Distributed Database Management Systems and the Data Grid

Heinz Stockinger

CERN, European Organization for Nuclear Research, Geneva, Switzerland
Inst. for Comp. Science and Business Informatics, University of Vienna, Austria

Outline

- Introduction: DB – Data Grid
- Related Work in Both Communities
- Replica Catalogues & Directory Service
- ODBMS
- Implications for Grid Applications
- Data Consistency and Replication Methods
- Update Synchronisation
- Conclusion

Heinz.Stockinger@cern.ch

Introduction

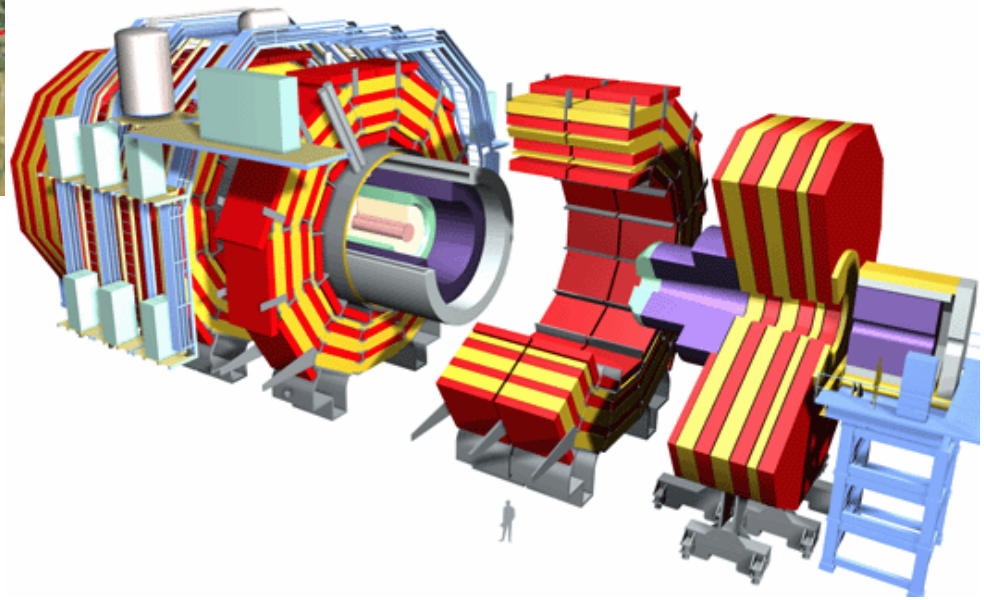
- Computational Grid vs Data Grid
  - scientific applications like data analysis in High Energy Physics (HEP), climate modelling or earth observation
- focus on the data intensive Large Hadron Collider (LHC) experiments of CERN – **DataGrid project**

- store Petabytes of persistent data
  - **OO database – Grid middleware**
- Grid research as well as distributed database research tackles the **problem of data replication** but from a different point of view

Heinz.Stockinger@cern.ch

CERN: European Organization for Nuclear Research

over a **Petabyte of data** per year
several thousand users



Heinz.Stockinger@cern.ch

DB vs Data Grid

- Data Grids are very new in the research community
  - identify the **characteristics** and **requirements** of Data Grids and how they can be met in a most efficient way
  - Special attention: **data consistency** and **communication issues**
- **Optimising data replication** and **access** to data over the **WAN**
  - not addressed sufficiently in database research
- In DBMS normally only one method for accessing data
- We elaborate on
  - different **data consistency models**
  - **global transactions**
  - **asynchronous replication**

Heinz.Stockinger@cern.ch

Related Work in DB Research

- Replica synchronisation is based on relatively **small transactions**
  - In **HEP** relatively "**large**" transactions
- Synchronous and asynchronous replication
  - evaluation techniques mostly based on the amount of communication messages
- Cost functions for network or server loads are rarely integrated
- Rather low amount of data as compared to HEP
- Jim Gray "**Distributed DBs over WAN don't work**"

Heinz.Stockinger@cern.ch

Related Work in Grid Research

- Globus project
  - Replica catalogue & replica management
- Replication mostly based on file level
  - Is easier than object replication
- Related projects in HEP (Europe – U.S.):
  - EU DataGrid – GriPhyN – PPDG (Particle Physics Data Grid)

Heinz.Stockinger@cern.ch

What to use ? Commonalities?

- Introducing a **replication middle-ware** layer that manages replication of files
- Each site will manage data locally with a database management system
- In Grids there are many tools for monitoring applications and network parameters
- **Hybrid solution**
  - But: restrictions for update-synchronisation
  - Relaxation of transparency and consistency

Heinz.Stockinger@cern.ch

Replica Catalogues & Directory Services

- Access to replicated data requires specific **data and meta data structures**
  - object location table vs replica catalogue
- Globus proposes an LDAP replica catalogue (RC)
- Generic RC can be used for any DBMS and file format
  - Manage replicas
  - Heterogeneous data stores
- **Expose RC** to large user community
- Communication protocol is necessary for the exchange of control messages

Heinz.Stockinger@cern.ch

Objectivity/DB

- Take Objectivity as a sample **ODBMS**
  - Many issues raised are specific to ODBMS
- Data Replication Option (DRO)
  - Synchronous replication
    - populate-replicate vs replicate-populate
  - Dynamic quorum
  - Not optimal for WAN
  - No optimisation for replica access (replica selection … like in many commercial DB systems)
- **Partial replication and associations**
  - Has important impact on the replication granularity
- **File catalogue**
  - Replica catalogue is required

Heinz.Stockinger@cern.ch

Implications for Grid Applications

- Users need to we aware of Grid – **sociological aspect**
- **Access replicated files**
  - "Grid open" needs a lookup in the RC
  - Cache files locally ?
  - Add files to the catalogue ?
  - Transfer entire file vs sub sets of files
- **Pre-fetching/read ahead**
  - Query optimisation problem (replica selection)
  - Application can give hint to pre-fetch files
  - Potential for reservation

Heinz.Stockinger@cern.ch

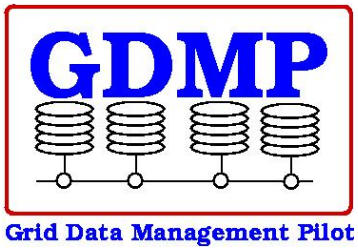
Data Consistency and Replication Methods I

- Consistency: one of the main issues of replication
  - Easy for read-only data
  - Depends on the frequency of updates and the amount of objects
- **Synchronous replication**
  - Highest degree of consistency (locally and globally)
  - 2-Phase Commit protocol for each write transaction
  - “bad” response time for write transactions
  - Consistency level has to be adapted to the application
  - A Data Grid needs to have several levels of consistency
  - Middleware: difficult to provide very high consistency
  - Global transactions required

Heinz.Stockinger@cern.ch

Data Consistency and Replication Methods II

- **Asynchronous Replication**
- Better write preformance than synchr. repl.
- **Primary-copy** approach (master slave)
  - Updates only done by primary copy
  - High consistency and improved write performance
- **Epidemic approach**
  - User operations are performed on any single replica and a separate activity compares version information
- **Subscription**
  - Don’t care about consistency
  - Site is free to get data
  - Implemented in GDMP



Heinz.Stockinger@cern.ch

Communication and Transactions

- Clear need for global transactions
- Do not need to lock all sites
- Difference between control comm. and update comm.
  - Control messages
  - Data transfer
- Use most appropriate protocol for specific use
  - Message passing library
  - FTP, GridFTP
- In ODBMS no such separation

Heinz.Stockinger@cern.ch

Append Transaction

- Enhance the traditional DBMS transaction system
  - Read – write – **append transactions**
- Write can either create new data or change existing data
  - Both transactions require different tasks
  - Append transaction only has to satisfy a **uniqueness condition**
  - Easy to satisfy for files
- Append transactions don’t require locks:
  - Different response time and consistency levels

Heinz.Stockinger@cern.ch

Update Synchronisation I

- Difficult for a middleware to do replica update at the object level
  - Cannot access single pages or object tables
  - Communicate only the differences between files

- **Binary difference approach**
  - DB file appears like a binary file
  - Find out about changes between old and new files
  - XDelta can be used for that

Heinz.Stockinger@cern.ch

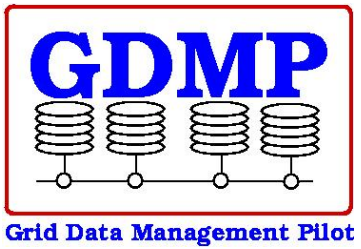
Update Synchronisation II

- **Object-oriented approach**
  - Create objects that are aware of replicas
  - **`object.create (site1, siteX, siteY);`**
  - Similar to stored procedure approach in RDBMS
  - Update synchronisation is done by a replicator object
  - **`object.update_parameter_x (200); // OID = 38-23-222-442`**
  - Exclusive look is required

Heinz.Stockinger@cern.ch

Conclusion

- data management efforts of the two research communities are combined
- it is appropriate to try to understand the research issues of both communities since data is stored in databases
- combine common ideas to form an efficient Data Grid
- first basis for such an effort
- more about Data Grids
  - **http://www.EU-DataGrid.org**
  - **http://cmsdoc.cern.ch/cms/grid**
  - **http://www.GriPhyN.org**
  - **http://www.PPDG.net**



Heinz.Stockinger@cern.ch