

Running Voyager in Multi-Server Mode

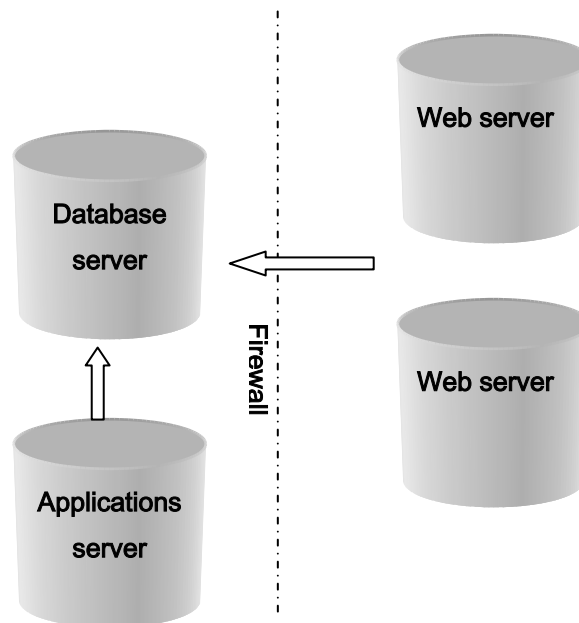
**by Stephen G. A. Thomas
University of Adelaide Library**

26 April 2006

Introduction

The University of Adelaide Library has been running Voyager since 2002. The original implementation used a single server, a Sun E3500 with 3GB ram and five 400Mhz processors. (Full configuration in Appendix D.) With the upgrades to Unicode and Voyager 5.0, this configuration proved to be inadequate, with continuous problems with load causing swap space errors and generally dismal performance.

Late in 2005, we found the funds for a server upgrade. However, since Voyager has a multi-tier architecture (Appendix E), providing load splitting capabilities, as well as the extra security of running the public web server outside of the University firewall (in the DMZ), we elected to go with a split server configuration, with four servers to be deployed as follows:



There will be four servers: 2 x V440 and 2 x V210.

The V440s will be used for the database server and apps server.

The database server (**voyager-db**) will utilise space on the SAN for the /m1 and /oracle file systems, and duplicate all the functions of the existing single server.

The applications server (**voyager-app**) will run the various *srvr Voyager applications supporting the client programs.

The apps server will also be configured to serve as backup for the database server in case of failure, by mounting the SAN disks on the apps server instead

of the database server. The apps server will also have a local copy of /m1 and /oracle, so as to provide a backup service during upgrades and other scheduled downtime of voyager-db.

The V210s will be used for public access to our catalogue – they will act as webserver to Voyager. Public web access will be through the URL <http://library.adelaide.edu.au> which will be redirected by a hardware content switch to one or other web server depending on load.

The public servers will also be used for authentication via cgi scripts, for which we need SSL. These authentication scripts previously ran off the Virtual host <http://auth.library.adelaide.edu.au>, in unsecure mode. These will now be accessed through the secure <https://library.adelaide.edu.au>, which allows us to use the University certificate and avoid the cost of a separate certificate for the library.

Requirements

All servers (other than voyager-db) will need an Oracle client installed. (Not a problem because we have a site licence for Oracle.) This is because we run our authentication scripts from the web servers, and these make use of SQL access to the Voyager database.

All servers will need Perl and a few additional Perl modules: DBI, DBD::Oracle, Net::LDAP and Convert::ASN1 (required by Net).

All servers will need two user accounts: voyager and oracle.

All servers should have /m1 and /oracle. On the database server, /m1 and /oracle will be on the SAN disks. On other servers, these will be local.

The existing server, “voyager”, has rsync server configured, so we can use rsync to suck across /m1 and /oracle to the new servers. A number of other files are also required, or need modification, for which I have created a tar ball:

```
# tar tvf /export/home/voyager/config-files.tar
-rw-r----- 102/103      824 Jan 11 04:59 2005 export/home/voyager/.profile
-rw-r----- 100/101      824 Jan 11 04:58 2005 export/home/oracle/.profile
-rw-r--r-- 100/101      753 Jan 11 04:56 2005 var/opt/oracle/oratab
-rwxr-xr--  0/1        2825 Jul 15 08:33 2003 etc/init.d/voyager
-rwxr-xr-- 100/101    3334 Sep 22 03:24 2005 etc/init.d/dbora
-r--r--r--  0/3        6393 Oct 22 05:19 2005 etc/inet/services
-r--r--r--  0/3        9685 Oct 22 05:20 2005 etc/inet/inetd.conf
-rwxr-xr-x 100/101    2554 Jun  4 05:58 1999 usr/local/bin/oraenv
-rwxr-xr-x 100/101    2428 Jun  4 05:57 1999 usr/local/bin/dbhome
-rw-r--r--  0/1       34129 Nov 29 18:27 2005 usr/local/apache/conf/httpd.conf
```

Some system configuration files need additional parameters. These are described below and detailed in the Appendices.

Voyager-app also requires a number of binaries, in /export/home/oracle/bin

The Gory Details

Configuration – all servers

1. Add required users and groups. All servers should have two users, oracle (uid=100) and voyager (uid=102), in groups dba (101) and endeavor (103) respectively.
2. Copy .profile for voyager and oracle to new accounts.
3. Copy scripts dbhome and oraenv to /usr/local/bin
4. Copy /var/opt/oracle/oratab
5. Edit Apache httpd.conf adding appropriate entries.

Configuration – voyager-db and voyager-app

1. Copy ~oracle/bin to new oracle home dir.
2. Add Endeavor ports to /etc/inet/services (Appendix A)
3. Add Endeavor services to /etc/inet/inetd.conf (Appendix B)
4. Add Oracle memory settings to /etc/system (these may need adjustment since we now have 16GB) (Appendix C)
5. Copy startup scripts voyager and dbora to /etc/init.d, ...
6. Edit the file
/oracle/app/oracle/product/9.2.0/network/admin/listener.ora
and set the hostname to voyager-db

(On voyager-app, this file does not exist; instead make the same change to tnsnames.ora)

Migration

1. Copy file systems /m1 and /oracle

* For web servers and Apps server, we **don't** require:

/m1/voyager/*/data
/m1/voyager/wilsondb
/oracle/oradata

2. In /oracle/app/oracle/product/9.2.0/network/admin/listener.ora, change the host name:

```
(ADDRESS = (PROTOCOL = TCP)(HOST = voyager-db)(PORT = 1521))
```

3. Edit the file /m1/voyager/adelaidedb/etc/webvoyage/voyager.ini replacing the host address with the IP of the database server.

Appendix A : Endeavor's description of the architecture

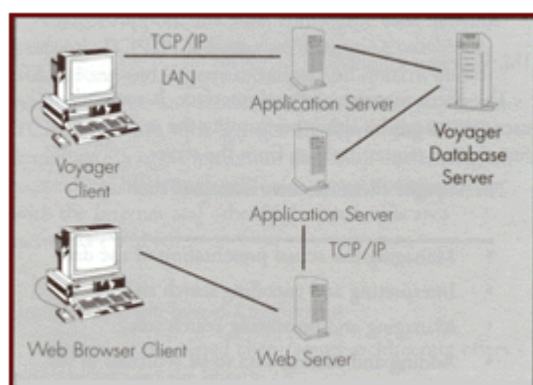
(<http://www.endinfosys.com/tech/arch.htm>)

Multi-Tiered Client/Server Architecture

Voyager is a true client/server application with a clear separation between the client functions, server functions, and database functions. With each component assigned a specific purpose, libraries realize significant improvements in performance and functionality over less-optimal designs or architectures.

Multi-tiered client/server architecture is essential for the needs of forward-thinking libraries because it provides client/server independence allowing:

- Rapid enhancements to the system
- An increase in the number of choices for distributing and growing the system
- Quicker integration of emerging technologies



Clear separation between system objects means a flexible, expandable architecture.

Flexibility is Key in Configuration

Flexibility is provided through Voyager by layering the server side and isolating the client from the database structure through the use of intermediate application layers. It's easy to add new technology quickly since only one layer needs to be changed. The client doesn't need to know the database structure, and the application logic is abstracted from the database. This is truly a great advantage to any library that employs advanced technology as a means to manage information.

Scalability of Voyager Meets the Needs of All Libraries

The multi-tiered approach, while keeping the client/server distinctions between storage and access intact, also allows the server architecture to be divided into application server and database server elements. This is particularly important in scaling the application. The Voyager server is sized by the number of simultaneous users (which determines processor resource requirements) and the size of the collection (which determines the database size and I/O requirements). Because the server is segmented into application and database elements, each part can be scaled independently.

Appendix B : Additions to /etc/inet/services

```
#
# Endeavor Entries Begin
#
olisten          1521/tcp          # Oracle listener port
Pwavirtualhost  7008/tcp          # vhost port
Pwebadmin       7009/tcp          # Webadmin port
Preview         7091/tcp          # Preview server port
Piasock        7500/tcp          # Symtrix listener
#
# Voyager Server entries for production database
#   adalaiddb - Version 2000.1   07/14/2001
#
Popacsvr        7000/tcp          # OPAC Server
Pcatsvr         7010/tcp          # Cataloging Server
Pacqsvr         7020/tcp          # Acquisitions Server
Pcircsvr        7030/tcp          # Circulation Server
Pselfchk        7031/tcp          # Self Check
Prptsvr         7040/tcp          # Reporting Server
Psysadminsvr    7050/tcp          # System Administration Server
Pkeysvr         7060/tcp          # Keyword Server
Pmfhdkeysvr     7061/tcp          # MFHD Keyword Server
Pfilesvr        7070/tcp          # File/Abstracts Server
Pcallslipsvr    7080/tcp          # Request Server
Pscansvr        7081/tcp          # Scandoc Server
Pmediasvr       7085/tcp          # Media Booking
Pz3950svr       7090/tcp          # Z39.50 Server
#
# Voyager Server Entries for Training Database
#   traindb - Version 2000.1     07/14/2001
#
Topacsvr        8000/tcp          # OPAC Server
Tcatsvr         8010/tcp          # Cataloging Server
Tacqsvr         8020/tcp          # Acquisitions Server
Tcircsvr        8030/tcp          # Circulation Server
Tselfchk        8031/tcp          # Self Check
Trptsvr         8040/tcp          # Reporting Server
Tsysadminsvr    8050/tcp          # System Administration Server
Tkeysvr         8060/tcp          # Keyword Server
Tmfhdkeysvr     8061/tcp          # MFHD Keyword Server
Tfilesvr        8070/tcp          # File/Abstracts Server
Tcallslipsvr    8080/tcp          # Request Server
Tscansvr        8081/tcp          # Scandoc Server
Tmediasvr       8085/tcp          # Media Booking
Tz3950svr       8090/tcp          # Z39.50 Server
#
# Voyager Server Entries for Sample Citation Database
#   wilsondb - Version 2000.1
#
Copacsvr        9000/tcp          # OPAC Server
Ccatsvr         9010/tcp          # Cataloging Server
Crptsvr         9040/tcp          # Reporting Server
Csysadminsvr    9050/tcp          # System Administration Server
Ckeysvr         9060/tcp          # Keyword Server
Cfilesvr        9070/tcp          # File/Abstracts Server
Cz3950svr       9090/tcp          # Z39.50 Server
#
# End of Endeavor System Entries
```


AppendixC : Additions to /etc/inet/inetd.conf

```
#
# Endeavor System Entries Here
#
#
#           Voyager Entries for Production Database
#           adelaidedb - Version 2000.1
#
Popacsvr      stream  tcp      nowait  voyager  /ml/voyager/adelaidedb/sbin/Popacsvr
Pcatsvr       stream  tcp      nowait  voyager  /ml/voyager/adelaidedb/sbin/Pcatsvr
Pacqsvr       stream  tcp      nowait  voyager  /ml/voyager/adelaidedb/sbin/Pacqsvr
Pcircsvr      stream  tcp      nowait  voyager  /ml/voyager/adelaidedb/sbin/Pcircsvr
Pselfchk      stream  tcp      nowait  voyager  /ml/voyager/adelaidedb/sbin/Pselfchk
Prptsvr       stream  tcp      nowait  voyager  /ml/voyager/adelaidedb/sbin/Prptsvr
Psysadminsvr  stream  tcp      nowait  voyager  /ml/voyager/adelaidedb/sbin/Psysadminsvr
/ml/voyager/adelaidedb/sbin/Psysadminsvr
Pkeysvr       stream  tcp      nowait  voyager  /ml/voyager/adelaidedb/sbin/Pkeysvr
Pmfhdkeysvr   stream  tcp      nowait  voyager  /ml/voyager/adelaidedb/sbin/Pmfhdkeysvr
/ml/voyager/adelaidedb/sbin/Pmfhdkeysvr
Pfilesvr      stream  tcp      nowait  voyager  /ml/voyager/adelaidedb/sbin/Pfilesvr
Pcallslipsvr  stream  tcp      nowait  voyager  /ml/voyager/adelaidedb/sbin/Pcallslipsvr
/ml/voyager/adelaidedb/sbin/Pcallslipsvr
# Pscansvr     stream  tcp      nowait  voyager  /ml/voyager/imagedb/sbin/Pscansvr
# Pmediasvr    stream  tcp      nowait  voyager  /ml/voyager/adelaidedb/sbin/Pmediasvr
/ml/voyager/adelaidedb/sbin/Pmediasvr
#
#           Voyager Entries for Training Database
#           traindb - Version 2000.1
#
Topacsvr      stream  tcp      nowait  voyager  /ml/voyager/traindb/sbin/Topacsvr
Tcatsvr       stream  tcp      nowait  voyager  /ml/voyager/traindb/sbin/Tcatsvr
Tacqsvr       stream  tcp      nowait  voyager  /ml/voyager/traindb/sbin/Tacqsvr
Tcircsvr      stream  tcp      nowait  voyager  /ml/voyager/traindb/sbin/Tcircsvr
Trptsvr       stream  tcp      nowait  voyager  /ml/voyager/traindb/sbin/Trptsvr
Tsysadminsvr  stream  tcp      nowait  voyager  /ml/voyager/traindb/sbin/Tsysadminsvr
Tkeysvr       stream  tcp      nowait  voyager  /ml/voyager/traindb/sbin/Tkeysvr
Tmfhdkeysvr   stream  tcp      nowait  voyager  /ml/voyager/traindb/sbin/Tmfhdkeysvr
Tfilesvr      stream  tcp      nowait  voyager  /ml/voyager/traindb/sbin/Tfilesvr
Tcallslipsvr  stream  tcp      nowait  voyager  /ml/voyager/traindb/sbin/Tcallslipsvr
# Tscansvr     stream  tcp      nowait  voyager  /ml/voyager/imagedb/sbin/Pscansvr
# Tmediasvr    stream  tcp      nowait  voyager  /ml/voyager/traindb/sbin/Tmediasvr
#
#           Voyager Entries for Sample Citation Database
#           wilsondb - Version 2000.1
#
Copacsvr      stream  tcp      nowait  voyager  /ml/voyager/wilsondb/sbin/Copacsvr
Ccatsvr       stream  tcp      nowait  voyager  /ml/voyager/wilsondb/sbin/Ccatsvr
Crptsvr       stream  tcp      nowait  voyager  /ml/voyager/wilsondb/sbin/Crptsvr
Csysadminsvr  stream  tcp      nowait  voyager  /ml/voyager/wilsondb/sbin/Csysadminsvr
Ckeysvr       stream  tcp      nowait  voyager  /ml/voyager/wilsondb/sbin/Ckeysvr
Cfilesvr      stream  tcp      nowait  voyager  /ml/voyager/wilsondb/sbin/Cfilesvr
#
# End Endeavor System Entries
```

Appendix D : Additions to /etc/system

```
*
* Endeavor Information System Entries
*
*   Suitable for
*       60 OPAC licenses
*       150 Staff licenses
*           1 Training databases
*           1 Production databases
*
* set semsys:seminfo_semmap=40
set semsys:seminfo_semmni=150
set semsys:seminfo_semmns=2850
set semsys:seminfo_semmnu=2850
set semsys:seminfo_semmsl=2720
set shmsys:shminfo_shmmax=2047000000
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=20
* set semsys:seminfo_semume=10
set semsys:seminfo_semopm=100
* set semsys:seminfo_semusz=96
* set semsys:seminfo_sevmx=32767
* set semsys:seminfo_semaem=16384
* set shmsys:shminfo_shmseg=10

forceload:      sys/semsys
forceload:      sys/shmsys

set slowscan=500

set noexec_user_stack=1
set noexec_user_stack_log=1
set nfssrv:nfs_portmon=1
set ufs:ufs_HW=1048576
set ufs:ufs_LW=917504
set maxphys=1048576
* Following added by Nap @ Endeavor 18 Mar 2003
set tcp:tcp_conn_hash_size=8192
*
* Endeavor Information Systems Inc.
*   End of entries
```

Appendix E : Oracle configuration

On voyager-db, we need the file

/oracle/app/oracle/product/9.2.0/network/admin/listener.ora
to be configured thus:

```
# LISTENER.ORA Network Configuration File:
/oracle/app/oracle/product/9.2.0/network/admin/listener.ora
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = voyager-db)(PORT = 1521))
      )
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = VGER)
      (ORACLE_HOME = /oracle/app/oracle/product/9.2.0)
      (SID_NAME = VGER)
    )
  )
```

And on the other servers we need tnsnames.ora to be configured thus:

```
# TNSNAMES.ORA Network Configuration File:
/oracle/app/oracle/product/9.2.0/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

VGER =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = voyager-db)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = VGER)
    )
  )
```

Appendix F : Existing server configuration

```
> /usr/platform/sun4u/sbin/prtdiag -v
```

```
System Configuration: Sun Microsystems sun4u 5-slot Sun Enterprise E3500
```

```
System clock frequency: 100 MHz
```

```
Memory size: 3072Mb
```

```
===== CPUs =====
```

Brd	CPU	Module	Run MHz	Ecache MB	CPU Impl.	CPU Mask
5	10	0	400	8.0	US-II	10.0
7	14	0	400	8.0	US-II	10.0
7	15	1	400	8.0	US-II	10.0
9	18	0	400	8.0	US-II	10.0
9	19	1	400	8.0	US-II	10.0

```
===== Memory =====
```

Brd	Bank	MB	Status	Condition	Speed	Intrlv. Factor	Intrlv. With
5	0	1024	Active	OK	60ns	2-way	A
7	0	1024	Active	OK	60ns	2-way	A
9	0	1024	Active	OK	60ns	1-way	

```
===== IO Cards =====
```

Brd	Bus Type	Freq MHz	Slot	Name	Model
1	SBus	25	3	SUNW,hme	
1	SBus	25	3	SUNW,fas/sd (block)	
1	SBus	25	13	SUNW,socal/sf (scsi-3)	501-3060