Table of Contents

CORAL server Project

Howto

OLDCoralServerCompilation

Development process notes

Normally, we send engineering releases to the experiments weekly. The releases are prepared and sent on Monday, and we collect the first results/observations from the experiments at the Tuesday afternoon meetings.

Releasing procedure:

- Zsolt decides the next release number and sends it on Thursday: RELEASETAG
- The server project developers check in and tag all the files from the Coral{Server,Access,Protocol,Messaging} modules that they want to include with RELEASETAG_PRE
- The server project developers send a short summary about the changes/new features to Zsolt
- Zsolt integrates the changes, and creates a new compilation from scratch, runs the tests
- In case of satisfactory results, the RELEASE_PRE is renamed to RELEASE
- Release letter is sent to the experiments containing the new label and change list

Purpose

The CORAL server intends to solve some existing performance, licensing and authentication issues.

- **Performance**: each client connection is handled by a separate process in the database servers. There are several client connections in the same time, meaning significant CPU/memory usage.
- **Licensing**: In some countries database technologies cannot be exported, so the third-party DB technology must be completely insulated
- **Authentication**: TODO

Overview

The CORAL server resides in the middle tier (MT) of a 3-tier system: the client (CL), the CORAL proxy/server, the database backend (DB) (see the block diagram). The middle tier acts as a remote CORAL client to which the 'local CORAL client' delegates all communication with the database servers.

The proxy sits between the CORAL server and the client processes. Its purposes are:

- reduces number of connections to CORAL server by *multiplexing* many clients connections into a single (or maybe few) physical connections to server.
- can *cache* results of the queries to reduce the number of repeated queries on server side.

To do the two task above efficiently, the CORAL server protocol should provide direct support for these features.

The communication between the MT and the DB is defined by the CORAL plugins, no change is required here. A protocol between the CL and the MT need to be developed.

In the client side, a new plugin must be developed (ProxyAccess). It is where the user calls to the CORAL

API are translated into packets to be sent to the MT, and data received by the MT is reinterpreted to be fed back to the local user on CL.

More detailed overview

Requirements

General statements

Deliverables

- The client-side plugin
- The CORAL server

Routing

Multiplexing (or rather reverse process or *de-multiplexing*) requires some form of *routing* supported by protocol. Routing in general can be a complex problem, but in case of the static system, with a single path between any client and server, routing implementation could be made simple and efficient. An example of such implementation could be a simple routing table inside the proxy process. In the proxy routing table the mapping will consist of the one-to-one mapping of (Connection Id, Client Id) on the client side and server side.

Caching

Server side cache: for caching to work, the proxy should be able to match the reply from the server with the request from client. This can be done, for example, if every message from client had a unique request Id and reply from server includes the same request Id. Client may know that some requests do not need to be cached at all. Besides the request Id and the caching flag the proxy needs to compare the requests themselves, to be able to match new requests with those already present in the cache. It would be better if proxy does not need to know protocol details for that, simple lexicographical comparison of requests would be the most efficient way. This implies that requests coming from clients cannot contain client-specific information at all.

Client side cache: data for bulk update is cached on CL and sent to MT only on flush; data retrieved from MT is retrieved in bunches, and only few rows at a time are served to the local user on CL.

Communication protocol

Description of the version 0 protocol

Detailed requirements

Detalied requirements

Assumptions

The assumptions are conditions that are fulfilled by the environment of the server.

ID	Description	Reference	Status
A001.	There is a single path between the client and the server.	link	TBD
A002.	Client Id is unique for any connection.	link	TBD
A003.	The proxy analyzes only the packet headers, the rest of the packets can have a free format.	link	TBD

A004.	The proxy does not do protocol translation	link	TBD

Internal references

- Some thoughts on the CORAL proxy
- Sharepoint collaboration site

-- ZsoltMolnar - 14-Jul-2008

This topic: Persistency > OLDCoralServer
Topic revision: r21 - 2010-06-22 - AndreaValassi