In a perfect world, all systems would have common configuration, monitoring, recovery, security, and performance-tuning management features. Most often, however, systems designers must carefully craft system architectures that integrate them. IBM® not only provides solution services but also has created a wide variety of tools and techniques for integration. Starting an enterprise architecture by considering the UIs, IT administration, and how the system can operate with autonomic features so that it is mostly self-maintaining is perhaps the best place to focus early on in system design. After all, a system that is difficult to use or maintain and extend won't be successful regardless of performance or other features. Finally, green systems management is a serious concern as data centers, cloud computing, and supercomputing are pushing scales to exascale: The ability to manage these huge-scale resources with efficiency in order to get the highest performance and utility with the least power is critical. This article arms the systems designer and solution architect with methods for success in the design of management for resources and the data and services they host in enterprise-scale systems.

Frequently used acronyms

- API: Application programming interface
- GUI: Graphical user interface
- IP: Internet Protocol
- LAN: Local area network
- NAS: Network attached storage
- RAID: Redundant array of independent disks
- SAN: Storage area network
- UI: User interface
- WAN: Wide area network

This article, the fourth installment in this series on infrastructure architecture, provides an overview of critical systems-management concepts, tools, and techniques for enterprise-scale resource management and
management of services and data. Management is most often thought of as the ability to maintain resource availability and the availability of data and services those resources host. Traditionally, management of enterprise systems is provided by an IT group within an organization that ensures that resources, data, and services are available and secure as well as properly configured, fully functional, and performing well. The goal of automating IT to the greatest extent possible to reduce the IT workload and to enable scaling of data and services has lead to an area of management research and development called autonomic computing.

The idea of autonomic computing is to design systems and subsystems to be as self-configuring, healing, optimizing, and protecting (self-CHOP) as possible. Whether data and services are made available through autonomic or traditional IT methods, a fundamental system design that enables management of redundant resources, replicated data, and scalable services that can also be migrated is required. As such, this article focuses on those base management capabilities that would support traditional and automated management approaches. System designs for enterprises are frequently terascale today and reaching petascale levels: High-performance computing (HPC) has an eye on exascale computing as the next grand challenge.

The third article in this series, "System design methods for scaling," focused on scaling. Clearly, the goals for scaling enterprise systems from terascale to petascale and exascale will require carefully designed management infrastructure and must leverage fundamental research on management automation coming from autonomic computing goals.

Configuration of systems

An ideal configuration interface for an enterprise system should have:

- **Single pane of glass:** This one-page interface provides high-level configuration information with the ability to drill down into lower levels of detail.
- **Authoritative configuration database and status:** Out-of-date or inaccurate system-configuration information is a leading cause of downtime, so management systems should carefully store and automatically compare current and stored configuration data.
- **Secure remote access:** Because most enterprise systems will be distributed by nature or because of disaster-proofing requirements and may interface with the Internet, security that controls access but does not prevent it for authorized users is essential.
- **Configuration change logs and audit capability:** As noted in the *IBM System Journal* article "An overview of IBM Service Management" (see Resources), up to 80 percent of all critical system outages can be traced to faulty change management.
- **Resource resizing for scaling up or down:** Thinking ahead about the need to scale up or down minimizes service outages resulting from poorly managed change.
- **Notification of configuration changes:** Focus is often on notification of system errors rather than configuration events, but monitoring changes is essential no matter how they come about.

The autonomic computing goal

The relentless pursuit of Moore's law and success in this pursuit of computing resources has led us from gigascale to terascale to petascale computing—and promises exascale computing within decades. Ultimately, computing has been developed to bring data and services to users and to automate all aspects of human life, so simply increasing the scale of computing resources does no good if they are not properly configured, optimized, available or recoverable, and secure. Today, there are exabytes of data in the world as a whole, with numerous terabyte and many petabyte systems. Likewise, supercomputing has achieved many teragrid-linked petascale computing resources. With traditional methods, these tremendous resources may not be able to be used efficiently, yet they hold promise to assist humankind with grand challenges such as energy, global warming, and disease prevention. Clearly, the ability to manage and use these resources at exascale levels to...
provide useful services is one of the biggest challenges facing computer architects today.

Skills and competencies: Configuration

The skills and competencies required for enterprise configuration management vary widely. However, such requirements typically include:

- **SAN**: Block-level access to storage resources over a Fibre Channel, Internet small computer system interface (iSCSI), or SCSI over Infiniband.
- **NAS**: File-level access to data through a NAS head or NAS gateway to a cluster of file servers from a client network.
- **Server**: A host for services and applications that most often also provides an interface between enterprise IP networks and storage.
- **Client networks**: Most often an IP network that may include redundant links and diverse routing for client access to enterprise services and data.
- **Cluster networks**: A high-speed local network of servers used to scale services and data access in an enterprise node or location.
- **Management networks**: Often, a secure, lower-speed WAN that enables enterprise IT teams to monitor, configure, and in general manage all IT assets.
- **Security**: Network encryption, user authentication, resource access control, data access control and data-at-rest encryption, and protection of resources from hackers (for example, denial of service [DoS] attacks, viruses, worms, phishing).
- **Storage virtualization**: The ability to abstract physical solid-state, disk, and tape storage devices into volumes that can be scaled up or down and mapped using RAID, and that allow devices to be interchanged without the need to rewrite applications.
- **Virtual machines (VMs)**: Scaling of compute nodes so that multiple operating system instances and application platforms can share one server.
- **Backup**: Either with remote mirrors, instant snapshots of volumes, or traditional information life cycle management, where data is migrated from high-access (solid-state or disk) to low-access storage (tape).
- **Disaster recovery**: Mirrors of services, data, and resources in geographically remote locations to prevent service or data loss during earthquakes, fires, floods, or an attack.
- **Logging**: Each asset, such as a server, should have logs, a storage controller, a client, and so on. The ability to coordinate and correlate logs should exist at the enterprise level.

Tools and techniques: Configuration

A wide variety of tools can be used to integrate component and subsystem configuration capabilities. In enterprise systems, each component or subsystem most likely includes several APIs and command-line interfaces (CLIs) as well as GUIs for configuring that component or subsystem. The challenge is integrating the management capabilities of each element into a consistent and cohesive enterprise management system. IBM offers several solutions, including service management and Tivoli® products (see Resources). Likewise, the open source community offers a wide variety of methods for integrating management tools, including Web services such as SOAP and Web Services Description Language (WSDL), so that existing APIs can be integrated into a common Web-based configuration and management interface (see Resources).

The best first step is to carefully plan the SAN, NAS, client network, management network, storage subsystems, servers, and an overall redundancy scheme, as shown in Figure 1, with careful study of management-integration options for each subsystem or component in an overall highly available and well-managed enterprise system.

**Figure 1. Example of high-availability (HA) SAN configuration and management**
System availability and self-healing

Enterprise data, services, and resources can suffer a wide range of failures. For data, possible failures include data loss, data corruption (silent or known), access problems (performance or loss of connectivity to storage), and data security. For services, failures include coding errors or bugs, resource exhaustion, vulnerability to security attacks, lack of access, and misconfiguration. Likewise, all resources in the system can suffer from hardware failures of numerous types, including disk failure, network link and power failure, data center site disasters, or simple reconfiguration mistakes.

The concept of self-healing invokes ideas of human healing, with reproduction of cells from raw material. This type of healing is still certainly way beyond today's technology (a focus of research in nanotechnology, perhaps). So the self-healing that can be incorporated in systems really requires resource, data, and service redundancy with redundancy management. The diagram in Figure 1 provides an example of IBM DS4800 storage subsystems and IBM System x™ 3850 servers that interface to each storage subsystem with Fibre Channel but also with remote mirroring of volumes over an iSCSI WAN. In the Figure 1 example, primary volumes (P1 through P8) are backed up by mirror volumes (M1 through M8) so that if any one server or storage node goes down, data and services can be accessed through the remaining seven. This arrangement is an N+1 redundancy-management scheme for data, services, and compute resources. An N+1 scheme would allow the IT team to perform planned upgrades that require downtime, and it also handles unplanned downtime. Because only one system can be down at a time without service interruption, this is a single-fault-tolerant design.

Skills and competencies: Designing for high availability

Designs for high availability can vary significantly depending on whether the goal is five 9s (expected outages amounting to less than five minutes per year)—where the system is unavailable in terms of data, resources, or service access—or whether it has no single point of failure (SPOF). The configuration in Figure 1 has redundant servers, data volumes, and storage subsystems and could be configured for redundant links (using multi-path management software for the SAN and for the client-management network)—with power management and redundant power sources—and therefore could be built out for no SPOF.

Clearly, such a configuration is costly, but it's the best approach for enterprise systems that simply can't tolerate downtime for maintenance or because of faults. Instead of full replication of subsystems, many
enterprise systems may have a few SPOFs that are five 9s and highly available because of internal redundancy management. The IBM System Storage DS® 8000 subsystem can not only be configured in no-SPOF high-availability designs but also includes significant internal redundancy, as shown in Figure 3, where all drives are dual-ported and the storage subsystem itself includes dual controllers.

This type of internal redundancy may obviate the need for duplicate subsystems. However, in today's world, with both natural and terrorist hazards, it still may make sense to fully replicate services, compute and storage resources, and data volumes for disaster recovery. Figure 2 shows an extreme example of N+2 disaster recovery, where three locations all back each other up in terms of data and data access for a SAN client. In Figure 2, two of the three sites could go down, and the SAN client would still have access to volumes 1, 2, and 3 through some combination of \{P1, M2, M3\}, \{P2, M1, M3\}, and \{P3, M1, M2\}. This level of redundancy would be very costly, because data is triple-mirrored but also very safe.

**Figure 2. Example of N+2 volume mirroring for disaster recovery**

![Figure 2](image)

Tools and techniques: Linux high availability

Management of high-availability systems designs like those depicted in Figure 1 and Figure 2 requires either constant IT monitoring so that access to services and data can be failed-over or have automated failover when systems are down. Most often, automated failover is done through SAN or NAS multi-path management that is tied in to server and storage controller monitoring. The Linux® high-availability framework (see Resources) provides an excellent approach to configuring Linux-based systems for automated failover using Linux infrastructure such as IP network interface bonding and SAN device mapper multi-pathing.

The internal redundancy of the DS8000, combined with remote mirroring for disaster recovery, provides unparalleled availability. Where and how much to invest in these features should be driven by service availability and data-protection needs. Fred Moore's Horison site (see Resources) provides some excellent insight into causes of downtime and data loss as well as associated costs. The ability to manage redundancy should also be carefully considered. One advantage of internal redundancy, such as that shown in Figure 3 for the DS8000, is the simplicity of built-in management.

**Figure 3 (from the IBM Redbook DS8000 Series: Architecture and Implementation). Example of internal redundancy management**
System monitoring and optimization

Ideally, systems that have self-healing features should be monitored by exception. Then IT personnel are not tied to GUIs or CLIs. Rather, they receive notifications through e-mail, mobile phone text messages, or some other asynchronous method so that they're free to manage more important things like the configuration, planning for scaling, and focus on services rather than monitoring. The Simple Network Management Protocol (SNMP) offers an excellent choice for remote or automated monitoring of WANs, LANs, and SANs, as well as servers and storage subsystems (see Resources). Many subsystems have been designed for SNMP and have management information bases required to describe resource and performance monitors.

System protection and security

A full discussion of enterprise security goes beyond the scope of this article. However, you'll find the "Big iron lessons" series on security (see Resources) a great start. Some new trends in security to keep an eye on include full-disk encryption as well as ever-evolving data in-flight encryption. Many of the goals of well-managed enterprise systems such as disaster recovery potentially open new security holes because of networking and geographically distributed resources, so security planning should be considered during all phases of design and management.

Green systems management

Green systems have become increasingly important because of evidence of global climate change, the high cost of energy, and predicted shortages of fossil fuels. Furthermore, renewable energy comes at a much higher cost per kilowatt compared to nonrenewable energy. As such, a great deal of focus has been put into efficiency and total cost of ownership for enterprise systems. Some of the most important considerations are:

- **Power consumption**: Choosing green subsystems helps in the overall cost immensely, so carefully consider input and output per watt, gigabytes per watt, million instructions per second per watt, and other key performance and power parameters.
- **Heating and cooling**: Power generates heat and requires data center cooling, which uses yet more power. Advanced liquid-cooling methods are available, and many green subsystems run much cooler as well as using less power.
- **Legacy equipment disposal**: Planning ahead to recycle used IT subsystems is critical.
Well-managed enterprise systems are no accident: They require careful thought up front. Systems that scale well with good subsystem management and options for integration typically have management that scales. The goals for exascale computing will, however, test not only our ability to scale resources, services, and data but management, as well.

Resources

Learn

- The *IBM Systems Journal* issues "SOA: From Modeling to Implementation," "IBM Service Management," and "Service-Oriented Architecture" are great resources for understanding the theory and practical implementation approaches for integrating a wide variety of applications, data, and resource-management components in a large-scale enterprise system.

- The *IBM Journal of Research and Development* issue "Storage Technologies and Systems" provides excellent in-depth articles on storage management, SAN, NAS, data-protection issues like silent data corruption, and analysis of emergent technologies such as solid-state disk.

- See the IBM Redbooks® on scalable managed storage systems such as DS8000 Series: Architecture and Implementation or smaller-scale IBM BladeCenter® 4Gb SAN Solution to learn more about SAN-based systems that scale, provide high availability, and supply managed resources to host enterprise data in large-scale data centers. In general, the IBM Redbooks are a great resource to help you learn about IBM enterprise hardware and software solutions.

- See Fred Moore's Horizon Web page for strategies, market analysis, and numerous publications on storage system technology and trends, including causes of downtime and data loss in enterprise systems.

- Learn more about security in the "Big iron lessons" articles "Introduction to cryptography, from Egypt through Enigma" (Sam Siewert, developerWorks, July 2005) and "The right coprocessor can help with encryption" (Sam Siewert, developerWorks, August 2005).

- Learn more about high availability and reliability, availability, and serviceability from "Apply RAS architecture lessons to the autonomic self-CHOP roadmap" (Sam Siewert, developerWorks, July 2005).

- Browse the technology bookstore for books on these and other technical topics.

- Get the RSS feed for this series.

- Stay current with developerWorks technical events and webcasts.

Get products and technologies

- Download IBM product evaluation versions and get your hands on application development tools and middleware products from DB2®, Lotus®, Rational®, Tivoli, and WebSphere®.

- SOAP is an excellent open source SOAP implementation for the creation of Web services, and the soap Web page contains many great tutorials for developers and architects new to Web services concepts and implementation.
Check out Tivoli software for service and asset management in enterprise systems.

Expedite your deployment of Web-based services and applications in a Service-Oriented Architecture (SOA) with IBM WebSphere software.

Enterprise systems architects may find VisioCafe useful for systems diagrams that integrate heterogeneous computing, storage, and networking resources.

The Linux-HA project includes many basic software-management tools needed for redundancy management in Linux systems.

Mdadm is a simple management utility for Linux software RAID devices. For links to current status and documentation, visit the Wikipedia mdadm page. Other useful Linux tools for redundant resource management include Logical Volume Manager version 2 (LVM2) and the Enterprise Volume Management System (EVMS).

The Linux Web page Linux SNMP Network Management Tools includes numerous resource links for SNMP monitoring of Linux-based platforms.

Discuss

- Participate in the IT architecture forum to exchange tips and techniques and to share other related information about the broad topic of IT architecture.
- Check out developerWorks blogs and get involved in the developerWorks community.

About the author

Dr. Sam Siewert is a systems and software architect who has worked in the aerospace, telecommunications, digital cable, and storage industries. He also teaches at the University of Colorado at Boulder in the Embedded Systems Certification Program, which he co-founded in 2000. His research interests include high-performance computing, broadband networks, real-time media, distance learning environments, and embedded real-time systems.

Trademarks | My developerWorks terms and conditions