

# Database Cluster for e-Science

**Dr Michael Doherty**

e-mail: m.doherty@rl.ac.uk

Data Management Group, CCLRC e-Science Centre, Rutherford Appleton Laboratory,  
Chilton, Didcot, Oxfordshire, OX11 0QX, United Kingdom.

*Kerstin Kleese, Shoaib Sufi*

## **Abstract**

CCLRC have recently introduced a database service for e-Science. This paper outlines the introduction of the service and the facilities it currently provides. We begin with a description of database and high availability options and then move on to the system selected and motivation behind this. Lastly we conclude with a short section on the projects currently using the CCLRC database service.

## **Databases in e-Science**

One of the fundamental building blocks of most e-Science projects is the database system. Databases are used for storing user data, application data and metadata. The Data Management group recognised the need to establish a database service within the CCLRC e-Science Centre. Many of the e-Science groups had a requirement for, or made use of some existing ad hoc databases systems. However no centralized database service existed. The Data Management Group therefore engaged in a project to provide this, which is now up and running with many projects. We describe the different architectures below and indicate the choice for our system.

## **Database Software**

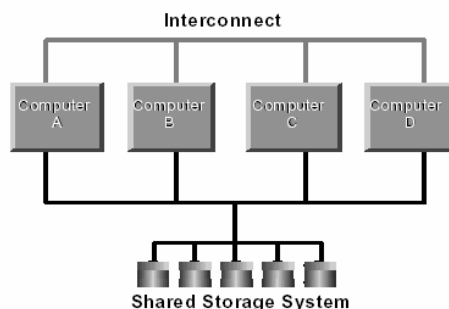
When considering database systems, it is important to decide on which database software should be used. The requirement for most projects is a relational database system (RDBMS). Commercial and community based RDBMS products are available, however for a reliable production service it was felt that an established commercial database should be used in the first instance. However, we did

decide that any system purchased should be capable of running community based solutions at some time in the future.

A key requirement for the service was high availability. This led us into the realm of database clustering. This is discussed in detail below as there are several options each with their own merits and pitfalls.

## **Database Clustering**

Traditionally, a database cluster is a group of independent servers that collaborate as a single system. The primary cluster components are processor nodes, a cluster interconnect and a disk subsystem. The clusters share disk access and resources that manage the data, but each distinct node does not share memory.



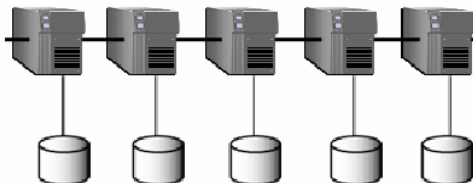
**Figure 1 Basic Database Cluster**

Each node has its own dedicated system memory as well as its own operating system, database instance and application software as seen in Figure 1. In the event of subsystem failure, clustering aims to ensure high availability.

There are 2 distinct database clustering topologies and these are discussed below [1]:

### *Shared Nothing Clusters*

In pure shared nothing architectures, database files are spread across the various database instances (running executable programmes) running on the nodes. Each instance or node has ownership of a distinct subset of the data and all access to this data is performed exclusively by this “owning” instance (Figure 2). In other words, a pure shared nothing system uses a partitioned or restricted access scheme to divide the work among multiple processing nodes.



**Figure 2. Shared Nothing Cluster**

A transaction executing on a given node must send messages to other nodes that own the data being accessed. It must also co-ordinate the work done on the other nodes to perform the required read/write. IBM DB2 uses this model.

### *Advantages*

- This works fine in environments where the data ownership by nodes changes relatively infrequently. The typical reasons for changes in ownership are

either database reorganizations or node failures.

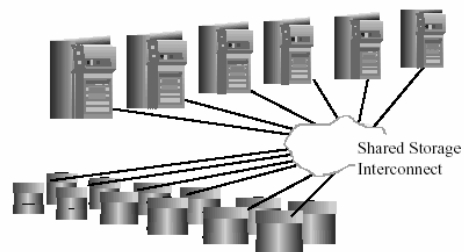
- There is no overhead of maintaining data locking across the cluster

### *Disadvantages*

- The data available depends on the status of the nodes. Should all but one system fail, then only a small subset of the data is available.
- Data partitioning is a way of dividing your tables etc across multiple servers according to some set of rules. However this requires a good understanding of the application and its data access patterns (which may change).

### *Shared Disk Clusters*

In a pure shared disk database architecture, database files are logically shared among the nodes of a loosely coupled systems with each instance having access to all data (Figure 3). The shared disk access is accomplished either through direct hardware connectivity or by using an operating system abstraction layer that provides a single view of all the devices on all the nodes.



**Figure 3. Shared Disk Cluster**

In the shared disk approach, transactions running on any instance can directly read or modify any part of the database. Such systems require the use of inter-node communication to

synchronize update activities performed from multiple nodes. When 2 or more nodes contend for the same data block, the node that has a lock on the data has to act on the data and release the lock, before the other nodes can access the same data block.

#### Advantages

- A benefit of the shared disk approach is it provides a high level of fault tolerance with all data remaining accessible even if there is only one surviving node.

#### Disadvantages

- Maintaining locking consistency over all nodes can become a problem in large clusters.

#### *Oracle Real Application Clusters (RAC)*

Oracle has been selected for the initial database. The Oracle clustering technology is called Oracle Real Application Clusters (RAC) and can only be used with Oracle Server Enterprise Edition (currently release 9.2.0). RAC is based on “shared disk clustering” (see above). At the heart of this is a mechanism to transfer database blocks directly in memory between nodes. This is based on memory cache coherency technology (Cache Fusion), that utilizes the high-speed interconnect. By, utilizing both local and remote server caches, Cache Fusion exploits the caches of all nodes in the cluster for serving database transactions.

#### **Server Operating Systems**

RDBMS systems run on almost every operating system. We looked for the most flexible operating systems in terms of both software available and price/performance [3]. This ultimately led to the choice of a Linux based system. Commercial RDBMS systems are generally only supported on

enterprise editions of Linux. For running the system we selected Redhat Enterprise Server AS edition. At the time of purchase, this was the only version to support Oracle RAC. However this has changed and now Redhat Enterprise Server ES edition is supported.

#### **Database Hardware**

When dealing with RDBMS systems, it is normal to decide on RDBMS software, operating system and then hardware last of all. Having decided on Oracle running on Linux, we examined Intel based architectures. We chose IBM x440 series nodes as the building blocks for the data clusters. The IBM X-Architecture has support for up to 16-way SMP capability and remote I/O. The systems can be partitioned and are designed to be highly expandable. This closely met the requirements for a database service that is likely to grow. We purchased 2 separate clusters, one based at Daresbury Laboratory and the other at Rutherford Appleton Laboratory. The clusters connect to their own 1TB RAID 5 storage arrays via a independent fibre channel Storage Area Networks (SAN). These clusters operate independently of each other and are used for specific projects. While it is possible to combine them, this is not considered necessary at present.

#### **Applications**

The database service is now up and running and has several users. Figure 4 shows the cluster based at Rutherford Appleton Laboratory.

The main projects using the service are:

- CMS Storage Resource Broker Service [4]
- E-Mineral Mini-Grid [5]
- CCLRC Data Portal [6]

- Various Metadata Catalogues for NERC



**Figure 4. CCLRC Data Cluster**

### Future Work

The service is currently open to internal projects and some outside projects. We will expand this with the addition of a larger database cluster in the near future. A significant development this year will be the installation of a 20 node cluster at CCLRC funded by JISC and open to the community via UK Tier 1 Grid access.

The group can also provide database advice to the general community and can be contacted at the address above.

### Conclusions

CCLRC have introduced an e-Science database service based on Oracle RAC and Linux. The service provides high availability and is now holding data for a number of e-Science projects. This number is growing continually and the service will be developed further.

### References:

[1] Technical Comparison of Oracle9i Real Application Clusters vs. IBM DB2 UDB EEE v8.1, Sashikanth Chandrasekaran and Bill Kehoe. An Oracle White Paper October 2002  
[http://otn.oracle.com/deploy/performance/pdf/ibm\\_db2.pdf](http://otn.oracle.com/deploy/performance/pdf/ibm_db2.pdf)

[2] A Technical Discussion of High Availability and Crash Recovery, IBM Software Group Toronto Laboratory. February 2003.  
<http://www-3.ibm.com/software/data/highlights/rac.pdf>

[3] IBM DB2 Integrated Cluster Environment for Linux, Boris Bialek, Rav Ahuja, Solution Blueprint, August 2003,  
<http://www-3.ibm.com/software/data/pubs/papers/linuxcluster/linuxcluster.pdf>

[4] SRB in Action, Michael Doherty, UK e-Science All Hands Meeting, 3-5 September 2003, Nottingham, England

[5] The E-Minerals Mini Grid, Kerstin Kleese, Lisa Blanshard, Richard Tyer, May 2003,  
[http://ws1.esc.rl.ac.uk/documents/staff/kerstin\\_kleese/e-min\\_intro.doc](http://ws1.esc.rl.ac.uk/documents/staff/kerstin_kleese/e-min_intro.doc)

[6] The CCLRC Data portal, Glen Drinkwater, Kerstin Kleese, Shoaib Sufi, Lisa Blanshard, Ananta Manandhar, Rik Tyer, Kevin O'Neill, Michael Doherty, Mark Williams, Andrew Woolf, [http://www.e-science.clrc.ac.uk/documents/projects/dataportal/DataPortal-WebServices0603\\_Glen.doc](http://www.e-science.clrc.ac.uk/documents/projects/dataportal/DataPortal-WebServices0603_Glen.doc)