

Microsoft®

Health Users Group

TECH FORUM 2008

GOLDEN TICKET

August 26-27, 2008 | Microsoft Corporate Campus | Redmond, WA

HimSS

www.mshug.org



Application of High-Availability and Scalability Techniques to Non High-Availability Clinical Applications

David H. Miller
Senior Key Expert
Siemens Healthcare



Agenda

- Definition of High Availability
- Definition of Scalability
- Understanding the Application
- Re-architecting
- Testing/Documenting
- Implementation
- Monitoring
- Alerting
- Summary/Q&A



High-Availability

- ITIL Definition - An approach or Design that minimizes or hides the effects of Configuration Item Failure on the Users of an IT Service. High Availability solutions are Designed to achieve an agreed level of Availability and make use of techniques such as Fault Tolerance, Resilience and Fast Recovery to reduce the number of Incidents, and the Impact of Incidents.
 - Source: ITIL® V3 Glossary v01 (2007,30 May 2007). Retrieved June 23, 2008 from http://www.best-management-practice.com/gempdf/ITILV3_Glossary_English_v1_2007.pdf
- **Clinical applications – 24 x 7 x 365 agreed level of availability**
- **Problem – many applications used in clinical settings not designed this way**



Scalability

- ITIL - The ability of an IT Service, Process, Configuration Item etc. to perform its agreed Function when the Workload or Scope changes.
 - Source: ITIL® V3 Glossary v01 (2007,30 May 2007). Retrieved June 23, 2008 from http://www.best-management-practice.com/gempdf/ITILV3_Glossary_English_v1_2007.pdf



Understanding the Application

- Key to architecting for high availability
- Read documentation
 - especially install documents
- Meet with:
 - Users
 - Developers if possible
 - Support personnel



Examples

- Example 1 - Single Server Application with:
 - IIS 6.0 web-based application
 - SQL Server 2000®
 - Message Queuing – proprietary
 - Files to be exchanged
 - Servers Post data to website
 - Workstations review data from website
- Example 2 – Single Server Application with:
 - Windows® application
 - SQL Server 2000
 - Servers send data to application
 - Workstations review data from application

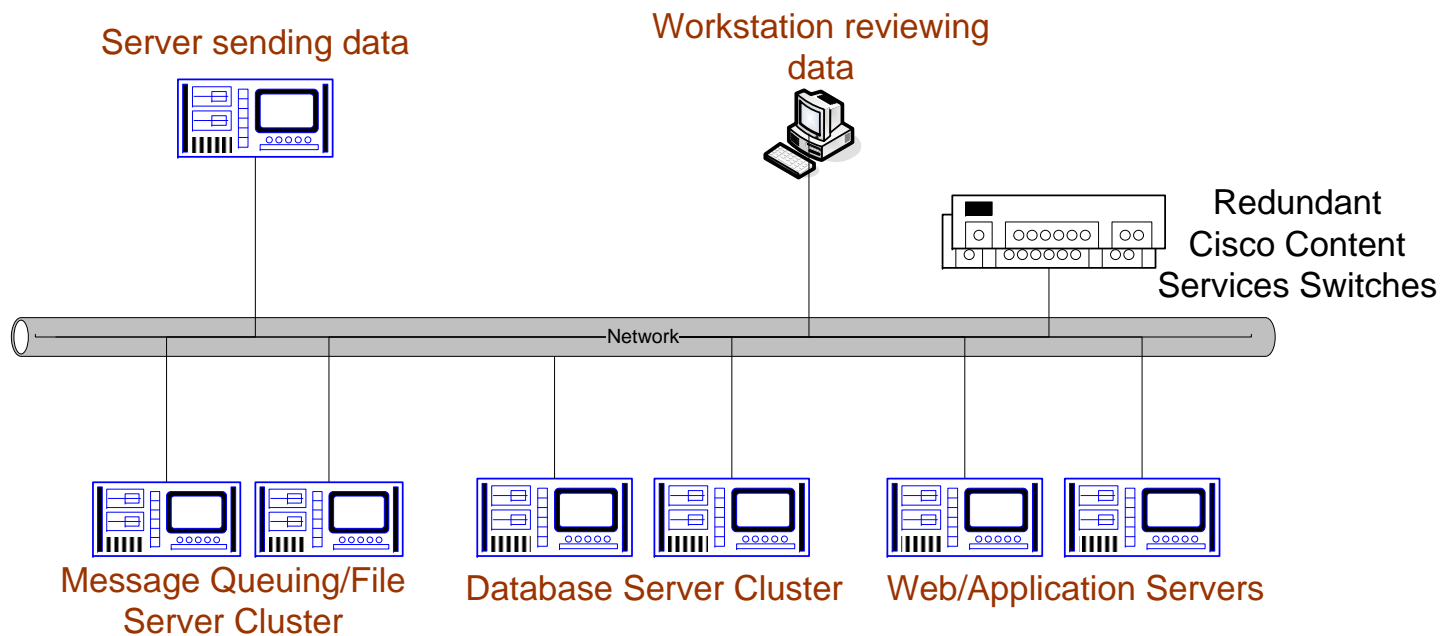


Re-architecting for High Availability

- Separate portions of the application into like tiers
- Use high-availability options for that tier
 - Use content engine to ‘spray’ requests to web/application tier
 - Use Microsoft® Clustering for database tier
 - Use Microsoft Clustering for application tier



Example 1 Re-architected



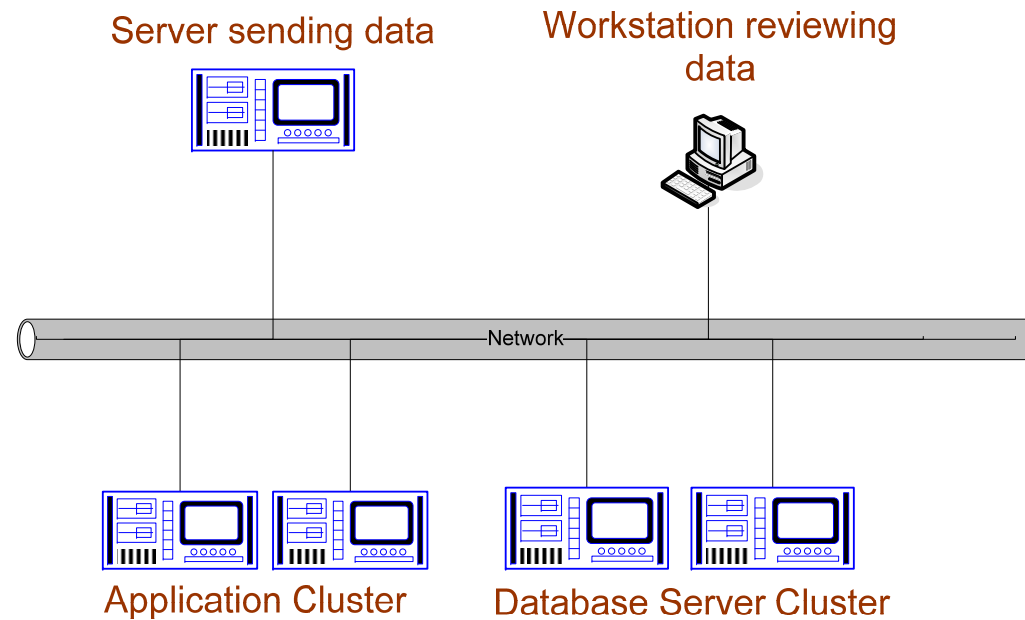


Example 1 Re-architected

- Multiple Web/Application servers
 - Cisco® Content Services Switch sends http requests from servers or workstations to one or other
 - If one server down, then all requests are handled by other
- Clustered Database Servers
 - If one server goes down, databases fail over to other
- Clustered Message Queuing/File Servers
 - If one server goes down, message queues and file servers fail over to other



Example 2 Re-architected





Example 2 Re-architected

- Clustered Application servers
 - If one server goes down, message queues and file servers fail over to other
- Clustered Database Servers
 - If one server goes down, databases fail over to other



Testing/Documentation

- Test proposed configuration in lab
- Update install documentation
- Example 1
 - Proposed configuration worked
 - Install documentation rearranged by tier
 - Run install multiple times
- Example 2
 - Proposed configuration did not work
 - Application start-up errors during cluster failover



Example 2 - Back to Developers

- Application startup tied to two specifics:
 - Entry in one table in database with server name
 - File on cluster disk tied to server name
- Since server name different on other cluster node, startup fails
- Solution
 - Script written to:
 - Update database table with correct server name
 - Update file with server name
 - Script made a cluster script
 - Cluster application startup is dependent upon script



Re-architecting for Scalability

- Separation of tiers allows for easier scalability
- Web/Application Tier
 - Add additional servers to 'spray'
- Clustered Database or Application Tier
 - Multiple SQL Instances
 - Multiple cluster groups
 - Use n+1 clusters
 - Add more powerful servers using clustering



Example 2 – Scalability

- Multiple Instances of application did not work
- Startup errors showed conflicts with ports
- Startup option allowed moving ports to different ranges for each instance
- Tip for tying ports to programs
 - “netstat –no” to list port numbers/PIDs
 - Task manager to show PIDs to executables



Implementation

- Use documentation that was developed to implement high-availability application in production



Monitoring/Alerting

- Assumes
 - 24x7 staffing
 - Monitoring software
- Document what support staff can do without requiring oncall server administrators
- Monitor OS and Cluster Basics
 - Ping, Services, Cluster failures
- Application Monitoring
 - SNMP Alerts
 - Log files
- Perform Root Cause Analysis of any Outages
 - Determine alert strategy for future occurrences



Additional Benefits to High-Availability Architecture

- Allows Planned Outages for:
 - Security patch applies
 - Regular reboots
 - Certain application upgrades
- Hard to achieve 100% Availability
 - OS upgrades (2003 to 2008)
 - SQL Server upgrades (2005 to 2008)
 - Major application upgrades



Summary/Q&A

- Re-architecting healthcare applications can provide much higher availability and scalability
- Designing it in is best option



Acknowledgements

- ITIL ® is a Registered Trademark, and a Registered Community Trade Mark of the Office of Government Commerce, and is Registered in the U.S. Patent and Trademark Office
- PRINCE ® is a Registered Trademark and a Registered Community Trademark of the Office of Government Commerce, and is Registered in the U.S. Patent and Trademark Office
- M_o_R ® is a Registered Trademark and a Registered Community Trademark of the Office of Government Commerce
- Microsoft, Windows, and SQL Server are registered trademarks of Microsoft Corporation