GRID job tracking and monitoring

Dmitry Rogozin

Laboratory of Particle Physics, JINR

07/08/2006 - 29/09/2006

Outline

Introduction

- Existing job monitoring systems, tracking and monitoring requirements

Motivation

- New monitoring project, Data model independence

Current activities

- DB structure, Abstract DB layer

Introduction

Resources available at a single site do not satisfy production cycles

↓

Grid technologies

↓

Distributed Systems and Web Services

↓

Service Oriented Architecture (SOA)

Introduction: tracking and monitoring requirements

STAR monitoring system should :

- allow to monitor the state of the submitted jobs
- provide ability to receive bundle summary and detailed information about submitted jobs
- work with distributed resources
- work on multiple architectures / OS's and be compatible with multiple database engines

Introduction: existing job monitoring systems

J-LAB production system

- Allows users to monitor and control their jobs from remote via a Web Interface. Based on the idea that information is useful as bundle summary and has detailed list.

The D0/CDF system (SAMGrid)

- Integrated production system for D0 and CDF experiments . Monitoring system has a visual map of sites available to the experiments.

BOSS / BODE (CMS Batch Object Submission System)

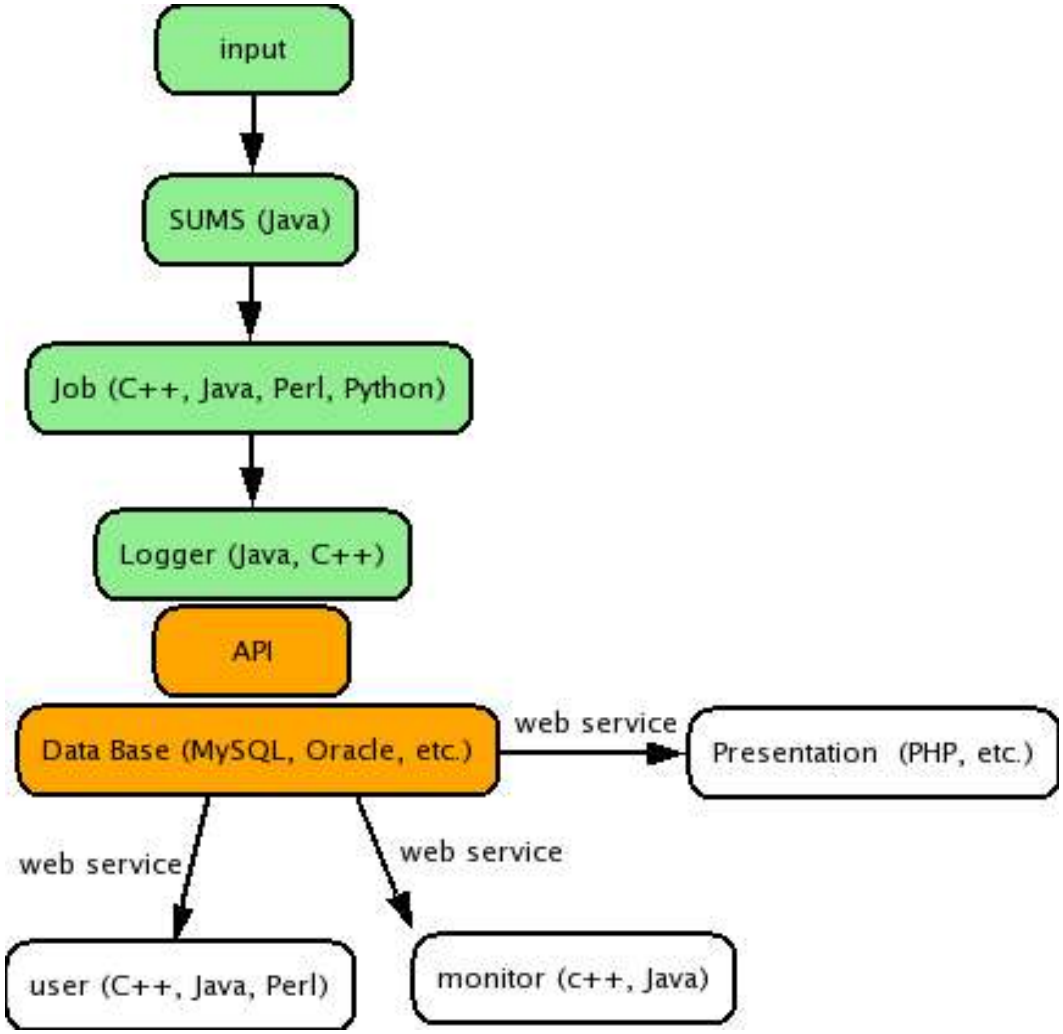
- Provides an easy to use book keeping system for jobs running

MAUI / MOAB Grid scheduler

- Grid based scheduler which allows distributed workload to be run across independent cluster. MOAB separates cluster and job monitoring (global and detail view)

STAR needs GRID monitoring system because systems described above do not fully satisfy our needs at this moment !

Database abstraction layer



Picture 1. Monitoring project

Applications that will use it:

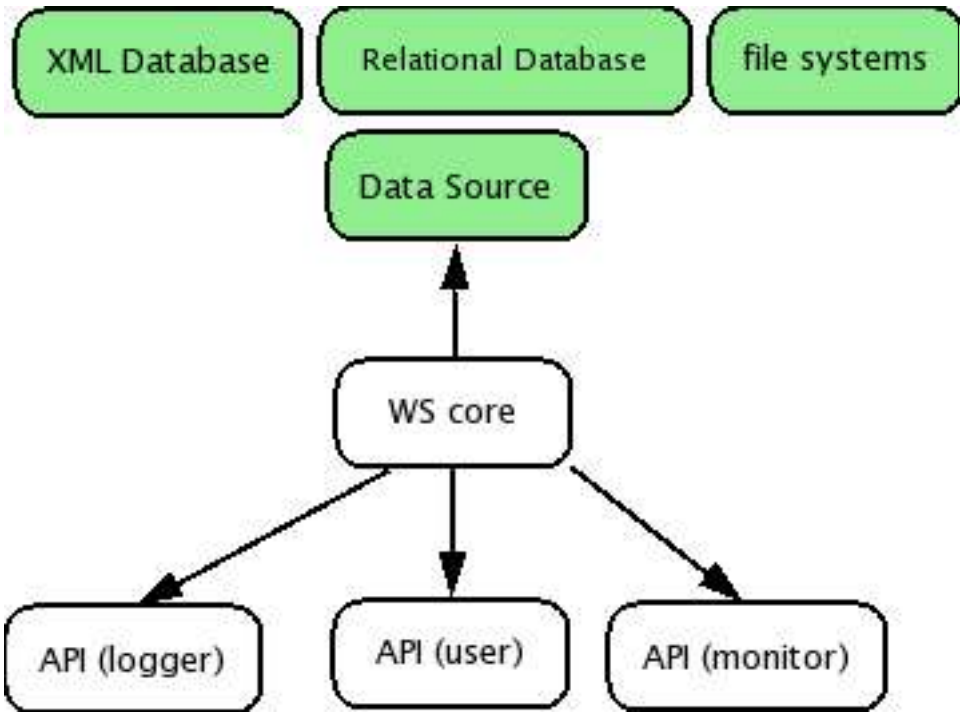
- STAR scheduler (SUMS) : Java
- FileCatalog module : Perl
- StDb : C++

Expected benefits:

- Clients would use single API implemented in native language (Java, C++, Perl) : simple maintenance
- Multiple database support: should be able to work with any RDBMS (MySQL, Oracle, etc.)
- API designed to be used within GRID infrastructure.

green – exist, orange – my part of project, other – to do

Possible future: Data Model Independence



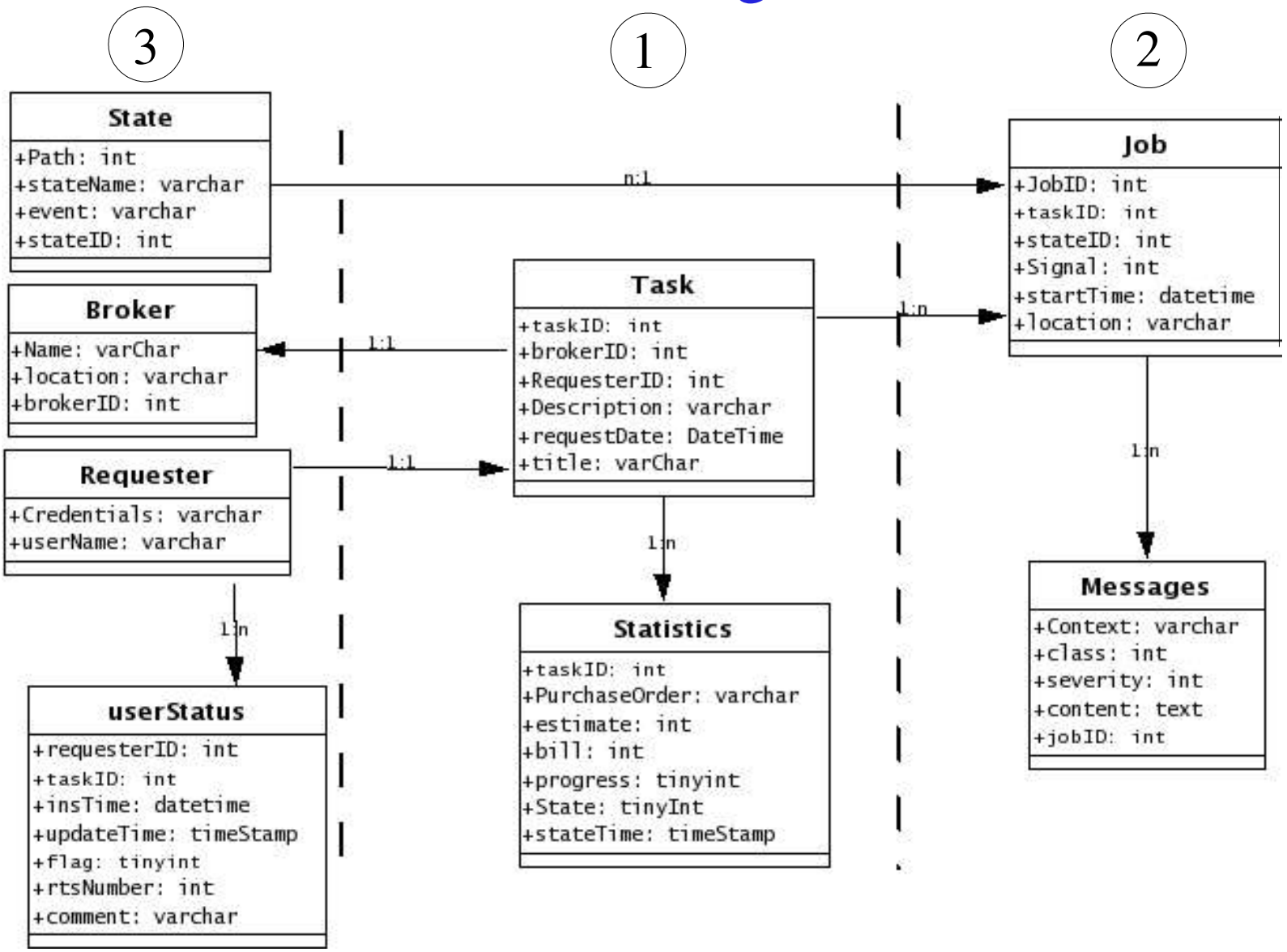
Picture 4. Project schema

- Grid Database Service Specification (GDSS) defines a Grid Data Service
- Grid Data Service (GDS) standard for accessing and integrating data stored in multiple types of data storage systems
- Data source: relational databases, XML databases, file systems

Example: OGSA-DAI Project,

would be used in my project at later stage

Job monitoring: data base schema



Three level DB schema:

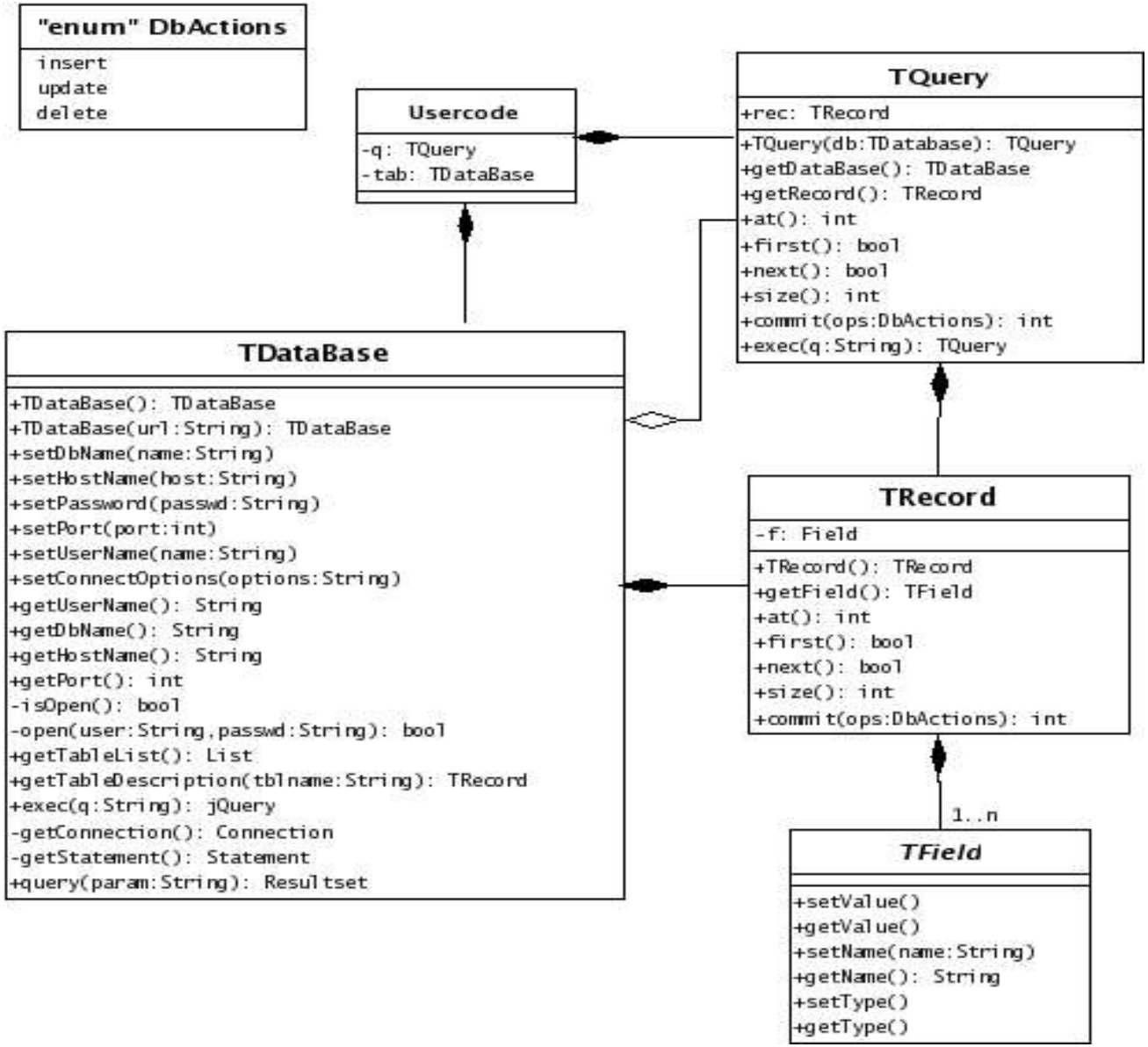
1 – Task level

2 – Job level

3 – Dictionaries level

Picture 2. Data base (“logger”) schema.

DB abstraction layer API: class diagram



Picture 3. Class diagram

Use case 1 (read information from DB)

Use case 2 (insert)

Use case 3 (update)

1. To be implemented in C++, Java, Perl

2. Current status: Java implementation and tests in progress

Summary

- Analysis of existing job monitoring systems was conducted
- Tracking and monitoring requirements were developed
- Job monitoring DB schema (“logger”)
- DB abstraction layer API design started: first steps done.

Work will be continued in Dubna.

PPL-STAR review deadline: 12/31/2006

Project links

jMonitoring package available at

http://www.star.bnl.gov/~rogozin/jMonitoring

jMonitoring documentation available at

http://www.star.bnl.gov/~rogozin/jMonitoring/dist/javadoc

MySQL Connector/J (the official JDBC driver for MySQL): mysql-connector-java-5.0.3.tar.gz is in jMonitoring package

j2sdk1.4- Java Software Development Kit (SDK) available at

http://java.sun.com/j2se/1.4.2/download.html