

Assuring High Availability in Healthcare Interfacing Considerations and Approach

High availability is a term used in the software industry to indicate that the application is available a high percentage of the time during its expected processing hours. Availability is measured in percentage of up time for the system. For example, if a system has a 24 hour expected processing time and is down 17.5 hours per year, it has a 99.8% up time or availability.

Most server class software attempts to be extremely reliable, but there are unexpected issues that cause downtime. These issues are primarily caused by:

- Network problems, such as switch or router failures
- Operating system problems, such as viruses or blue screens
- Hardware failures, such as network cards and hard drives
- Application failures, such as the inability to process any messages

In any of the above scenarios, patient clinical transactions stop moving, and patient data is no longer available for the physicians, nurses, and other healthcare professionals.

Although software creators can attempt to resolve the possible application failures, they are unable to resolve the network, operating system, and hardware failures. Some type of backup system needs to be employed that can take over and keep processing when there is a failure.

How quickly the backup system is able to take over and resume processing directly affects the availability of the software. More importantly, it directly impacts how quickly a caregiver receives the patient information that they require to perform their critical responsibilities.

Determining the Need for High Availability

Just like having regular backups of systems, virus protection, and disaster recovery plans, high availability is important. There are costs to implement a high availability solution, but it will likely be less than being in a fire-drill to correct a critical system failure. The high availability upfront costs typically include additional hardware, software, and additional configuration time.

It becomes a business decision of balancing risk with costs – in other words, balancing patient care delivery with costs of supporting it.

In patient care, a healthcare organization needs to consider that HL7 or XML messages being exchanged contain a patient's clinical data and, in most cases, the need to have the data available at the right time is great. First and foremost, it is an issue of quality of patient care, but there also are regulatory requirements, due in part to HIPAA.

On the cost side, how much is downtime?

If the system is responsible for maintaining 100 connections that send and receive 1,000 messages an hour between various healthcare applications, the costs will be significant to manually perform the data recovery tasks. Each message would need to be recovered and then re-sent in a labor-intensive manner. In this example, it may be well worth the cost to assure the system was available.

If, however, the system is responsible for maintaining two connections that send 10 messages an hour between several healthcare applications, the costs to manually recover would be significantly lower than the previous example. Nonetheless, it would be a manual effort to recover and re-send even these 10 messages but still may not be worth the extra cost to ensure high availability.

Providing High Availability for Critical Systems

Assuring high availability requires a holistic focus. It cannot be fully addressed by only purchasing better servers. High availability includes many parts, including hardware, database, and software.

Hardware

One main aspect of failure prevention is reasonable hardware. It is of little use to put highly available software on hardware that is unreliable. Reasonable hardware needs to be used for critical systems.

Reasonable hardware begins with having enterprise class servers for the critical applications. Additional features are required to protect the servers, such as redundant power supplies, backup fans, and added tolerance for brown outs. Moreover, storage needs to be highly available with approaches such as a Redundant Array of Independent Disks (RAID) configuration. Each piece of the hardware configuration for the critical systems should be evaluated for reliability.

Database

If there is a database associated with the application, refer to the database's documentation for current best practices on providing high availability. Typically, this involves a secondary server.

Software

Nearly every server class software system, including healthcare interface engines, strives to be 100% reliable. Given the right conditions, however, the application may fail.

There are at least three approaches a software application can take to provide high availability. Each has its advantages and disadvantages, and the importance of these will differ depending upon the design of the original software application.

- **Operating System:** Windows Clustering is an example of high availability provided by an operating system. In this approach, requests are shared across multiple servers. This option works best for transaction-oriented systems and is frequently used for e-mail servers and databases. An HL7 interface engine, although transaction oriented, requires a persisted connection to various external applications or systems. Maintaining connection to the external applications or systems between multiple servers in this environment is challenging.
- **Third-Party Products:** There are tools available to attach to the side of an application that will manage the high availability. It may be a “slightly” easier approach than the operating system approach. The biggest challenge, however, is the ability of the software on the primary server to know that there was a failure, and it is no longer the active server. When the primary server comes back online, it is difficult to failback gracefully, because there is no knowledge of its failure. In healthcare messaging, it is important to only process each message once and in the correct order. Third party tools usually cannot ensure these imperatives.
- **Native:** The application itself contains the high availability functionality. This ensures that the application has direct knowledge of the failover process. The high availability configuration may be easier since much of it is handled internally within the application – user knows the existing application. Corepoint Integration Engine has selected the native approach because it allows for the easiest implementation of hot standby and synchronous backup.

Corepoint Health Approach: How Does It Work?

Corepoint Health has implemented the native approach because:

- It allows for the easiest implementation of hot standby and synchronous backup.
- It is cost-effective for budget-conscious healthcare organizations.
- It does not require additional, new IT skills or personnel.

- It ensures that each message is processed only once and in the correct order – every time, in real time.

Although there are various definitions of high availability, with Corepoint Integration Engine Assured Availability™ (A2), the backup server immediately takes over and keeps processing, providing a reasonable level of continuous operation given a downtime event.

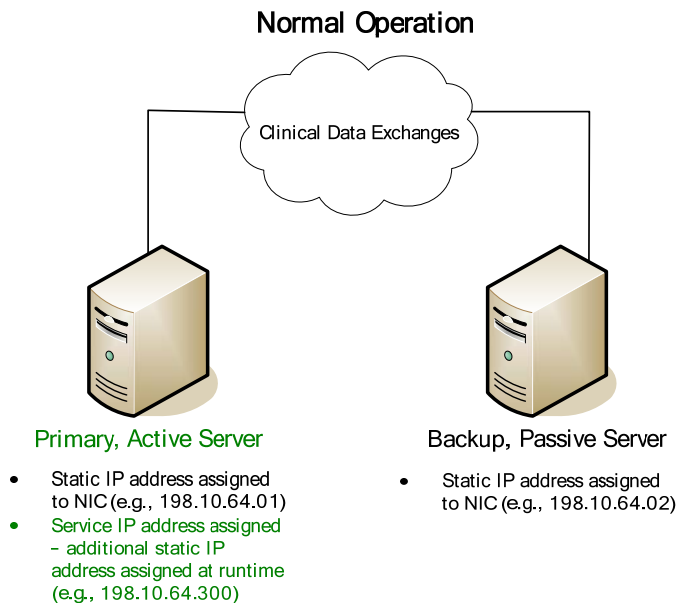
Since high availability is built natively into Corepoint Integration Engine,

- There is intimate knowledge of the health of the primary and backup server, both in active and passive states, allowing for synchronous replication of all the messages and configuration information providing for graceful failover and failback.
- The synchronization of all configuration files and connection profiles is handled by Corepoint Integration Engine making configuration of high availability less labor intensive and almost bulletproof

A2 in Action

In order to enable A2, a primary and a backup server running on Windows 2003 are required, each with a static IP address. The backup server should be similar hardware configuration as the primary server and only used for failover.

An additional static IP address (Service IP address) is required to be assigned at runtime, based upon the health of the server. The Service IP address is the address that external systems will use to connect to Corepoint Integration Engine. The Service IP is assigned to the server that is acting as the active server. During normal processing, this is the primary server. The Service IP address allows for a continuous connection point for the external systems that are connecting to Corepoint Integration Engine.



With the A2 solution, the messages being communicated are written to the data persistency queues on the primary and backup servers. Once the message is enqueued on both servers, the acknowledgement is sent. The primary server processes the message, updating the backup server when the message is complete.

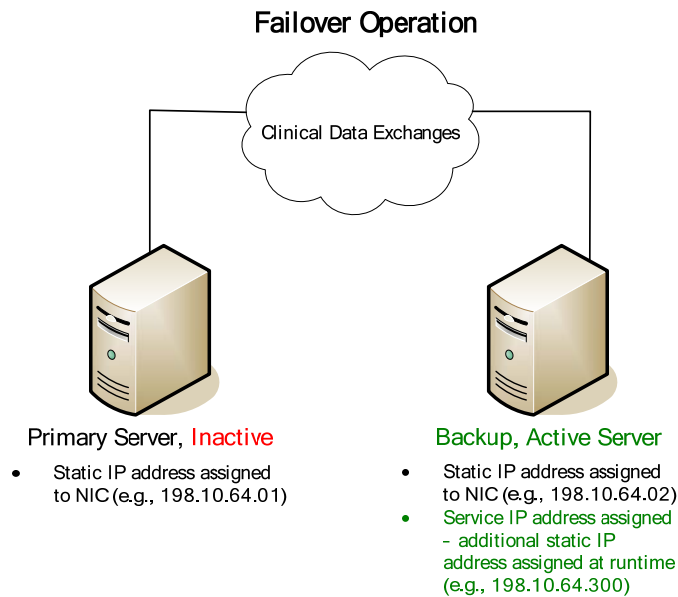
In normal operation, the primary server is the active server and the backup server is the passive server, available for failover. During processing, the primary server copies over configuration and connection changes as soon as changes are saved. Consequently, the backup server has the configuration and connection information needed to pick up processing, if necessary. The backup server is also constantly monitoring the health of the primary server, checking its status to determine if action needs to be taken.

Primary Server Failure (Failover)

If the backup server is unable to contact the primary server through normal means, it attempts to PING the primary server. If it does not get a response from a PING, it attempts to contact the Reference IP address(es) to ensure that the problem is not with the backup server.

If it is successful in contacting the Reference IPs, it assumes that the primary server has failed and initiates a failover alert. Automatically, the backup server takes over the service IP address, processes all incomplete messages, and accepts and communicates new messages. It will continue acting as the active server until the manual process of failback takes place.

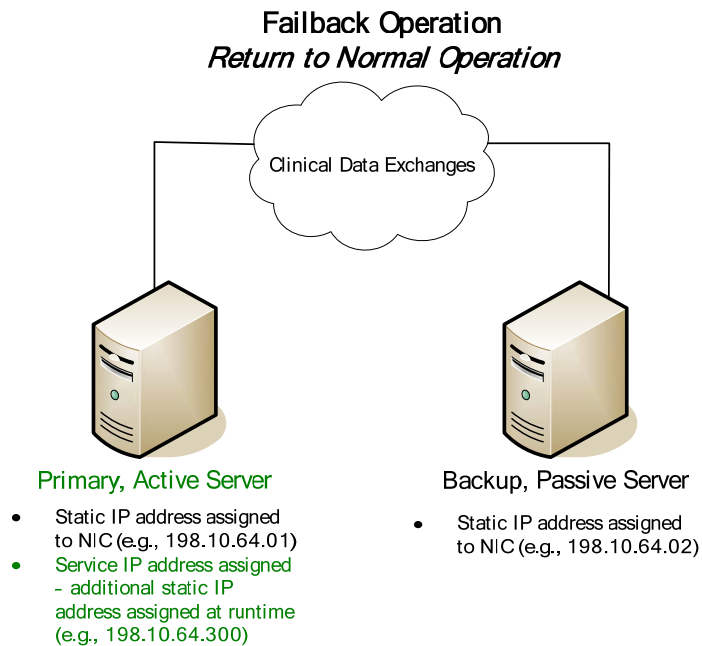
Clinical data exchanges continue with no degradation in service. Healthcare operations and patient care delivery continue seamlessly.



Failback (Return to Normal Operation)

Upon startup, the primary server contacts the backup server to confirm the current running state. If the backup server is active, the primary server knows that it is not in control and waits until a manual failback to start processing messages. The manual process for failback is to prevent confusion while the primary server is being restored.

When the manual failback is initiated, all message data is synchronized. Once complete, the primary server picks up the Service IP address, starts the Corepoint Integration Engine service, and automatically starts connections and begins processing. The backup returns to the passive state.



Summary

A2 effectively delivers high availability for any clinical data exchanges that are queued within Corepoint Integration Engine. It is an effective solution from multiple points of view: cost, productivity, robustness, manageability, etc.

Healthcare is a unique environment. The delivery of patient care cannot stop and wait for required information. Efficiency of healthcare operations cannot wait for the right information to show up sometime.

Corepoint Health offers a balanced approach to ensuring that the patient data is delivered in a timely and continuous manner.

About Corepoint Health

Corepoint Health solutions deliver interoperability for healthcare organizations and simplify the complexities of healthcare data through practical software applications, consulting and training. Our innovative and proven software solutions leverage clinical data flow efficiently for a diverse group of healthcare entities including hospitals, imaging centers, laboratories, clinics and healthcare vendors. This next generation approach to healthcare data and streamlined workflow is where Corepoint Health specializes in helping customers discover the power of integration.

www.corepointhealth.com

Corepoint Health
6509 Windcrest Drive
Suite 160B
Plano, Texas 75024
469-229-5000
info@corepointhealth.com