

## Remote Dataset filtering and retrieval using a database

### 1. Scenario Overview

This is the first and most basic scenario in a collection of data grid use cases of (I think) ever greater complexity. It deals with selecting a subset of a large remote dataset and moving the result to the user. The important feature is that the format of the dataset is assumed to be so complex that "standard" (if they exist) data grid filtering services will not suffice. Instead the owner of the data is assumed to make services available for filtering the dataset. The original use case (in the unpublished Appendix to the GACG proposal) dealt with file based datasets and publication of subroutines as custom grid services. That one will be described in another document, with the title: "Remote Dataset filtering and retrieval using special purpose grid services" (to be written).

I am here adding a database based filtering using SQL, temporary storage in a MyDB and standard or custom retrieval of the query result.

#### 1.1 Background and Purpose

The result of post-processing the results of a large scale cosmological simulation (Millennium simulation, Springel et al 2005), has produced various synthetic catalogues which are available both as a collection of binary files with a complex structure and in a relational database. We wish to provide users having various types of privileges remote access to the data and allow them to define subsets by various means of filtering mechanisms: users of the database will be allowed to define SQL queries and execute these through a web based interface (portal or web service), users of the files will somehow need to get access to a set of special-purpose routines for subselecting parts of the data.

The main datasets of interest for this scenario are synthetic halo and galaxy catalogues, where the latter may have spectra attached to them.

Examples of scientific cases:

- calculate spatial N-point correlation functions of simulated objects as a function of, for example, luminosity
- compare results of the simulations to similar results of observations, for example obtained by querying the SDSS mirror database at MPA.
- etc.

#### 1.2 More information

- The Millennium simulation is described in: Springel et al., Nature, Volume 435, Issue 7042, pp. 629-636 (2005).
- A reference to the construction of synthetic galaxy catalogues is: Croton et al, 2006, Mon. Not. R. Astron. Soc., 365, 11-28 (2006)
- For an example of the SQL query interface on a small version of the Millennium simulation see <http://www.g-vo.org/mpasims>

-

---

## 2. Current Scenario description

---

Currently GAVO has implemented a web application giving access to a small scale version (1/500 times the number of particles) of the Millennium simulation. The data is stored in a SQL server relational database. The user can go to the GAVO website to get a web interface (see <http://www.g-vo.org/mpasims/QueryManager?>). Queries can be defined and results are returned directly to the user. The user can simply retrieve the dataset in a desired format or use a VOPlot applet to visualise the result on the client side. As this is a public website without user registration, limits are placed on the maximum result set size (10000 rows) and the maximum query time (60s).

We are currently experimenting with a query interface that requires authentication but then gives access to a local database of maximum size 1 GB "next to" the main database. Registered users can query the main database and store the results in the user's own database (using `SELECT ... INTO MYTABLE ..`). There is no limit on the number of rows that can be returned now, though the MyDB size limit of 1GB gives some restrictions, but a 60s query timeout is still in place. The reason for the latter is that we at the moment do not have a mechanism for users to interrupt a query once it has been submitted.

### 2.1 Environment

#### 2.1.1 Hardware

Describe the hardware resources that are currently used.

- Processing

- . single PC with (soon) 3 opteron processors for database, one simple PC with 1 processor for web server

- Storage

- . 10Tb RAID10 disk system with 3 controllers

- Network

- . Currently needs to handle a maximum of 10000 rows of any width.

#### 2.1.2 Software

- Describe used software such as operating system, software libraries, e.g. HDF5-plugin for GridFTP, ...

- . Database on Microsoft Windows 2003 server
- . Microsoft SQLServer database
- . Apache Tomcat web server on RedHat Linux

- Web application is implemented in Java.

- How is the program deployed?

- . pre-compiled binaries, ...

- How is the program compiled?
  - . Web application built using Ant
- State the program license and any commercial 3rd party licenses.
  - NA

## 2.2 User Interaction

Describe the user interaction necessary for starting the program and additional interaction with a running program.

### 2.2.1 Initiation

User initiates session by pointing web browser to website.

- compilation (cf. Section 2.1.2),
  - None
- Where is the program executed?
  - . mainly on the database, some web server action, VOPlot visualisation applet runs on client and can not handle very large datasets

### 2.2.2 Monitoring/Steering/Visualization during the run-time of the program

- What type of data is produced by the program during run-time used for monitoring/steering/visualization?
  - . log-files are produced by the web application, but only about certain state changes, user login, error messages, etc. Steering is not available.
- What methods/tools exists for accessing data produced by the program during run-time?
  - . none
- Does your application support any standard for monitoring/steering?
  - . log4j "standard" for Java logging
- Describe any security measures related to program access for monitoring/steering/visualization.
  - . Maximum row count and query timeout are set. No other measures available on public website.
  - . On the "MyDB" prototype there is a user login with a web server based session management allowing single sign-on.
- Who can access the running program OR run-time produced monitoring data?
  - . only me so far, and anyone else with direct access to the machine where the web server is running.
- From where can run-time produced monitoring data be accessed?
  - . the machine itself only.

- How is the program termination detected?
  - . NA
- How much monitoring data and how often is monitoring data transferred during a program run (min/max/avg)?
  - . NA
- Does your program generate metadata and stores this externally (e.g. in a catalog)?
  - . A VOTable can be produced which is self describing, i.e. contains metadata about the produced data contained in the same file.
- Who accesses this metadata? From where? Does your program access metadata generated by other programs?
  - . NA
- How many executions/jobs must be monitored/steered in parallel? By how many users?
  - . As many users as are connected to the web server/database.

## 2.3 Input

### 2.3.1 Parameters

- . An SQL string and a single parameter indicating how many rows of the result set should be shown on the browser screen. After the result set is available (cached in memory on the web server session, the user can retrieve the data in the desired format, one more parameter.

### 2.3.2 Input data

- How is the input data prepared?
  - . I interpret this as meaning, "how is the database instantiated": by hand. There is no further input data.
- Are file-names known in advance (before the program is started)?
  - . Table names are known, or at least given on the help page of the web portal.
- Are data locations (directory, server, ...) known in advance?
  - . Yes, web portal knows where database is. Will be stored in metadata, not yet done.
- Describe the different ways data is accessed.
  - . SQL
- Non-file based data access (XML, database, ...) should include description of
  - . name of the database management system: Microsoft SQLServer

- . how the database is accessed : JDBC via WWW interface
  - . typical access patterns (bursty, continuous, ...): bursty
  - . physical location/distribution (local, externally, ...): close to the web server, which has to make a JDBC connection.
  - . possibility to replicate the data through some mechanism,
  - . any security related restrictions,
  - . are user-defined Stored-Procedures used,
  - . types of indices used (e.g. Hierarchical-Triangular-Mesh (HTM))
  - . ...
- How much data is accessed at each run?
    - . number of files/data sets: min/avg/max,
    - . total-size: min/avg/max,
    - . retrieved-size: min/avg/max
  - Is it possible that a data set/file is accessed multiple times over a short period of time?
    - . For example by different "threads" of the program. Then, replicating and/or caching might be interesting.

NA, if same table is queried database system is doing some caching.
  - How many users are using the same data simultaneously? Unknown  
Are these users geographically distributed? Yes
  - Elaborate on the use of metadata related to input data.
    - . amount, small, Database system tables.
    - . how it is accessed, by database system
    - . security restrictions, NA
    - . metadata format [key/value ?], NA
    - . life-cycle: creation, usage time, used by multiple program runs, deletion, ... NA

### 2.3.3 Additional Notes

Describe any additional information regarding the input data which has not yet been covered.

## 2.4 Output

This covers what data products are generated (INTERMEDIATE and FINAL results), where they are generated and how they are handled after the program finished (transferring data or removing it, ...).

### 2.4.1 Output data

- Where is the output data stored? Describe all centralized or distributed locations.
  - . local file system, remote file server, RAID, Database, ...
  - Database owned by user, "My DB"
- How is the output data structured?
  - . single file, distributed file, database table(s)...

one or more tables

  - . data formats: plain text, HDF5, FITS, VOTable, proprietary, ...

database tables -> proprietary

- Describe what happens when the program finishes? How are the results used?
  - . remains at the output location,
  - . moved/copied/deleted elsewhere [manually/automatically],
  - . used as input to a subsequent call or to another program, ...

May remain at output location as tables for a while, but are likely to be moved to user's local file system after a dump of a table for example to a file on the database machine's file system.

- Describe the different ways data is created/changed.
  - . POSIX write, XPath, SQL, ...

SQL

- Non-file based data access (XML, database, ...) should include description of
  - . name of the database management system,
  - . how the database is accessed (ODBC, JDBC, WWW interface, command line, ...),
  - . typical create patterns (bursty, continuous, ...),
  - . physical location/distribution (local, externally, ...),
  - . possibility to replicate the data through some mechanism,
  - . any security related restrictions when data is written,
  - . ...

Same as input

- How much data is written by the program at each run?
  - . size: min/avg/max,
  - . number of files/data sets: min/avg/max

Varies too much. Likely order of  $10^6$  rows or more for this scenario to be interesting. Corresponding file sets ~ 1GB.

- Describe the parameters which influence the amount of data and number of files/data sets generated.

The query, hard to tell in advance.

- Elaborate on the use of metadata related to output data.
  - . amount,
  - . how it is accessed,
  - . security restrictions,
  - . metadata format [key/value ?],
  - . life-cycle: creation, usage time, used by multiple program runs, deletion, ...

Not really applicable

Note, the decision where results are stored is supported by information about the further use or free data storage.

#### 2.4.2 Additional Notes

Describe any additional information regarding the output data which has not yet been covered.

#### 2.5 Information resources

Give a summary of each information resource that is accessed by the program. Include information about data input/output, locations, access methods (XQuery, SQL, ...), security related restrictions, search of metadata (exact key search i.e. "ABC", range queries i.e. "AB\*", ...)

#### 2.6 Data Stream Management NA

Definitions:

Data Stream - intermediate results can be processed by the subsequent processing module before the current module has processed the last element of the input.

- Can single operations be performed on any compute node or do they need special hardware or software?
- Are data exchanged between distributed parts of the application? does this happen at the beginning, during run-time or at the end?
- Are operations compute intensive?

#### 2.7 Resource Security and Access Restriction

Describe all security related information that considers access of resources. User based, Group based, by IP-address/netmask, certificates, nodes/resources within a private network, firewall restrictions, ...

Users are given access to a Database through a web based interface (portal and or web service) which does the database login, i.e. they do not have direct access to the database and are not registered there.

#### 2.8 Additional Information

Give additional information not covered by the sections above.

- How are workflow/pipeline steps interrelated to each other?

NA

- Is the application executed in several phases where each phase may have different resource requirements or may be executed at a different resource?

No

- How long (avg) does the scenario execute (minutes, hours, days)?

Currently due to timeout restrictions, queries can not last longer than between 1-5 minutes. Result sets are restricted by the size of the My DB user database which is about 1 GB.

- How often will the scenario be executed?

Currently not known, depends on number of users.

- Are the executions time-critical?

Main time will likely be spent in the database query execution.

-----  
3. Future Scenario and AstroGrid-D Usage  
-----

Describe the future scenario and envisioned usage of AstroGrid-D as detailed as possible. It is not assumed that the questions can be answered as detailed as in Section 2. Focus on what is expected by the Grid environment and how this new functionality can be used.

Note, there is no special section to describe workflows/pipelines or details about a phased execution of a program. If your scenario is a workflow/pipeline OR your application is executed in several phases, describe EACH step/part covering the sections 3.1 - 3.5. In addition you must describe how these steps/parts are interrelated to each other (in Section 3.6).

### 3.0 General goals

Want to scale up to the Full Millennium simulation. This dataset is 500 times the size of the current milli-Millennium. Queries will last longer and need to be executed asynchronously, with data results stored in a user specific "My DB" database attached to the main database server, and afterwards they should be enabled to download this data in some convenient, hopefully griddified way to a location local or elsewhere on the grid where they will do further processing.

Such as:

- use more compute resources,
- use other data resources,
- provide data to other users,
- completely new scenario,
- overcome deficiencies of current approach,
- ...

### 3.2 Environment

- Are there any constraints due to your participation in other projects or international collaborations?
  - . specific Grid middleware, hardware, standards, virtual organizations

Main data must be stored in the SQLServer database at MPA which owns the dataset. Otherwise none.

### 3.3 User Interaction

- Which parts should be automated?
  - . resource selection,
  - . data transfer before initiation and after termination,
  - . ...

user registration/authentication  
My DB creation  
data transfer after termination

- Which user interface are you planning to use?
  - . WS [SOAP], API, WWW portal, ...

WS, www portal, wget

- Are you planning to use any standard for application monitoring/steering?
  - . Do you want to use such standards in collaboration with the DGI or the other communities OR will you develop your own methods?

Don't know. If they are available yes.

- Aspects of a Portal / WWW based interface:
  - . Which portal features are mandatory/optional (e.g. credential management, job management, job monitoring/steering, data transfer, ...)?

. How are user managed? Where is information about users defined / stored?

User credentials probably stored on a database. This will include limits on result sets and maximum query times.

. Which authentication/authorisation methods are needed ?

No special requirements. Is only necessary if the service is popular.

. Do you want to access specific data services (web services, databases, etc.) via a portal?

Millennium database

. Are there any existing programs, on which the user interface should be based OR which should be replaced by the portal?

Yes, <http://www.g-vo.org/mpasims> is an example. We may want to keep this unless a better alternative comes along. In particular we may want to adapt a version of the SDSS skyserver web site.

. Should there be a central AstroGrid portal OR do you want to set up a portal server for each scenario/application ?

No central AstroGrid portal for this.

. Does the scenario require any special interfaces OR is it sufficient to use generic interfaces ?

JDBC is used, for the rest it is desirable that result data can be visualised

online. Currently done using a VOPlot applet. This has memory restrictions.

- Aspects of a generic Grid Application Programming API (GAT)
  - . Which GAT functionality would you like to make use of (e.g. job submission, file handling, resource brokering, etc.) ?

Don't know yet.

. What programming languages must be supported ? Which platforms ?

Web app currently written in Java. Currently deployed on Apache Tomcat web server on Linux machines. May/will go to a Windows server.

. Which Grid Middleware should be supported (Globus, Unicore, gLite, etc.) ?

No particular preference or unique requirement.

- . For specific GAT functionality, which protocols/packages/tools should be supported ?
  - e.g. for job management: clusters with PBS, SGE, Condor

We want asynchronous/batch-like behaviour where users put their request in a queue and get notified or get a portal when it is done.

### 3.4 Input

- Do you handle input data manually or do you need an automated management of data?

There is not really input data.

### 3.5 Output

- Do you handle output data manually or do you need an automated management of data?

Would be good if user could get a view in his/her personal database (My DB) showing the tables that are available etc.

### 3.6 Additional Information

- How long (avg) does the scenario execute (minutes, hours, days)? Do you aim at a specific speedup?

Queries of around an hour should be supported.

- How often will the scenario be executed?

Frequently hopefully.

- Which restrictions of the current approach (as described in section 2) do you want to overcome?

result size and query timeout limitations, synchronous behaviour, moving result sets from user database to user's file system

## 4. Bigger Picture for the far future

NA for now.

#### 4.1 Organization of Multiple Runs

Maintain a list of all simulations, to repeat simulations with a different binary, with different input data, to check if a program was already executed with a certain set of parameters/input data, ... .

#### 4.2 Handling relationships between data products

For example, store metadata on how a data product was generated (from which input data, by which program, with which parameters) and how it can be used by others.

#### 4.3 Constructing More Complex Runs

For example, combine existing single programs.