Project no. 004758

GORDA

Open Replication of Databases

Specific Targeted Research Project

Software and Services

Deployment Guides For Replication Strategies

GORDA Deliverable D5.5

Due date of deliverable: 2008/03/31
Actual submission date: 2008/05/21

Start date of project: 1 October 2004

Duration: 42 Months

Universidade do Minho

Revision 1.0

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

Alfrânio Correia Júnior, U. Minho
José Pereira, U. Minho
Luís Soares, U. Minho
Robert Hodges, Continuent
Rui Oliveira, U. Minho

Abstract

This report suggests how the different replication protocols proposed in D3.1 and D3.2 may be applied to the use cases described in D1.2 and deemed representative of the requirements database replication in current information systems.

Contents

Chapter 1

Introduction

Different fault-tolerant solutions are required to fulfill different needs and requirements established by different applications. Different replication protocols may fit better in some scenarios and perform poorly in others. For instance, to replicate information among databases connected through long-distance links, one needs to take into account partitioning, communication low bandwidth and high latency. Deploying a replication protocol that does not consider these factors may lead itself to unavailability or inconsistency when partitioning is not considered, or poor performance when resource limitations are disregarded.

The system's workload is also extremely important in this choice. Workloads with long-running transactions and hot-spots are not suited for optimistic protocols, for example. The optimistic execution of transactions in these protocols lead to aborts when concurrent transactions happen to conflict. This makes abort rate of these protocols highly sensitive to high service or queuing times and hot-spots.

Here too, there is no *one-size fits all* approach. This document discusses how the different replication protocols proposed in D3.1 and D3.2 may be applied to the various use cases described in D1.2. We start by summarizing the use cases and replication protocols. Then we discuss their mapping. Finally, we briefly describe the current status of the different prototypes and how they fulfill the requirements established by the various use cases.

1.1 Objectives

The GORDA Deployment Guides For Replication Strategies report has the following goals:

- discuss the advantages and disadvantages of using a protocol to fulfill the the needs of a given use case;
- identify the current state of the available replication solutions;
- point-out open issues for future work.

1.2 Relationship With Other Deliverables

This document builds essentially on previous deliverables D1.2 - User Requirements Report, D3.1 - Wide-area Protocols Report and D3.2 - Wide-area Protocols Report discussing, to a certain extent, how the project solutions serve the identified user requirements.

Chapter 2

Use Cases and Protocols Overview

The following sections briefly presents the use cases used as a start point to develop the replication solutions proposed by GORDA focusing in key characteristics such as the read-write ratio and propagation requirements. Then the replication protocols proposed in D3.1 and D3.2 are outlined.

2.1 Use cases

GORDA report D1.2 defines a variety of requirements for GORDA driven by use cases that stem from standard industrial benchmarks to Continuent's experience with different clients. Specifically, these use cases fall into two broad categories: LAN and WAN replication. The former category is built upon industrial benchmarks and a benchmark that resembles telecoms' workloads. The latter category draws a scenario with hierarchical replication resembling a matrix connect to several branches through long-distance links.

A **Basic Availability Case** is proposed to emphasize that some clients are driven towards the replication solutions only to fulfill one requirement: availability. Sometimes they even accept to trade performance for availability.

The TPC-C and the TPC-W represent the industrial benchmarks. **TPC-C** is the industry standard on-line transaction processing benchmark. It mimics a wholesale supplier with a number of geographically distributed sales districts and associated warehouses. TPC-C is a mixture of read-only and update intensive transactions and has a long-running transaction named *Delivery*. **TPC-W** is a transactional web benchmark that simulates the activities of a business oriented transactional web server. TPC-W simulates three different profiles by varying the ratio of browse to buy operations: primarily shopping (WIPS), browsing (WIPSb) and web-based ordering (WIPSo). TPC-W is a read intensive workload even with the WIPS profile which has a higher number of updates.

The **Telecom** use case defines the operation provided by a Telecom operator. A major Telecom equipment vendor provides software and hardware to telephone service carriers that allows them to fully support Local Number Portability (LNP). They need a simple replicated database solution for their Local Service Management System (LSMS) targeting availability.

The **FSecure** use case defines a company that provides a wide range of computer and network security products via subscriptions. In this application, a central or master database, hosted at a single network operations center (NOC) maintains the subscription authorization and profile information for all of the customers for a specific security product. Updates on subscription content must be immediately propagated to a set of backup subscription servers hosted in a number of different NOC in geographically diverse areas. Changes that have been made on subscriber machines as well as statistical information collected on each local, backup, server must be propagated to the primary in some pre-defined interval.

2.2 Protocols

Different replication protocols were studied, tested and developed.

DBSM (Database State Machine) is a non-centralized replication technique based on a certification procedure that guarantees one-copy serializability (1SR).

VDBSM (Versioned DBSM with Load Balancing) introduces versions for each data item to reduce in-core changes to the DBMS and has a certification procedure similar to the DBSM. By means of an integrated load balancing, it reduces conflicts and improves performance.

DBSM SI (DBSM with Snapshot Isolation) does not use read sets for certification and detects write-write conflicts thus providing snapshot isolation as the consistency criterion which reduces the amount of information disseminated among replicas.

WICE is a database replication protocol based on group communication that targets interconnected clusters. In contrast with previous proposals, it uses a separate multicast group for each cluster and thus does not impose any additional requirements on group communication, easing implementation and deployment in a real setting. Nonetheless, the protocol ensures one-copy equivalence based on certification procedures.

NODO (NOn-Disjoint conflict classes and Optimistic multicast) is a conservative protocol that guarantees that concurrent conflicting transactions are executed sequentially achieving one-copy serializability (1SR). Conflicts are determined by conflict classes which are commonly tables. In contrast to previous protocols, this is a middleware approach.

Sequoia (former C-JDBC) is a middleware solution for database clustering on a shared-nothing architecture built with commodity hardware. Sequoia hides the complexity of the cluster and offers a single database view to the application. Similar to NODO, it provides a conservative execution.

The following table summarizes which protocols are middleware or in-core approaches. Roughly, the former may be deployed with off-the-shelf databases, in contrast with the later which requires modifications to databases. In addition to that, it shows which protocols are targeted for a LAN or WAN and which ones are optimistic or conservative. In what follows, we use the terms cluster and LAN interchangeably and the same applies to the terms certification based and optimistic.

Protocol	In-Core	Middleware	LAN	WAN	Optimistic	Conservative
DBSM	x		x		x	
VDBSM	x		x		x	
DBSM SI	x		x	x	x	
WICE	x			x	x	
NODO SI		x	x			x
SEQUOIA		x	x			x

Chapter 3

Mapping Use Cases to Protocols

For each use case, the following sections describe the advantage of using the protocols proposed in deliverables D3.1and D3.2.

3.1 Basic Availability

This use case does not impose any performance restrictions or any very specific feature. Any cluster replication protocol in D3.2 might be used.

3.2 TPC-W

This a transactional web benchmark with a high read/write ratio. For that reason any cluster replication protocol defined in D3.2 might be used. In particular, the load balancing solution and the caching provided by Sequoia might improve the overall cluster performance thus making this protocol the best choice for this scenario.

Sequoia however, has a single point of failure represented by a single controller that handles client requests and dispatches them to databases. To circumvent this drawback a certification-based protocol to propagate changes among different controllers is proposed. Unfortunately, this extension trades performance for availability as described in deliverable D5.3. So, it should be deployed with caution.

The load balancing provided by VDBSM makes this protocol a good choice too. Considering just performance, the DBSM SI is also a possible approach. Further studies are however required to see if the consistency criterion defined by the benchmark is not violated with the weaker criterion guaranteed by the protocol.

In D1.2, this industrial use case is extended with a set of cluster operation tasks briefly described below:

- **State Transaction to Active** represents an operation performed to start up a peer.
- **State Transaction to Standby** represents an operation performed to remove a peer from the cluster in order to start, for instance, a backup.
- **State Transaction to Deferred Standby** represents the execution of a DDL operation.

Again any replication protocol and reference implementation might be used to fulfill such requirements. There is no restriction regarding them.

3.3 Telecom

The **Telecom** use case defines the operation provided by a Telecom operator. One might use any optimistic protocol to fulfill the availability and performance requirements defined by the LSMS. On the other hand, the LNP databases can be clustered by the previous protocols and also by Sequoia or NODO.

3.4 TPC-C

TPC-C represents an OLTP application that typically can be found in any large enterprise that sells products or services. The high update ratio, a strong TPC-C's characteristic, shapes the choice for protocols. In such cases, the optimistic approaches perform better as the conservative approaches might generate high contention.

The optimistic approach however should use a load balancing mechanism as defined by VDBSM protocol. Otherwise, the abort ratio might be high due to conflicts. If this is not the case both DBSM and DBSM SI might be used. Researches have determined that a TPC-C application provides a serializable execution even when running with snapshot isolation. Thus the DBSM SI might be used without any concern. Different applications however should be analyzed before using the DBSM SI to avoid harming any database constraint or simply it should not be used.

In this scenario, we also need to take into account that due to conflicts, long-running transactions may never have a chance to commit. The TPC-C has a long-running transaction named delivery that may show this behavior if it is not chopped into smaller transactions as suggested by the benchmark. If this sort of transaction cannot be chopped so, we need to trade performance for fairness by switching to Sequoia or NODO.

3.5 FSecure

FSecure defines a master site that receives subscription updates for a product and synchronously propagates them to remote backup sites. Remote replicas accept read operations on subscriptions in order to validate them and register information on subscriber hosts and their configurations which are asynchronously propagate to the master. The master site is connect to the backup sites through long-distance links with low bandwidth and high latency. And these sites may be composed by a single peer or a set of peers.

In this scenario, the Wice protocol is the best choice as it does not use group communication among remote sites thus being feasible to be deployed to replicate information among sites connected through long-distance links. Although not required by the scenario, updates on remote sites are also synchronously propagated to the master as specified by the Wice protocol. If the long-distance links are reliable and the amount of changes to be replicated is not high, the DBSM SI might be a choice as it reduces the impact on the network by not propagating read sets. The weak consistency criterion provided by the protocol is not a issue in this scenario as different sites update and handle different sets of information.

Chapter 4

Analyzing Current Solutions

Most replication protocols proposed by GORDA were developed and are available for download as prototypes.

DBSM SI works on PostgreSQL-G, NODO works on Derby-G and PostgreSQL-G. These are databases that provide support for the GORDA Replication API (GAPI). VDBSM and Wice have not been implemented yet. In addition to the cluster protocols presented, it is possible to have a synchronous or asynchronous primary backup protocol based on group communication.

In contrast, Sequoia allows to use any off-the-shelf database except for its high availability version (i.e. Sequoia-G) which requires PostgreSQL-G.

In addition to providing different replication approaches, GORDA also has management and deployment modules as described in D5.5 and D5.4.

In the following sections, we map the current solutions to the use cases and their requirements.

4.1 Mapping Use Cases to Current Solutions

4.1.1 Basic Availability

One might use DBSM SI on PostgreSQL-G, Sequoia or Sequoia-G. Or it is possible to use a primary backup protocol as there is no performance restriction.

4.1.2 TPC-W

One might use DBSM SI on PostgreSQL-G, Sequoia or Sequoia-G. Or it is possible to use a primary backup protocol when a single replica manages to handle all updates.

4.1.3 Telecom

One might use DBSM SI on PostgreSQL-Gwith 2 or 3 replicas to provide high availability properties to LSMS. On the other hand, the LNP databases can be clustered with DBSM SI on PostgreSQL-G, Sequoia or Sequoia-Gproviding a synchronous and fault-tolerant data access.

4.1.4 TPC-C

In this case, the DBSM SI on PostgreSQL-G seems the best choice despite the fact that it does not provide an integrated load balancer. If the abort rate is high it is necessary however to change the application in order to introduce basic load balancing techniques. In particular, for TPC-C this would mean to redirect clients from the same warehouse to the same replica.

4.1.5 FSecure

The DBSM SI on PostgreSQL-G is the best choice in this case and the only feasible and available solution for WAN. By not propagating read sets, it reduces the amount of traffic exchanged among replicas.

4.2 Mapping Requirements to Current Solutions

This section briefly maps the current reference implementations to the requirements defined in D1.2. Whenever appropriate, current drawbacks and issues are described.

4.2.1 Application Transparency

This requirement states that applications should not be changed in order to use any replication solution proposed by GORDA. It takes into account: (*i*) SQL Transparency, (*ii*) Concurrency/Lock Granularity, (*iii*) Permission Access, (*iv*) Error Status Transparency, (*v*) Connection Status Transparency and (*vi*) Client API Transparency. Most reference implementations fulfill these aspects lacking minor details. Sequoia-G does not support stored procedures and concurrency control based on predicate locking. Derby-G and PostgreSQL-G do not replicate statements and both implementations require that every relation has a non-composite unique index or primary key. Resuming the current status of the prototypes:

Key Point	References	Observation
SQL Transparency	D3.2	D3.1 and D3.3 do not have it implemented. D3.2 does not support stored procedures.
Concurrency/Lock Granularity	D3.2, D3.1, D3.3	D3.2 does not have predicate locking.
Permission Access	D3.2, D3.1, D3.3	-
Error Status Transparency	D3.2, D3.1, D3.3	-
Connection Status Transparency	D3.2, D3.1, D3.3	-
Client API Transparency	D3.2	In a near future, this feature might be easily extracted from D3.2 in order to be used by D3.1 and D3.3.

4.2.2 Database Consistency Criteria

This requirement states that solutions should guarantee the consistency criterion requested in a transaction's context. Sequoia-G just provides support for serializability and the Derby-G and PostgreSQL-G currently just provide support for Snapshot Isolation. None currently provide support for other isolation levels and bounded inconsistency as defined in D1.2.

Key Point	References	Observation
Database Consistency Criteria	-	None of the current implementations provide support for all possible isolation levels.

4.2.3 Performance and Scalability

This requirement states that solutions should provide read scalability and minimize negative impact on performance for write intensive workloads. In particular, this requirement takes into account three aspects : (*i*) read scalability, (*ii*) connection scalability and (*iii*) load balancing. In contrast to Sequoia-G that fulfills them, Derby-G and PostgreSQL-G just provide support for the first two. Performance assessment however presented in D5.3 shows that Sequoia-G performs poorly with write intensive workloads.

Key Point	References	Observation
Read Scalability	D3.2, D3.1, D3.3	-
Connection Scalability	D3.2, D3.1, D3.3	-
Load Balancing	D3.2	In a near future, this feature might be easily extracted from D3.2 in order to be used by D3.1 and D3.3.

4.2.4 System Management

This requirement takes into account the following. Clusters are managed and monitored from a single point through three different means: JAVA API (JMX Interface), SNMP traps and a text-based socket interface. The JVM provides upon the JMX a small SNMP agent that provides information on the JVM and we developed a text-based protocol that can be used by any JMX component. To provide information for the dynamic configuration procedure, extensive statistics are gathered through such interfaces. In particular, the Jade+ is responsible for the dynamic operation, for instance, adding and removing replicas when a configurable threshold is triggered (e.g. cpu usage).

Key Point	References	Observation
Centralized Management	D5.4	-
Text-Based Socket Interface	D5.4	-
Java API	D5.4	-
SNMP Traps	D5.4	-
Extensive Statistics	D5.4	-
Dynamic Configuration	D5.4	-
Integrated Configuration	D5.4	-

4.2.5 Security

This requirement is transparently provided through the group communication toolkit that enables SSL encrypted communication.

Key Point	References	Observation
Security	D3.5	-

4.2.6 WAN Support

This requirement is fulfilled by a synchronous or asynchronous Primary/Backup replication approach or DBSM SI as previously described. The WICE protocol specifically addresses this requirement.

Key Point	References	Observation
WAN Support	D3.1	-

4.2.7 Failure Handling

This requirement states that failures should be transparently handled. This implies that fault replicas are automatically removed from the group and that clients are automatically transferred to a healthy peer. Currently, This last property is only supported by Sequoia-G.

Key Point	References	Observation
Automatic Fail-over	D3.2	-

4.2.8 Maintainability and Testability

This requirement is provided through the extensive use of Java and established technologies such as Log4J, unit testing and continuous integration.

Key Point	References	Observation
Fine-Grained Dynamic Diagnostics	Most technical deliverables	-
Fine-Grained Dynamic Tests Hooks	Most technical deliverables	-
System Diagnostic Dump Facility	Most technical deliverables	-

4.2.9 GORDA System Software Upgrades

This requirement should be done without causing any system downtime. In order to achieve this, one may remove one replica at a time, do the software upgrades and right after re-add it to the replication group.

Key Point	References	Observation
Zero Downtime Field Upgradeable	D3.2	-
Rolling DBMS Upgrades	D3.2	-

4.2.10 Network Equipment Requirements/Compatibility

This requirement defines that GORDA shall be capable of operating in any standard inter-networking environment.

Key Point	References	Observation
Standard Network Equipment	-	There is no restriction. Even when group communication is used, it can be emulated by multiple sends.