Project no. 004758

GORDA

Open Replication of Databases

Specific Targeted Research Project

Software and Services

Evaluation Report

GORDA Deliverable D7.3

Due date of deliverable: 2008/03/31
Actual submission date: 2008/05/28

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

UMINHO

Revision 1.0

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

José Pereira, U. Minho
Nuno Carvalho, U. Minho
Luís Rodrigues, U. Lisboa
Luís Soares, U. Minho
Rui Oliveira, U. Minho

Abstract

This document provides an evaluation of the demonstration. This is done quantitatively, by providing measurements of performance, and also qualitatively, by explaining how each deliverable is evaluated and each objective is met.

Chapter 1

Introduction

1.1 Objectives

The goals of this document are twofold:

- Provide a summary of performance results obtained in demonstration Scenario 2 that can be compared with the performance of a non-replicated database management system.
- Evaluate how demonstration activities contribute to illustrate the capabilities of the GORDA Integrated Prototype, in particular, regarding requirements identified early in the project.

1.2 Relationship With Other Deliverables

This deliverable is an evaluation of the activities of D7.2 (Demonstration), which includes activities during Review Meetings 3 and 4 and a screen capture movie. The detailed description of demonstration scenarios is available in D7.1 (Deployment Plan).

Therefore this document does not provide a full evaluation of all performance aspects of database support and replication protocol prototypes. These evaluation is available in deliverables D5.1 (Algorithms Performance and Reliability Assessment Report) and D5.3 (Interface and Modules Performance Assessment Report).

The qualitative evaluation is done according to requirements identified early in the project and described in D1.2 (User Requirements Report).

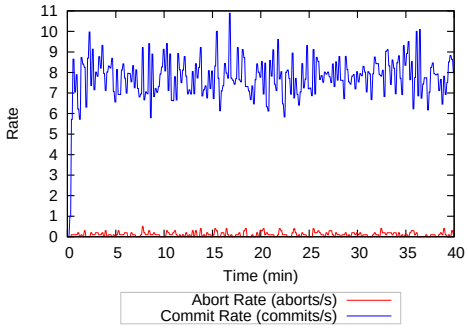
Chapter 2

Quantitative Evaluation

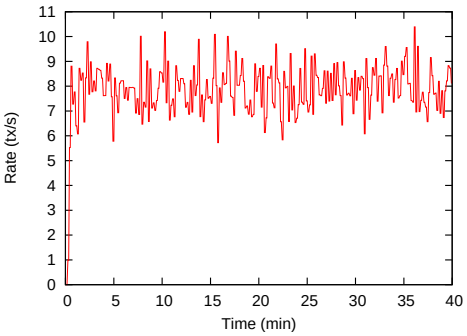
This chapter provides a quantitative evaluation of the demonstration by presenting and analyzing data obtained in Scenario 2. The presentation of results closely follows the configuration of the interactive console used in the demonstration.

2.1 Cluster

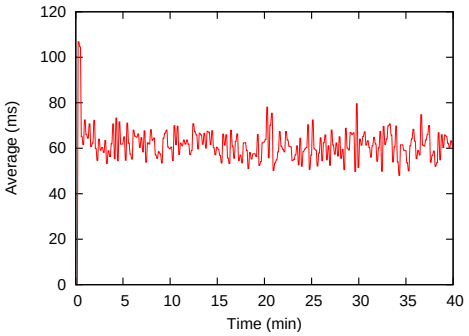
Figure 2.1 summarizes overall cluster performance, as observed by the workload generator, with approximate throughput of 8 tps and end-to-end latency of 60 ms. It can also be observed that only a small number of transactions are aborted due to concurrency control in the cluster.



(a) Cluster performance



(b) Request rate

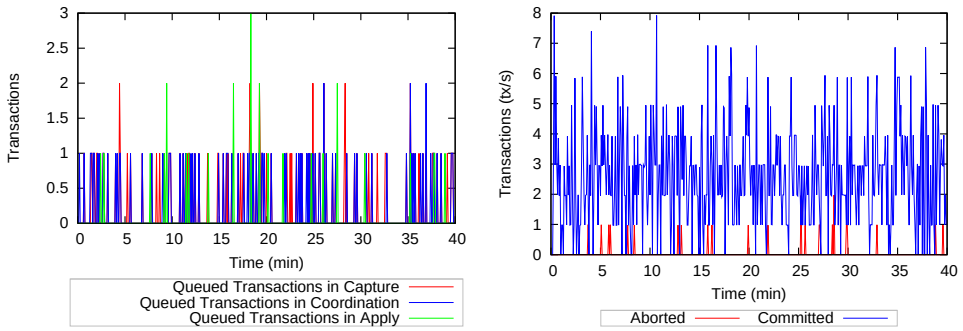


(c) Request latency

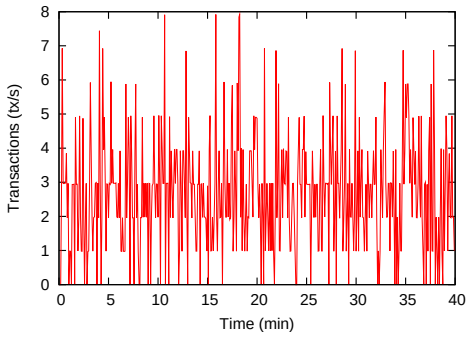
Figure 2.1: Cluster view

2.2 Replica

Figure 2.2 shows measurements obtained at the replication protocol within a single replica. As expected, the incoming transaction rate is one third of the overall throughput shown in Figure 2.1. It can also be observed that the replication protocol does not represent a bottleneck, as queuing is not noticeable.



(a) Queued (b) Locally submitted transactions

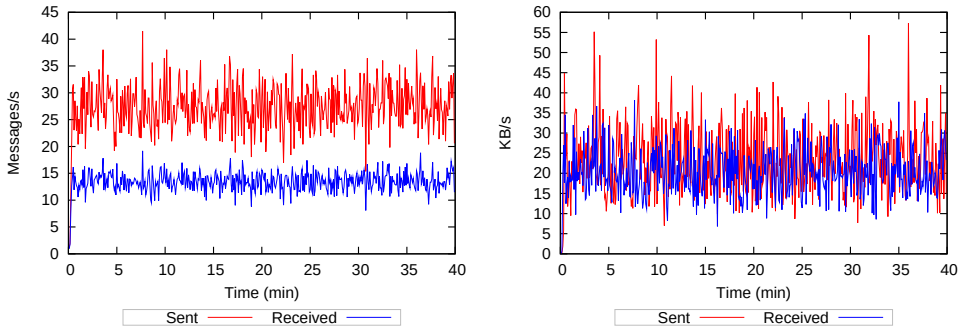


(c) Incoming

Figure 2.2: Replica view: Replication

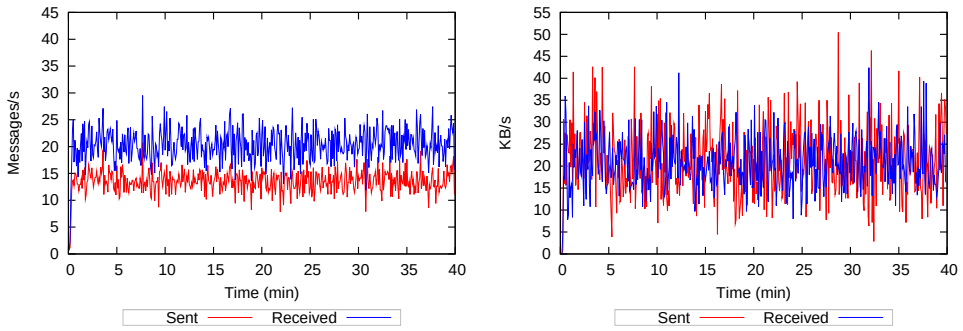
2.3 Group Communication

Figures 2.3 and 2.4 show statistics of traffic as recorded by the Appia group communication protocol. In contrast to all other measurements, in which all replicas are symmetrical, it can be observed that one of the replicas sends more messages and receives less messages than others. This happens because Appia is configured with a sequencer based total order protocol, in which one group member in each view is selected to order messages by sending sequence numbers. Therefore, it sends more messages but receives less, since messages to self do not traverse the network. The impact in terms of bytes transmitted is however much less obvious, since additional messages sent by the sequencer are short control messages with a small payload.



(a) Messages (b) Bytes

Figure 2.3: Replica view (coordinator): Group communication

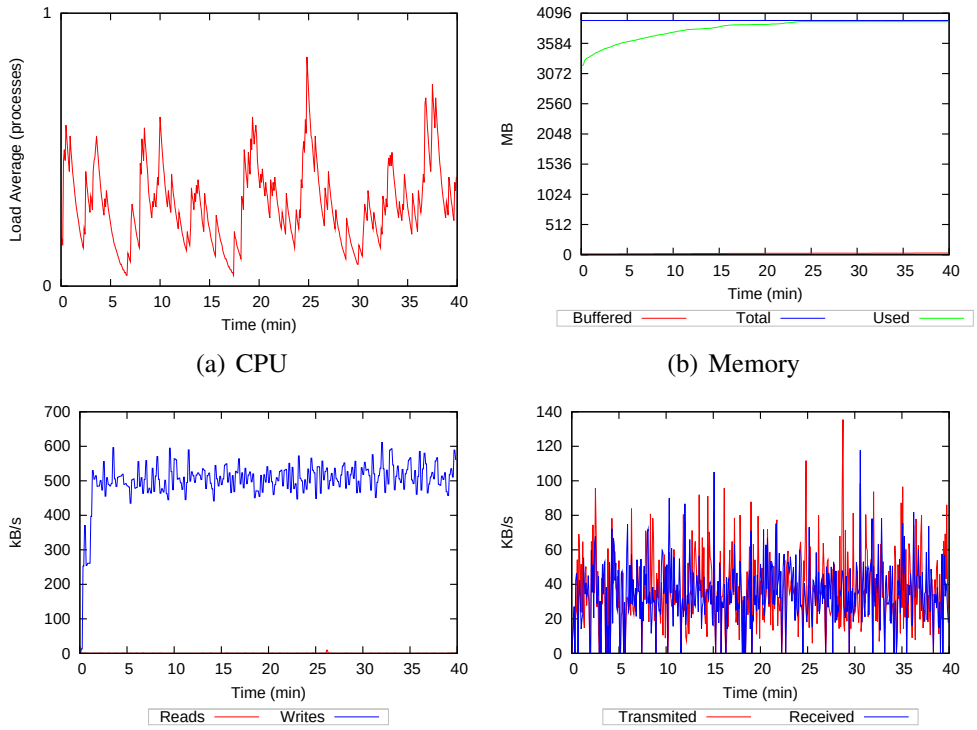


(a) Messages (b) Bytes

Figure 2.4: Replica view (others): Group communication

2.4 Resources

Data obtained by observing operating system information about resource usage is presented in Figure 2.5. This shows that the system is correctly configured, since reading from storage is minimal and most storage bandwidth is dedicated to writing. It is also interesting to compare network usage as measured at the operating system level to the results obtained within the group communication protocol, which does not consider TCP/IP or Ethernet header overhead. As group communication implies a number of small control messages, the overhead is noticeable.



(a) CPU (b) Memory

(c) Storage (d) Network

Figure 2.5: Replica view: System information

Chapter 3

Qualitative Evaluation

This chapter provides a qualitative evaluation of the demonstration by matching demonstration scenarios described in the Deployment Plan with requirements identified in D1.2 and software packages delivered by the GORDA project.

3.1 Software Packages and Integration

As shown in Table 3.1, all software packages delivered by the project are used in at least one demonstration scenario. Integration of the resulting prototypes is illustrated by:

- Replication protocols and group communication protocols being reused in all scenarios. This shows that the project has achieved the goals of promoting the interoperability of DBMSs and replication protocols by defining generic architecture and interfaces, as well as providing general-purpose and widely applicable database replication protocols.
- Management tools are reused in all relevant scenarios. This shows that the project has achieved the goal of providing uniform techniques and tools for managing replicated database systems.

Furthermore, the demonstration of an unmodified MySQL DBMS within the GORDA integrated prototype shows that it is indeed possible to have an interim middleware-based solution that allows immediate integration of current DBMSs.

Package	**Deliverable**		**Scenarios**
ESCADA	Replication modules reference implementation	D3.3	all
Appia	Group communication protocols	D3.5	all
Derby/G	In-core proof-of-concept	D4.3	0
Sequoia 3.0	Middleware proof-of-concept	D4.4	3
PostgreSQL/G	Hybrid proof-of-concept	D4.5	1a,1b,2
Jade	Management tools set	D5.4	1a,1b,2,3

Table 3.1: Software packages summary table.

3.2 Requirements

3.2.1 Application Transparency

Application transparency is demonstrated particularly in Scenario 0, in which a replicated database cluster is configured from scratch and the original DBMS command-line client used to issue SQL statements.

3.2.2 Database Consistency Criteria

Scenarios 1a, 1b, and 2 scenarios demonstrate how PostgreSQL's native Snapshot Isolation consistency criteria is extended to a replicated setting. The synchronous primary-backup protocol of Scenario 3 shows how MySQL's 2PL-based Serializability is extended to a replication settting.

Although not demonstrated, replication protocol reference implementations allows also multi-master serializability (except phantoms) to be achieved in DBMS that expose read-set extraction.

3.2.3 WAN Support

Towards demonstrating WAN support, Scenarios 0 and 3 show a primary/backup replication protocol which can be configured to convey updates synchronously or asynchronously. Furthermore, the protocol used for Scenarios 1a, 1b, and 2 is based on Snapshot Isolation and thus suitable for WAN usage as described in D3.1.

3.2.4 Security

SSL support has not been demonstrated, but is transparently available in the Appia group communication protocol.

3.2.5 DBMS Support

The demonstrations show how both PostgreSQL, Derby and, through Sequoia 3.0, MySQL are supported.

3.2.6 Supportability and Testing

Towards fulfilling the requirements for supportability and testing, the integrated prototype provides a common logging facility based on the standard Apache log4j platform. This is demonstrated in Scenarios 1a, 1b, 2, and 3.

3.2.7 System Management

Requirements for system management are fulfilled by building on Jade and JMX technologies as shown in Scenarios 1a, 1b, 2, and 3, allowing centralized deployment and control of a group

of replicas. The collection of extensive statistics is demonstrated in Scenario 2 and presented in Section 2.

Although not demonstrated, by using the Java Management Extensions (JMX) the integrated prototype satisfies also the following requirements for a text based console, by loading a script interpreter (e.g. BeanShell) and mapping in managed beans, and t for SNMP support by using the Java Dynamic Management Kit to bridge JMX to SNMP.

The requirement for integrated database configuration is not entirely fullfiled, since this would require an uniform interface for DBMS configuration, which is a time consuming undertaking and out of the scope of the project. However, proposed interfaces offer the necessary hooks for a vendor to validate and veto specific DBMS configurations in order to achieve it.