Efficient Data Sharing Within Database Clusters

Miroslav Křipač

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00, Brno, Czech Republic
kripac@fi.muni.cz

Abstract. Database clusters meet current demands for scalability and high availability of mission critical systems. However, conventional implementations of single-purpose shared disk subsystems limit their widespread adoption primarily for cost reasons. This paper focusses on an alternative software approach to data sharing methods and its impact on system performance. Reports show that presented solution was successfully used as a part of several commercial projects.

1 Introduction

There are two main reasons for implementing clustering techniques into production environment nowadays — high availability and performance. High availability which is an important issue especially inside database systems is dependent on special system architecture. Unlike other distributed computing methods in high availability database clusters all the data must be available at all times at all nodes [2]. For example should a failure occur on node I states of its transactions must be accessible to other nodes so that node II can take over operations of node I transparently. Because of this aspect of data availability additional expensive mechanism known as a shared disk subsystem has to be added to the setup [3]. This subsystem provides physical access to data and system meta-data to all nodes connected into the cluster.

Implementations of current database clusters require expensive[1] hardware disk arrays. This technology generally provides enough throughput for simultaneous access of all cluster nodes. However, it can be very difficult to purchase this single-purpose hardware solution because its price usually exceeds typical budget of small and medium projects. Hardware shared disk subsystems lock projects into their suppliers, because they pose a long term financial liability (warranty fees and not easy migration paths).

In this paper we shall describe a practical solution that brings an ability to support efficient data sharing within high available database clusters under Oracle Real Application Clusters (RAC) without a price tag. RAC is a successor of Oracle Parallel Server [8] and requires transparent data sharing subsystem

[1] Price for Dell/EMC CX400 SAN Storage System recommended for Oracle Real Application Clusters environment starts at USD 66,000 for a 180GB configuration as of July 2003 (source: Dell.com).

provided by operating system as raw disk devices. The exact type of shared disk solution is not mandated therefore any possibility can be used as long as it provides the core raw device features.

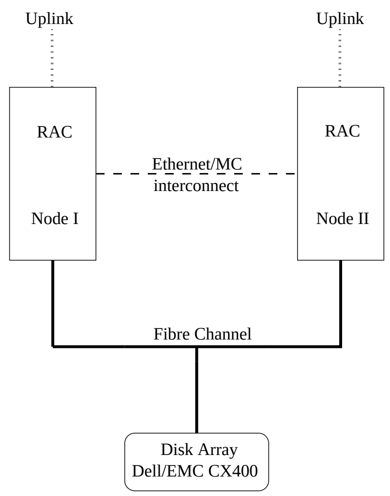

Fig. 1. Typical two-node Oracle RAC configuration.

Figure 1 represents a typical RAC configuration with two nodes. The RAC nodes are accessible from outside via uplink connection usually an Ethernet network. All cluster nodes are also connected via Ethernet or MemoryChannel (MC) to ensure internode communication. The third connection at the bottom provides access to shared disk arrays for both nodes usually via FibreChannel (or via other possibilities such as SCSI interconnect or IEEE 1394 FireWire) connection.

The goal of this paper was to provide an alternative solution of shared disk subsystem by using software emulation of highly priced hardware disk arrays built on common non-proprietary components. In this case, we use standard technology Network Block Device (NBD) which has been integrated into Linux operating system and generally provides efficient data sharing over common networks. Unlike Network File System (NFS) or other network sharing techniques, NBD is not a filesystem therefore it can be integrated to Oracle RAC instances transparently as a raw storage.

2 Network Block Device Technology

Network Block Device is an integrated part of Linux operating system providing a transparent block access to data physically stored at other machine over the network. This client-server architecture allows one or more clients to access data stored on one server. However, support for locking and parallel access management is not a part of the protocol and must be handled by the application layer.

2.1 Network Block Device Server

At the server side NBD technology is implemented by user-space daemon that listens on specified TCP port and handles communication with clients at the other side. Each NBD server process serves one file stored in local server's filesystem. Thus, should the NBD system provide access to three datafiles, three NBD server processes must be started on different ports.

As the server is a user-space process which performs only standard network and filesystem operations it is possible to compile and run it on several operating platforms. A whitepaper with technical description [5] is available explaining in detail aspects of NBD server implementation under Sun Microsystems' Solaris operating system. That shows the capability of Network Block Device implementation in different heterogeneous systems. For example, you can use your current trustworthy Sun UltraSPARC data server as the NBD server hosting platform so that your data is kept stored safely even if the rest of the cluster is implemented using common cost-efficient Intel-based hardware.

2.2 Network Block Device Client

Unlike NBD server the NBD client is implemented as an operating system block device driver and is currently supported only on Linux operating system. This means that all clients (RAC nodes in our case) have to be implemented on Linux platform. Architecture of NBD client is also different. The driver is usually implemented as a Linux kernel module which means its code is executed as a part of kernel in kernel space mode and can be loaded on demand. NBD client kernel module provides standard block device operations to applications. On the other side Linux kernel module cooperates with another user space processes which perform network operations and communicate with the NBD server.

Once both server and client are set correctly appropriate block device on the server side provides data access to datafiles stored at the server side. Detailed explanation of Linux block device access is described in [1].

2.3 Using NBD as an Oracle RAC Shared Disk Subsystem

The NBD technology as described above can also be used as the data access provider for Oracle RAC implementation. Detailed instructions of implementation including all necessary steps for correct configuration are described in [5] and some other aspects of this implementation are also included in [6]. Brian Hengen published his point of view [4] and followed up the original project while researching general cost efficient methods of RAC implementation.

NBD implementation for Oracle RAC requires a set of naming conversion. Local instance datafile name is mapped in the file system to a raw device which is converted into appropriate block device inside the Linux kernel. This block device is then served by the NBD subsystem. This configuration allows Oracle RAC instance to store its local data and metadata (including system and temporary files) remotely over standard TCP/IP network. Thus, expensive hardware disk array can be replaced by any other common technology (such as Intel-based computer with an Ethernet interface) and save significant amount from limited budget.

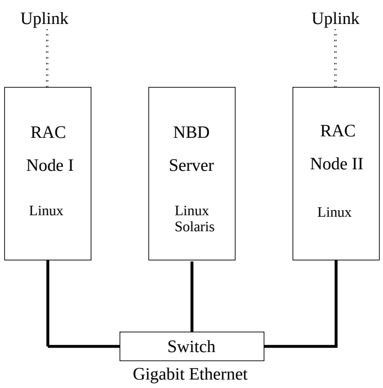

Fig. 2. Network Block Device configuration of Oracle RAC.

Unlike typical solutions NBD configuration allows cluster nodes to be connected via one Ethernet network. The disk array is substituted by a common based Ethernet switch and a Linux or Solaris machine takes the role of the NBD server.

3 Performance

The motivation for this project was to provide a cost efficient and easy to use solution for testing Oracle RAC fundamentals. Proving its viability for production deployment would be a nice add-on. It is important to note that even IEEE 1394 FireWire is endorsed by Oracle Corporation [7] only for testing of Oracle RAC not for production environment.

In our NBD-based software solution high speed interconnect was replaced by definitely slower Ethernet environment[2]. Also, latency and throughput of both communication methods differ and Brian Hengen notices [4] that subjectively perceived performance of this solution is bad (even if he does not provide exact measurement). The slowdown is however expected and we do not try to exceed

[2] According to technical specifications theoretical capacity of high speed FibreChannel interconnect is up to 2 Gbps; theoretical maximum capacity of current commodity Ethernet solutions is 1 Gbps.

physical limits of the media. Nevertheless, we can try to identify possible configurations and usage patterns of Oracle RAC where NBD can be used as the shared disk subsystem without any significant performance loss. Here, we will discuss these types of systems which showing that Oracle RAC with NBD is not designed only for testing purposes.

One of these situations are systems where the number of disk operations is relatively small. When only a small fraction of system's operation is spent in disk input and output the slowdown impact is minimized. Online Transaction Processing (OLTP) Systems that work on only a small portion of the whole data set are naturally better candidates than Data Warehousing (DW) applications where terabytes of data need to be fetched from disk storage and processed often in full-table access manner. Small working data sets means that the data can be cached on clients and therefore disk read operations are eliminated.

Web-based systems are ideal candidates sporting the above-mentioned OLTP characteristics. Number of read accesses often exceeds writes by orders of magnitude and at any given point of time interests of users of such systems focus on limited data areas thus reducing the scale of active data stored in clients' cache. Based on above premises, we shall explore medium-scale production Web-based administrative system.

Information System of Masaryk University (IS MU) is a Web-based system supporting study and researched based agendas of Masaryk University in Brno, the Czech Republic. It serves more than 10 thousands unique users per day driving HTTP traffic to 100 thousands hourly peaks[3]. Following tests analyze real-life operations of IS MU and compare them with NBD-based Oracle RAC environment loaded with Web traffic patterns taken from the production system.

3.1 Testing Based on Simulation of Real Service

To improve usability of NBD technology within Oracle RAC specialized test environment was developed. Its purpose is to simulate real system usage. Details of each HTTP requests placed on IS MU is stored for potential future processing (statistics, logs of operations of individual users, auditing). Logged HTTP requests can be replayed on the testing environment by sending them to the HTTP server of the test system. As the client side of the testing suite sends these requests in the same order in which they were served by the production system load of all parts of the test system is distributed as in the real run. Even if this model does not reflect different conditions of real clients (different network connections), experience shows that this type of testing emulates real work load with a great accuracy.

Tests performed in this testing environment focused mainly on measuring two types of values – CPU load and I/O (disk or network) usage. There are several possibilities to obtain these values. For the NBD tests `proc` filesystem features on Linux and standard `kstat` library on Solaris [6] were used.

[3] IS MU is strictly OLTP system which means that operations are computed online and users are always presented with up-to-date data.

3.2 Tests of Production IS MU

To compare CPU and I/O loads of the proposed NBD-based Oracle RAC setup with the traditional database tests of the production IS MU were needed to receive upper limits of necessary I/O capacity. The first test visualized in Figure

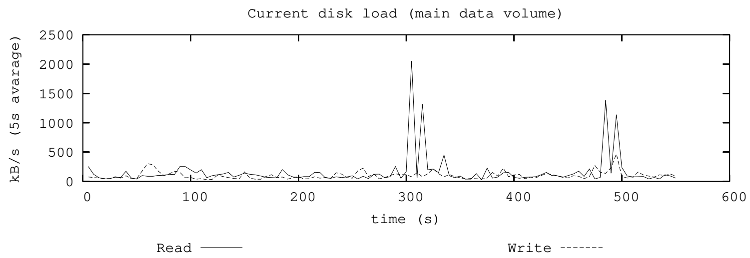

Fig. 3. Sample of disk I/O load of production IS MU.

3 shows throughput of disk operations in kilobytes per second (five second average). This sample was taken during typical working day and shows a snapshot of normal user operation. The peaks of read load are caused by some applications that process large amounts of data (statistics, aggregate operations on many student or course records).

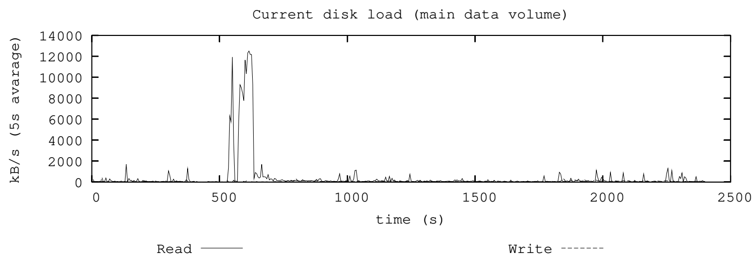

Fig. 4. Sample of disk I/O load with system start (the peak).

Figure3 also shows typical maximal I/O load during common system service which does not exceed 16 megabits per second on top load. Disk configuration on which the test was performed consists of two mirrored FibreChannel disks[4]. If the database system is tuned[9] correctly these disks holding the main data base of the system are permanently idle or with minimal load. This means that

[4] Configuration of IS MU database server: Sun Microsystems' SunFire V880, 4 x Sun UltraSPARC III 750MHz processors, 8GB RAM, 6 x 72.8 GB 10Krpm 1Gbps FibreChannel disks.

the FibreChannel technology could be replaced by lower throughput software solution without any significant performance loss.

While trying to understand the system I/O load we can not reduce our testing process to typical everyday use. From time to time I/O load may rise because specialized data intensive operations have to be carried. Figure 4 shows one of these cases — the database system start.

During startup of Oracle database instance the RDBMS must check integrity of all system logs and datafiles as well as perform other expensive I/O operations. Therefore, load peak demonstrated by this graph is about six times higher than the normal database operation.

3.3 Tests of Network Block Device Setup

Tests of Oracle RAC with NBD as a shared disk subsystem were performed on a two-node RAC configuration. RAC instances were hosted on Intel-based systems with Linux operating system and the NBD server ran under a Sun SPARC workstation with Solaris operating system. The HTTP requests were processed by a cluster of four servers based on Linux operating system and Intel PC hardware platform. All hosts were connected into one 100-Mbps network.

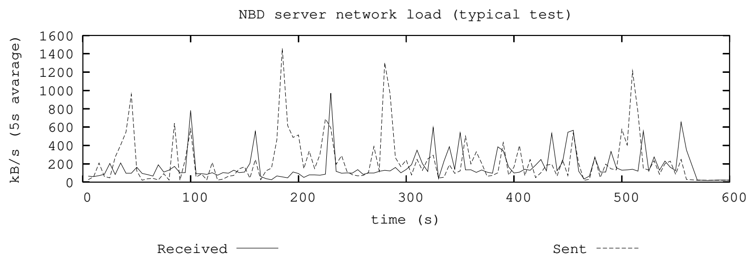

Fig. 5. NBD server network interface communication during typical test

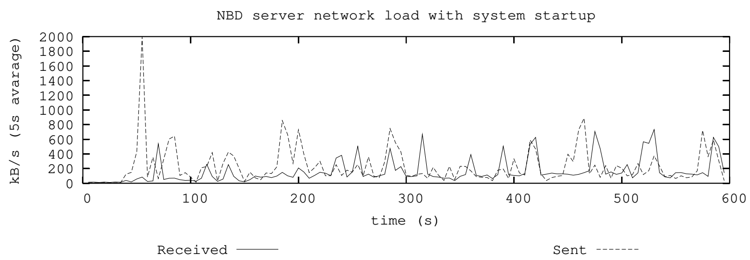

Fig. 6. NBD server network interface communication including system startup.

The network traffic produced by this testing environment on NBD server is shown in figure 5. This essentially represents disk I/O performed by the Oracle RAC nodes against the NBD server. Even if up to 40 thousands requests per hour were reached during this type of user operations the 100-Mbps Ethernet interface connection was able to serve it without performance loss.

Figure 6 shows a startup of Oracle RAC instances. Clustering environment of Oracle RAC configuration is an important advantage for this situation because instances can be started individually and each starting instance performs only limited number of operations while the whole system is running.

4 Conclusion

In this paper we discussed alternative possibility of data sharing methods within contemporary database clusters. The goal was to prove the viability of this solution and possibility of its deployment within specific types of systems such as Web-based information systems. Even if the Oracle RAC implementation configured with NBD technology does not provide such performance as standard hardware data-sharing subsystems it can successfully satisfy common needs of these specific systems.

References

1. Daniel P. Bovet, Marco Cesati: *Understanding the Linux Kernel* O Reilly & Associates, ISBN: 0-596-00002-2, October 2000.
2. Vineet Buch, Sandra Cheevers: *Database Architecture: Federated vs. Clustered*, Oracle Corporation, March 2002.
3. Sashikanth Chandrasekaran, Bill Kehoe: *Technical Comparison of Oracle9i Real Application Clusters vs. IBM DB2 UDB ESE V8.1*, Oracle Corporation, April 2003.
4. Brian Hengen: Developing a RAC System On a Limited Hardware Budget In: *IOUG Live! 2003*, May 2003.
5. Miroslav Křipač: *Implementing Oracle Real Application Clusters Using Network Block Device Technology*, 2002, Technical Solution Description.
6. Miroslav Křipač: *Massive Database Load Distribution on Production Information Systems*, diploma thesis, FI MU Brno, 2003.
7. Caroline Kvitka *Get Fired up About RAC*, Oracle Magazine, May/June 2003, page 12, Oracle Corporation.
8. Tushar Mahapatra, Sanjay Mishra, Jonathan Gennick, Deborah Russell: *Oracle Parallel Processing*, O Reilly & Associates, ISBN: 156592701X, 1st edition, 2000.
9. Richard Niemiec: *Oracle9i Performance Tuning Tips & Techniques* McGraw-Hill Osborne Media, ISBN: 0072224738, May 2003.