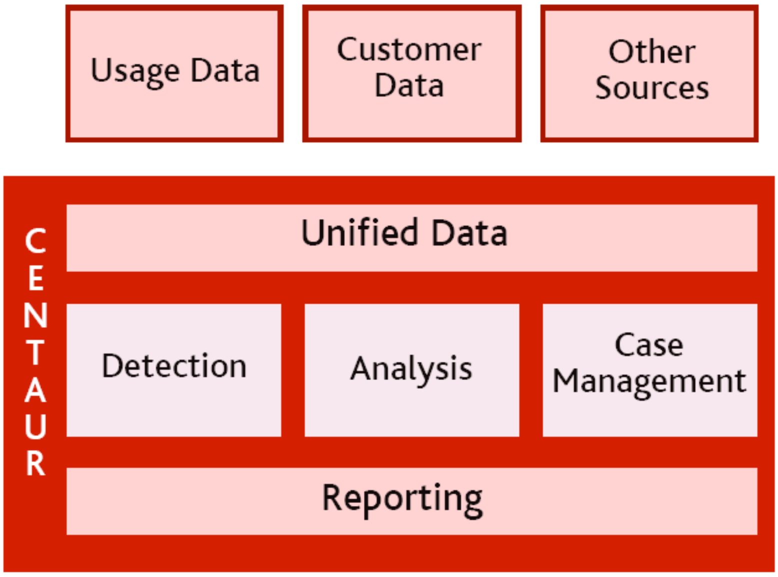
Background

Telbit was founded in 1998 as a spin-off of PT Inovação (Portugal Telecom R&D Company) in Aveiro, Portugal. Telbit provides services and solutions which go through all the software development lifecycle: Development (core-business), Tests, Support and Maintenance.

Several international organizations have estimated that fraud may affect between 3% and 6% of an operator's gross revenue. Having an efficient fraud management system may easily reduce these values. Telbit and PT Inovação have developed Centaur to target the specific information needs of fraud officers and analysts and provides them with a simple and fast access to detailed information regarding suspicious call activity, that may result in revenue loss.

Scope of the Pilot

Centaur's overall architecture, as represented in the next figure, is supported by several components, which bring together innovative technological aspects with flexibility and simplicity in its use.

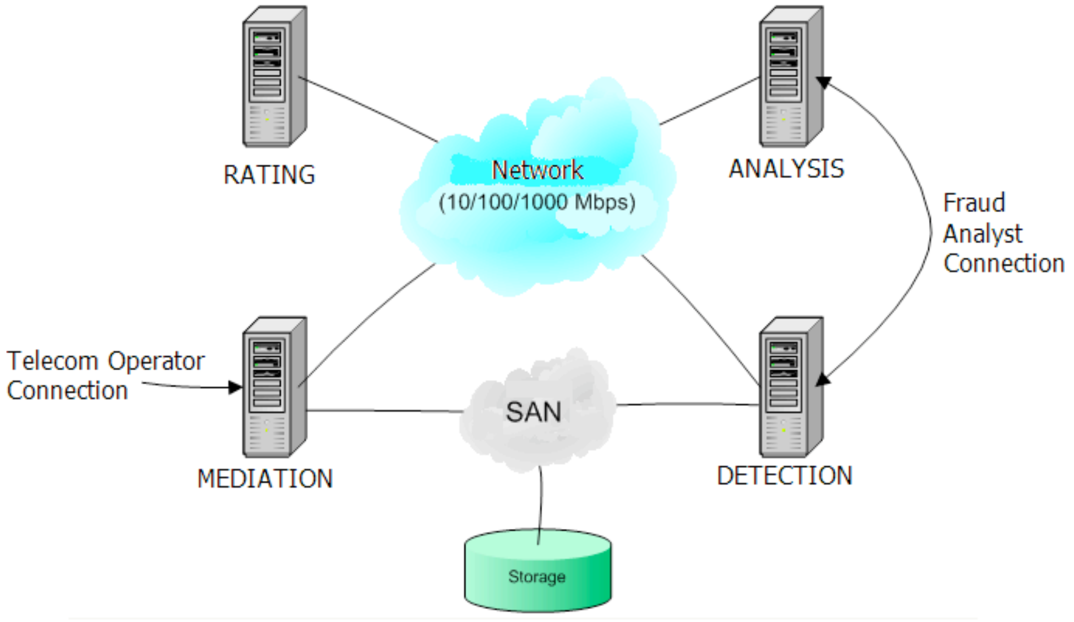


Centaur's high-level architecture

As a fraud management system for telecommunications, Centaur has multiple detection techniques that concurrently analyze customer data. In systems that deal with very large volumes of data, scalability is one of the most important aspects to consider.

Replication Setting

To guarantee optimal scalability and performance, Centaur was built using a distributed approach. As an example, in the next figure we present the typical approach for a medium-sized operator, with around 5 million customers and an average of 50 million call records a day.



Standard components distribution

In this architecture, each of the main components of the system is installed in a separate machine, with its own database, properly sized and tuned for its specific task but some information must be kept up-to-date on more than one system. Centaur deals with the following replication scenarios:

1. Alarms table – synchronous multi-master replication;

Centaur, currently, does not implement a replication mechanism for this scenario. It uses a single table of alarms in the ANALYSIS database. All operations are performed against this table. Remote operations are performed via a database link that connects DETECTION to ANALYSIS.

Issues:

- Possible loss of alarm events, if the ANALYSIS database is down during the alarm generation phase of the detection processes;
- Increased time to generate or update alarms if the ANALYSIS database is heavily loaded or in the event of network problems.

2. Suspicious records – asynchronous primary-secondary replication;

Centaur has a suspicious records update process that runs daily and sweeps the call tables in search of new suspicious records to be copied to the corresponding suspicious record tables in ANALYSIS. This is accomplished by a standard PL/SQL stored procedure.

Issues:

- Very high workload while fetching the suspicious records from the tables that contain all call records and subsequent copy of these records to a remote database via a database link.

3. Customer information – synchronous primary-secondary replication

Centaur updates customer information on a daily basis. To accomplish synchronous primary-secondary replication, in this particular case, replication is performed on all systems at the same time during the final merge phase. After the information is ready to be published to all systems, the sequential SQL merge statements are issued for each of the systems involved (all remote systems are accessed via database link) and performs a COMMIT in the end of all operations.

Issues:

- Failure to update all systems in case of a single failure in one of the systems;
- The overall process takes longer since the replication is performed sequentially to each system.

Results

The Centaur team has already identified some situations where GORDA in-core replication can be used to guarantee the required data dissemination patterns and consistency. As described, full database replication is not a requirement for the Centaur application, it requires only some database tables to be replicated.

1. Alarms table – synchronous multi-master replication;

The Alarms table is used both by the Detection and Analysis processes. In this scenario, GORDA replication removes the dependency between Detection and Analysis as each process updates local data during transaction execution and the GORDA replication protocol ensures Alarms table consistency upon commit. Also, GORDA replication enables horizontal scale-up, allowing several instances of the Detection and Analysis processes to run, while, currently only, vertical scale-up would be permitted.

2. Suspicious records – asynchronous primary-secondary replication;

The adoption of GORDA replication protocols, allows to transparently replicate the selected records from DETECTION to ANALYSIS, and also keeps the clean-up process from running on both databases. It suffices to run in one of them, as the replication protocol ensures that the records will be deleted from both databases upon commit.

3. Customer information – synchronous primary-secondary replication

Finally, with respect to customer information, it is actually generated in MEDIATION and copied to both DETECTION and ANALYSIS. In this process, every change in a customer's data results in deleting all of that customer's data and re-inserting it with the changes. With GORDA replication, customer information is replicated simultaneously to both DETECTION and ANALYSIS, instead of being copied to one of the databases only after the previous one has been updated. Updating records instead of deleting them and re-inserting a duplicated one with the altered data may also reduce network traffic.

Benefits

The GORDA in-core implementation uses a PostgreSQL database, while Centaur is currently implemented in ORACLE, using stored procedures and PL/SQL.

In order to convince the Centaur team of the GORDA replication protocols advantages, a three-phase work plan has been established. In the first phase, the Customer Information process will be converted to PostgreSQL, and adopt GORDA replication. This process has been chosen as it is the simplest one, and the one that is less dependent on ORACLE's PL/SQL.

The second phase encompasses the evaluation of both PostgreSQL and ORACLE in a predefined set of queries, so both teams get confident that the adoption of PostgreSQL will not affect the performance of the Centaur application.

Finally, the Centaur application will be re-implemented in PostgreSQL, and use the GORDA replication protocols, as described above.

Evaluation

It is necessary to evaluate whether a solution based on PostgreSQL can prove adequate to the Centaur system's performance requirements.

In order to do so, a sample application, was adapted to be compatible with PostgreSQL. Three different scenarios were considered:

> **Scenario A**: For comparison purposes, we ran a series of timed test runs to determine how long it takes to populate the example database, via JDBC, using a vanilla PostgreSQL. Here, two hosts were used: one to run the populate application and the other to host the PostgreSQL database.
>
> **Scenario B**: In this scenario, GORDA in-core replication is introduced. Again, the populate application is run in a separate host. Here, the replication scenario mimics Centaur's Customer Information Process, whereby customer data is inserted in one replica (MEDIATION) and replicated to others (DETECTION and ANALYSIS). Again, runs of the populate application were timed.

Scenario C: Here, we tested an ad-hoc replication procedure, using vanilla PostgreSQL databases in all replicas. Replication is achieved by leveraging PostgreSQL's dblink module, which provides inter-database communication. The setup is identical to that described for the previous scenario.

Results of the timed runs were as follows:

- For scenario **A**, simply to populate the database, it took 225m5.569s ;
- For scenario **B**, it took 650m10.945s to populate the primary replica and replicate the data to the other two replicas;
- For scenario **C**, it took 849m21.946s to populate the primary replica and replicate the data to the other two replicas.

From these results we can draw some conclusions. First, we can see that the GORDA approach achieves results. Second, we can also conclude that the ad-hoc replication scenario does not pose a viable alternative to the GORDA in-core implementation approach in terms of performance. Also, GORDA offers additional guarantees in terms of dependability and data consistency, issues that are not addressed in scenario C.

In face of these results, the GORDA in-core implementation proves to be the recommended approach to support the Customer Information Process.
As mentioned above, in order to attain a comparison with the system currently in use with Centaur, based on the ORACLE database system, the next step is to run scenarios A and C on an ORACLE setting, and from these results to form a decision on whether to use PostgreSQL instead.

Availability of Results

All project results are publicly available through the project's website at http://gorda.di.uminho.pt

All research reports and papers are publicly available occasionally under usual copyrights of the publisher. All software source code is publicly available licensed as follows: Interfaces mappings under the BSD license so that the code can be linked against any other system, the ESCADA replication middleware under the GPLv3 license, and all patches licensed in strict compliance with the target software packages.

Partners

Universidade do Minho Universidade do Minho at Braga, Portugal is represented in the project by the Distributed Systems Group (GSD) of the Computer Science Department. The GSD has a strong background in research and implementation of advanced database replication based on group communication, acquired in previous research projects.
Contact: Prof. Rui Oliveira, email: rco@di.uminho.pt

The Università della Svizzera Italiana, Switzerland is represented in the project by the Distributed Computing group. The group does research on theoretical and practical aspects of the design of reliable applications for large-scale networks and infrastructures to enable integrated and collaborative use of high-end computers, networks, and databases.
Contact: Prof. Fernando Pedone, email: fernando.pedone@unisi.ch

The Fundação da Faculdade de Ciências da Universidade de Lisboa, Portugal is represented in the project by the Distributed ALgorithms and Network Protocols (DIALNP). The DIALNP research group, which is part of the LASIGE laboratory at University of Lisboa, is devoted to the study of algorithmic and communications support to build complex and dynamic distributed applications.
Contact: Prof. Luís Rodrigues, email: ler@ist.utl.pt

The Institut National de Recherche en Informatique et en Automatique (INRIA), France, is represented by the Sardes project located in the Rhône-Alpes Research Unit. The overall goal of the Sardes project is to investigate the construction of distributed software infrastructures (operating system and middleware) to support global computing.
Contact: Prof. Sara Bouchenak, email: Sara.Bouchenak@inria.fr

Continuent offers the industry's leading, patent-pending high-availability, fault tolerance and clustering software for application and hardware vendors. Continuent, prior Emic Networks, was founded by Internet security and traffic management specialists in response to the explosive growth in the need for reliability and performance in mission-critical applications.
Contact: Dr. Robert Hodges, email: robert.hodges@continuent.com

MySQL AB is the company that develops, supports and markets the MySQL database server globally, distributed at zero price under the GNU General Public License (GPL), or sold under a commercial license to those who do not wish to be bound by the terms of the GPL. Today MySQL is the most popular open source database server in the world with more than 4 million installations powering websites, datawarehouses, business applications, logging systems and more. Customers such as Yahoo! Finance, MP3.com, Motorola, NASA, Silicon Graphics, and Texas Instruments use the MySQL server in mission-critical applications.
Contact: David Axmark, email: david@mysql.com