Enterprise Scheduling and Workload Automation

Evaluation Guide and Checklist

How to Choose an Enterprise-Class Workload Scheduling Solution

How to Choose an Enterprise-Class Workload Scheduling Solution

Purpose

Among the handful of disciplines practiced routinely in the data center, management of day-to-day business process workloads may be the most important of all. This is a bold statement given that workload management competes with other important systems management functions such as backup and recovery, network management, and systems security. While these are important disciplines in their own right, there is little dispute that, depending on the size of the enterprise, management of thousands - and in many cases, tens of thousands - of individual mission-critical business processes every day contributes more to the running of the business than any of the other disciplines.

Script-based job scheduling solutions that rely on native operating system utilities, such as Oozie, CRON, NT Scheduler, PERL, and VB scripts, collapse quickly under the weight of their own unmanageable “spaghetti code” and custom language requirements. Given this backdrop, it is easy to see that automation of these workloads is indispensable to the data center.

This document discusses the requirements for workload automation and job scheduling software in complex enterprise environments that include multiple, interconnected applications that touch every aspect of the organization. It provides a comprehensive list and description of vital features that are critical for the smooth and efficient delivery of timely and accurate business processes. A companion checklist is also available to assist anyone who is evaluating this discipline actively. The checklist is a useful tool for gathering information quickly and efficiently about workload automation products being considered.

To set the proper context for a detailed discussion of critical job scheduling functions, this document includes a brief marketplace history. This background will help you understand how changing business and industry requirements affect the data center operating environment.

Background

The job scheduling discipline was first established in the 1970s when it became a critical strategic infrastructure component for large mainframe-oriented data centers. Various products were created and extensively marketed, and it became widely accepted that a reliable solution was required to manage batch-oriented business processes on the mainframe effectively.

During this period of mainframe-centric computing, a common understanding of the main features of job scheduling began to emerge. When UNIX began to make inroads into the data centers in the mid-1990s, IT managers widened their search for solutions to manage batch processing in these new distributed computing environments.

As the shift to UNIX-based applications began, only a few mainframe vendors expanded their offerings to meet this new environment’s demands. Instead, mainframe vendors and many mainframe-oriented data centers experimented with management of distributed workload from the mainframe. As UNIX servers continued to make their way into the core data center, a new group of competitors entered the market with products created specifically to manage nonmainframe, distributed workloads.

Evolution of the Distributed Job Scheduler

The early solutions for distributed computing environments proved to be reasonably robust and did a passable job of mimicking mainframe scheduling features and functions, but they suffered from being first to the marketplace. Eventually all these first-generation products failed in terms of scalability, flexibility, and ease of use. Ultimately, in the late 1990s, mainframe companies, still grappling with the unique issues and challenges of managing jobs in the ever-increasing distributed computing environments, acquired these first-gen solutions.

During this consolidation phase of the job scheduling marketplace, new competitors began to emerge that focused on development of more advanced solutions for job scheduling in the distributed environment. These newcomers had several distinct advantages over their earlier counterparts.

Improved Development Technologies

The first-generation distributed scheduling products were challenged because of the limitations of the base technology used to develop them. By the early to mid-1990s, development tools for the distributed environment had not matured sufficiently to be of significant value, so products from that period were generally developed with older, harder-to-maintain code platforms and used the UNIX Motif standard for their GUI. In fact, a number of the first-generation products were actually ported from other proprietary platforms in the early 1990s, with the usual complications that result from the porting of software products. The combination of older, inflexible development environments and the lack of robust software design tools and any GUI standards created problems as the market began to mature and the requirements for distributed scheduling began to change. These first-to-market products could not keep pace with changing requirements and began to falter in terms of scalability, ease of use, ease of installation, configurability, and maintainability.

True Cross-Platform Development

Like their mainframe predecessors, the early UNIX products essentially had a single-platform orientation, the only difference being that the platform was UNIX instead of MVS or OS/390. This problem came into focus when Microsoft Windows 95 and 98 took over the desktop, and Microsoft Windows NT and 2000 servers became used more widely in the data center. The early UNIX batch scheduling products had the initial lead on Microsoft Windows NT and 2000. However, their shortcomings for true cross-platform, distributed scheduling became evident as more companies attempted to incorporate Microsoft Windows NT or 2000 into their data center operations.

The latest generations of job scheduling products are capable of exploiting the ease-of-use of the Microsoft Windows desktop and recent software development technologies. These products also reflect a deeper understanding of the rigorous architectural and systemic requirements for building a true cross-platform solution that incorporates UNIX, Microsoft Windows, and other platforms.

Scalability

The latest generation of scheduling products has been developed in an era that has large, distributed application workloads already running in the enterprise. The size and volume of workloads commonly seen today had yet to confront early schedulers. When the requirements for increased scalability surfaced, the early scheduling tools did not have the power or design to manage large, distributed workloads.

These shortcomings, which would force users to look for replacements, did not become apparent for several years. Many data centers are now seeking to replace their first-generation job schedulers because of scalability and maintenance issues.

The latest generation of scheduling solutions is well positioned to meet the rigorous demands of modern, distributed data center operations. Unfortunately for the first- and second-generation solutions, it is much easier to design a product to be scalable from inception than it is to try to retrofit scalability into an older design.

True 24/7/365 Operations

When mainframe job schedulers were first created, there were two distinct types of operation in the data center, online and batch, each performed in a specific time period. Typically during the day, databases were online and available only to end users for entering transactions at their terminals. During off-peak hours (typically at night), the databases were brought offline, and batch-processing activities (typically reporting and other batch-intensive jobs) were allowed to commence. This batch-processing window was often as long as 12 hours (for example, from 6 p.m. to 6 a.m.), but soon after the first generation of distributed scheduling products entered the market, business demands began to reduce the size of this batch window.

By the late 1990s, when the latest generation of job scheduling products was being created, two significant issues had started to change the landscape for corporations and software providers: globalization and e-business. This shift in the business landscape began to overwhelm data center operations staff by demanding true 24/7/365 operations: operations that occur 24 hours a day, every day of the week, and every day of the year.

The net effect was the elimination of the batch window and the creation of a new set of requirements for job scheduling tools. Modern workload management solutions have the capability to respond to these evolving requirements with new features for improved scalability and faster process management. They need to respond to a wide variety of operational and environmental events to launch and manage job streams and complex processes. They can't rely merely on the traditional date and time scheduling model. A number of these business-critical features, including event management, cloud infrastructure management, load balancing, global resource flexibility, dependency mapping, and predictive analytics, are discussed in this document.

Modern Data Center Requirements: Movement Toward Workload Automation

Over the years, the job scheduling function has matured as a discipline and has become accepted as a requirement for any comprehensive data center operations management strategy. During this maturation phase, the industry has developed an understanding of what functions are mandatory for a product to be considered an enterprise-class workload management solution. This section of the evaluation guide discusses the core set of features commonly acknowledged as requirements for any enterprise-class workload automation (WLA) tool.

At a strategic level, any solution must provide a means to configure targets and meet service-level agreements (SLAs) that are aligned with business objectives and their attached service definitions. The primary reason for automating these scheduled and event-based processes is to meet SLAs more quickly. These gains allow IT resources to focus on projects that help lower costs and increase business innovation. The first question you need to ask is this: Has the solution been proven in real-world situations to automate the alignment of IT needs with the needs of the business?

Scalable, Flexible, Fault-Tolerant Architecture

N-Tier architectures are flexible, enterprise-class, service delivery systems to consider when evaluating a scheduling solution that can deliver performance, extensibility, and scalability. With its separated logical layers, an N-Tier architecture is a flexible framework that allows the scheduling product to handle heavy workloads while maintaining client stability and reliability. It can also support a flexible deployment model that enables users in remote regions to have closer, dedicated scheduling access that yields faster response times.

In this model, the central processing engine (CPE) focuses primarily on interacting with the scheduling database and guiding the workload. Input and interaction are performed through a browser and handled by a middle-tier web server, which in turn manages the flow of information to and from the CPE. This multilayered and highly flexible architecture separates core functions and helps keep processes running smoothly while supporting the actions of high numbers of distributed and concurrent users.

Important requirements to assess when choosing an architecture are flexibility and fault tolerance. Flexibility is a measure of how easy the product is to deploy in a given environment and how quickly new environments can be added or changed. Can the CPE be installed on any server? Can agents and application adapters be deployed on most machines in the computing environment? Is there an agentless option? Can more than one master scheduling node be running, enabling the various masters to interact in such a way as to provide increased flexibility or fault tolerance?

Some vendors have attempted to disperse scheduling data processors throughout the data center, making each scheduling node the equivalent of the CPE. However, the challenges associated with error recovery in the event of machine or communications failure have largely ruled out this type of architecture as a serious choice. Because the vast majority of jobs are dependent on other job outputs on other machines, there is little justification for moving scheduling data around the network when it can be maintained more reliably and securely on a central resource.

Comprehensive Set of User Interfaces

In computing environments, WLA solutions should provide true graphical management of the workload. Solutions can take a variety of forms but should include an easy-to-understand GUI from which all jobs and schedules can be viewed. This list of jobs should be indexed and filterable so that only certain types of jobs can be made visible or can be sorted to the top of the list. Additionally, the console should be color-coded so that certain jobs - failures, in particular - can be highlighted for easy viewing. A more sophisticated console may have other indicators: for instance, to show users at a glance whether one or more jobs is in an error state or has completed normally.

More sophisticated products have dashboard features that present the ongoing results of scheduled activity in graphical formats, such as line, pie, Gantt, and PERT charts. These charts can give novice and experienced users a quick view of the progress of the daily schedule, the number of jobs that have finished successfully and unsuccessfully, the percentage of the daily schedule that has already been processed, and other common measurements. These analytics views can also show the operator which jobs and workflows are in jeopardy for missing key milestones or SLAs.

In addition to easy-to-use GUIs, WLA solutions should present job streams graphically to make it easier to understand job status and interdependencies. These pictorial representations make the creation and troubleshooting of complex workflows significantly easier to manage. In addition, a modern scheduler should provide graphical approaches to the definition of jobs and job streams, job dependencies, alerts, events, mobile capabilities, and other scheduling tasks, eliminating the need for command-line interactions and custom script creation.

In today’s computing environment, it is standard practice for applications to have a browser- or Java-based GUI, and job schedulers are no exception. A web-enabled GUI simplifies administration of the tool itself and also gives operations personnel the flexibility to log in anytime to monitor and control their scheduling environment from anywhere. It is also important to know how the tool manages access rights for individuals and workgroups. By limiting access to specific jobs or functions, data center operations managers can allow business-unit administrators to run or manage their own jobs, freeing data center resources for more mission-critical tasks.

Self-Service Portals

Many IT departments are handing over some controls of various aspects of the workload management environment to business-unit IT administrators, allowing self-service delivery of non-mission-critical jobs. Data center IT staff can set security access criteria for business unit operations staff to allow them to define and run specific workloads. A web-based GUI, easily configurable using supplied web service APIs, adds a new dimension for management of run-time users and their level of access and what they are allowed to build, run, and change. Such a portal gives end users an easy-to-use way to manage their job-run requests, and it moves repetitive, nonessential activities out of the data center.

When evaluating a self-service web portal, consider whether the processes and security measures it provides makes self-service a reality.

Support for a Wide Range of Modern Enterprise Applications

The primary reason for implementing an enterprise-class WLA solution is to deliver mission-critical applications processing accurately and efficiently. This requirement existed for homogeneous computing environments, and it still exists today.

What has complicated modern SLA delivery is the requirement to integrate successfully with many of today's enterprise applications, such as SAP, PeopleSoft, Oracle E-Business Suite, JD Edwards, Informatica, and Cognos products. Unfortunately, the feature sets and capabilities of most scheduling tools are insufficient for managing individual jobs originating from other vendor-supplied and custom-developed applications within a large enterprise. The capability to connect to a wide variety of distributed business applications allows the data center to:

- Run any end-to-end business workflow, including downstream business intelligence (BI) reporting and analysis tools
- Incorporate rapidly other application steps or data integration into its business processes requirements so that enterprisewide workloads can be automated
- Perform discovery of the options available for the environment so that configuration changes are automatically tailored with the correct options, easing management of multiple jobs and enabling faster configuration with fewer errors

In addition, to reduce overhead and security concerns, WLA solutions ideally should provide an option to run processes using agentless technology: to have no footprint on the target machine.

Some solution providers have extended their products to try to encompass the needs of these critical applications and have created interfaces that make job requests directly to the core scheduling function. Commercially available interfaces integrate these adapter solutions, and it can be worthwhile to evaluate them. At the same time, not all adapter interfaces are created equal, so take the time to understand how a provider actually develops and certifies its interface by asking:

- Is the interface current with the latest application APIs and application version?
- What specific features are available to support the application?
- Does the solution use vendor-approved integration points?
- Can native service-oriented architecture (SOA) jobs be invoked using representation state transfer (REST) web services methods, or Simple Object Access Protocol (SOAP) and Web Service Definition Language (WSDL), or both?

In addition to managing workloads on traditional enterprise resource planning (ERP) platforms, enterprise schedulers must take into consideration connections to applications developed in Java and those developed for SOA platforms. These custom interfaces should support broad Java Message Service (JMS) and web service connectivity so that scheduling operators can take advantage of these messaging and job scheduling capabilities to incorporate Java-based applications and SOAP- and REST-based web service methods into the broader scheduling environment.

A viable solution should also integrate transparently with the latest business intelligence; big data; enterprise data warehouse (EDW); extract, transform, load (ETL); and data backup and recovery applications. These applications process data that spans the entire enterprise and deliver the information required to make the right strategic decisions at the right time. Management of dependency logic and automation of data-based job-step sequencing are critical for the delivery of relevant and timely information for strategic decision making. To achieve these goals, four unique states are necessary:

- Broad enterprisewide visibility of the entire application, storage, and data warehouse landscape
- The capability to automate a wide range of functions, from dependency mapping and application and data warehouse job steps to event and alerts management
- Support for business intelligence application-specific user interfaces that reduce the need to rekey jobs that have already been defined
- API integration for SQL databases and big data processing applications, such as Hive, Sqoop, and MapReduce
- Detailed historical and real-time visibility and analytics into the entire job stream critical path

By integrating an efficient and error-free WLA solution into the companywide decision-making environment, IT operations staff can define connections and dependencies, scheduling the correct business processes in the right order. This capability helps ensure that the correct mission-critical data is always delivered to the right place at the right time.

Database Support

Another aspect of enterprisewide workload management is database support. On the mainframe, it was and is commonplace to use flat, or indexed, files as the underlying data repository for all scheduling objects and rules. When these products were developed and in widespread use, relational databases were not widely used; in fact, IBM DB2, now the leading mainframe database product, did not arrive until the late 1980s. Early distributed scheduling tools followed the flat-file model because they lacked reliable relational database technology. However, when relational technology arrived in the middle 1990s, the leading products adopted it.

Another benefit of relational database technology is the capability to mirror databases, if desired, and to incorporate mirroring into fault-tolerance, backup, and error-recovery strategies, further facilitating unattended operations.

Integration of Big Data Workloads into the Enterprise

Big data has become the latest industry trend. To meet the needs of big data, businesses must address the challenges of capturing and managing the amount, type, and sources of data coming into their data centers. Concerns include the capture, storage, searching, sharing, and analytics reporting of mobile application use. IT commonly encounters these challenges in e-commerce, search, biological research, finance, and business intelligence solutions.

For the truly scalable WLA solution, these concerns should not matter. An enterprise-class solution should scale to millions of jobs and interact with the appropriate applications and databases. You should ask prospective vendors the following:

- Does the solution scale automatically for resource availability?
- Can I schedule from and to multiple staging areas, FTP sites, databases, and new unstructured data manipulation engine environments like Hadoop?
- Can you act as a trusted partner to help me solve the larger network, computing, and storage infrastructure effects on my data center?

Comprehensive Programming Interfaces

In traditional job scheduling environments, the scheduler was the controlling entity and ran or invoked other applications. In today's complex and distributed computing environments, it is often desirable for other applications to interface with, or call, the scheduler directly so that the scheduler can provide services on the application's behalf. Therefore, a full-featured, unified API for programmatic integration is necessary in an enterprise solution in order for IT staff to bring all necessary applications, systems, and technologies into this single interactive SLA delivery model. Enterprise-class schedulers provide comprehensive programming interfaces that facilitate integration with other applications easily. These can be invoked through command-line, API libraries that can be called using languages such as VB, C++, and Java messaging services.

As part of the API, there should be a command-line interface (CLI) for UNIX and Microsoft Windows environments that allows scheduling interactions from a remote system. A CLI-based scheduling approach is important because it provides an alternative means to create and change jobs and automate job definitions and updates that can be faster than use of the standard user interface. Jobs can also be programmed to take advantage of the CLI processing syntax, giving users the capability to dynamically insert or update future jobs. Web service interfaces are also becoming increasingly important to expose scheduling functions as a set of services in an SOA environment.

Today's heterogeneous data centers include composite SOA-based applications, which means that business process management (BPM) engines expose workflows as web services that can be called through standard REST-based methods and SOAP and WSDL protocol support. To function reliably, workflows in these environments must integrate flawlessly with job streams defined as part of a larger jobs-processing environment. The scheduler should simplify workloads in these environments and provide visibility and control over jobs of all sizes and complexities across all operations.

Platform Coverage

Because a typical distributed enterprise includes more than one type of operating system, you must analyze current and future needs thoroughly with regard to the scheduler's platform and technology support. A vendor should support the data center's current and future operating system requirements and have a history of staying current when operating system updates occur. You also should consider the data center's historic and projected preferences for computing technology. For instance, although Microsoft Windows is the clear choice for the desktop, can the WLA solution reside on both UNIX and Microsoft Windows servers? Being positioned to support both operating systems is the best strategy because needs may change over time.

Also consider whether popular application operating environments, such as Microsoft Windows, UNIX, Linux, OpenVMS, IBM OS/400, IBM z/OS, and HP NonStop, are supported. An IBM z/OS data center will also likely want the flexibility of an enterprise solution that provides enterprisewide integration with its mainframe scheduling solution.

In cases in which OS-level processing agents can't be deployed to manage systemwide workflows because of security concerns, machine ownership, or resource constraints, do the solutions being evaluated support agentless scheduling as well? Agentless approaches allow scheduling to be extended to environments in which installation of an agent is not allowed due to security concerns, company policy, or constrained system resources.

Questions to ask include the following: Can Microsoft Windows applications be managed using a remote job adapter (RJA) function? Or should agentless Secure Shell (SSH) capabilities for scheduling on Linux, UNIX, and other SSH-enabled devices, such as networking hardware, be considered?

Agentless solutions have other attributes as well:

- With agentless scheduling, network connectivity is used only when a job is delivered to the target.
- Users do not have to wait for a certified agent to be installed to schedule on these resources.
- System administrators do not need to install or deploy agents if the necessary resources are going to be used only to run a small number of jobs.

Business Calendars and Time Zones

At the center of any scheduling automation solution is the capability to start jobs at a particular date and time. Companies rely on many types of calendars in their businesses: fiscal, manufacturing, payroll, and holiday.

Enterprise workload management tools are expected to provide a comprehensive calendaring function that is easy to use, provides highly visible forecasting data, and permits calendar combinations that achieve date-based business processing requirements. For example, the calendaring function should accommodate rules that define the day on which payroll processing should be run if the pay date happens to fall on a holiday (with different locations having different holiday schedules). Not so surprisingly, complex calendars are required to manage a modern corporation, with some companies managing several hundred, depending on their size and number of geographic locations.

Enterprise business processes are often global in nature, with various steps of the process running in different parts of the world operating in various time zones. Cross-time-zone scheduling should be factored into your decision because modern businesses operate in multiple theaters and must be aware of time-zone idiosyncrasies, such as Daylight Savings Time, which changes at irregular intervals. The best workload management tools can account for these situations and give the user the capability to specify the time zone on a per-job basis. Such a capability helps ensure that the job will launch correctly no matter where the job resides and that it will complete as expected.

Because of the vital nature of these features, buyers are advised to scrutinize calendar and time-zone capabilities to verify that they meet their needs for flexibility, ease of use, maintenance, and complexity.

Dependencies

After calendars, the next most important workload environment feature to address is job dependency. Businesses need to be able to control the processing of certain jobs so that they run not just on a certain date and time, but in a particular order and in conjunction with other tasks.

For example, job B may be dependent on job A, so that job B cannot run unless job A has been completed. However, this simple concept can quickly become complicated. Job B might have to wait for a particular file to arrive, it may have to wait for a user to enter data, it may have to wait for a different job to complete that uses the same database table that it needs to access, or it may be restricted from running during certain hours. When looking at the dependency capabilities of a solution, you need to have a clear idea of the types of dependencies that exist in your environment and to be able to quickly map those dependencies to the product being evaluated. If the product cannot create the needed calendar, run jobs in the correct order, or satisfy the dependency mapping that you need, it is not likely to meet core business needs and will quickly become unusable.

Event-Based Processing

When job schedulers were first created, the methodology for job management was based on time schedules and resource dependencies. Job schedulers were developed to run jobs at the right time and in the right order. Although those requirements still exist, new requirements call for management of jobs on a real-time, event-based, or impromptu basis. Instead of running jobs based on date and time criteria, new WLA solutions should be able to introduce jobs (and entire business processes) dynamically into the processing environment in response to an event, such as a specific file's arrival or the update of a specific database table.

With an event-based workflow processing approach, the WLA solution does not continually ask (by polling) whether an event has occurred, but is told immediately through communication directly with the operating system. Hundreds of events can be used to trigger jobs, including database changes, email arrival, network events, system events (such as memory use or disk use), and insertion of third-party files. Event-based scheduling provides a means of removing latency from batch business processes and allows IT to be more responsive by delivering complex job streams faster and more accurately.

Additionally, if a WLA solution is well integrated into the data center application environment (see the section Comprehensive Programming Interfaces), the tool can process events coming from other systems management products, such as network managers or even built-in or custom applications.

The adoption of Java and .NET for hosting applications has sparked a need to manage processes that are distributed across multiple platforms to fulfill the complete business process. However, neither platform inherently possesses mission-critical WLA features. To support these highly distributed application platforms, an effective WLA solution must accommodate the needs of both real-time and batch tasks across a diverse set of infrastructure platforms.

To make it easy to run processes as events, a WLA solution should also supply specific adapters that support the major business event sources. These should include the following event types:

- Job events (for example, a job completes abnormally, or a job runs longer than expected)
- System events (for example, a lost connection to a scheduling agent)
- File events (for example, a file arrival, a file reaches a size threshold, or a file is stable for period)
- Email (for example, an order is received in a sales inbox)
- Databases (for example, a new row is added to a table, or a row in a table is modified)

In addition, the business may be a need to create customized event sources to trigger actions from applications: for example, the business my want to start a batch process whenever the number of transactions exceeds some threshold or whenever the cumulative value of customer orders exceeds some value. To accomplish this, the solution should expose an interface that allows the applications to externally update custom scheduler variables. Then the tool can watch until the variable value exceeds a condition that identifies an event of interest. When it does so, the solution should be able to trigger the appropriate business-process response.

Resource Management and Workloads in the Cloud

Resource management provides the mechanism for controlling the processing of jobs based on the amount of resources they will consume. This capability prevents jobs from competing with each other for limited data center resources. For example, a job may require higher VPN bandwidth to a trading partner or access to concurrent database client licenses and have any number of other resource constraints. An added complexity in recent years is the move toward cloud-based IT resource management. These pools, or clouds, of resources are made available on-demand as data center environments change. The best WLA solutions have some mechanism to help ensure that sufficient resources are available to service a specific job. You should ask the following questions:

- Can the WLA solution interface with modern resource management tools such as VMware products and Amazon Web Services (AWS)?
- Can the WLA solution process resource requests in private, public, and hybrid cloud environments?
- Can the WLA solution detect and demand increases in resources by instructing an IT process automation (ITPA) or cloud management solution?
- Can resources be decommissioned when the WLA solution has completed the necessary workloads?

Resource requirements can be defined in two ways:

- Resource rules can be used to start a job only if all the required resources are available, reducing the likelihood that the job will fail or be delayed.
- Resource rules can be used to restrict the number of jobs that can run concurrently and that consume one or more of the same critical resources. For instance, you can set up a rule to allow only one job run to run at a time to update a specific database table.

Resource management needs to support both virtual and physical resource models. Virtual resource management allows you to throttle and control workloads and jobs based on a logical resource that may not be tied to a physical system attribute. Physical resource management allows you to throttle and control workloads and jobs based on a physical system attribute, such as CPU availability.

Together, queues and resources provide the means to reduce delays in processing, increase throughput, and manage resource consumption. Be sure that the WLA solution you are considering can run complex workloads based on queues and resource consumption - whether physical, logical (SOA), or cloud resource based and that the solution can autodiscover and update resource information as needed.

Queues and Load Balancing

Modern WLA solutions need to manage both the number and the mix of jobs that can run concurrently on a given system or processes running within a specific application. If too many jobs compete for the same resource, they can affect processing throughput significantly, preventing timely completion.

What makes this facet of the business-processing environment particularly difficult to address is that not all jobs are the same. Not all jobs have the same priority or resource requirements. In addition, there is a finite pool of individual resources, and resource-intensive jobs can deplete this pool, creating resource contention with other jobs and applications.

Enterprise WLA solutions should provide a means of managing the number of simultaneously running jobs by selecting eligible jobs to run based on job and queue priority. For example, a queue can be created that supports only lower-priority jobs. All jobs of a certain priority or lower are placed in a queue that is set to a lower priority than a queue that manages more important workloads. A common approach is to place all user-submitted jobs in one queue for processing during off-peak times, and to place regularly scheduled production jobs in a queue with a higher priority.

A robust solution should also support methods of distributing large and complex jobs across a distributed computing landscape. By giving operations staff the flexibility to manage performance load within their environment, enterprise job scheduling can increase job throughput and optimize service levels.

FTP, SFTP, and FTPS Support

Increasingly, businesses are working to integrate their internal operations and to develop closer links with partners and customers. Companies in financial services, insurance, healthcare, e-commerce, and retail enterprises depend on transparent data exchange to integrate their operations with partners or service providers. The technology most commonly used to exchange data or transfer files among applications and partners is File Transfer Protocol (FTP).

Because there is a close relationship between data exchange and an organization's batch processes, both must be managed transparently and simultaneously to attain the high levels of automation, security, speed, recovery, and accuracy required. However, support for FTP is not enough because FTP is inherently insecure and transfers data without encryption. User names, passwords, and FTP commands can be viewed with packet sniffers. Consequently, a viable scheduling solution needs to support both FTP and SSH File Transfer Protocol (SFTP) or FTP over SSL (FTPS).

Modern workload automation solutions should not require scripting to support automatic file transfers as part of the batch business processes. These solutions should natively recognize when files arrive over FTP or SFTP or FTPS and trigger one or more events to move and process the data.

Framework and Network Management Integration

Many companies today have implemented some type of event or systems management console that provides a variety of features and functions. In many cases, they are implemented to provide a single-console view of mission-critical events happening within the data center and beyond. These consoles are typically "management by exception" solutions. In other words, given the enormous number of events that occur in medium-sized to large data centers, they need to view and manage only exceptional conditions - usually the most serious anomalies.

Because a large number of IT processes run in batch mode, the scheduling solution must be able to integrate with the data center's chosen framework or events management console. Typically, this integration is twofold. First, the integration should allow the scheduler, using standards-based messaging protocols, to inform the console when some type of job failure occurs. Second, the integration should allow the system and network management tools to monitor the scheduler and its associated infrastructure. Although not as obvious, this integration gives the event management framework operator the capability to monitor the health of the scheduler itself. In this way, if the scheduler or one of its distributed components experiences an error, the management console should be able to report on any failures.

Security

Security is a vital requirement of any job scheduler. Clearly, if a WLA solution is in charge of running the mission-critical processes in the data center, access controls must be in place. In addition to controlling access to the scheduler, an administrator may want to control access to individual features in the product by user, by group, or by job.

The primary security controls for any scheduling solution should pertain to its capability to authenticate and track users in the system. You can use a single source of authentication by connecting to a standard corporate employee directory database. This approach allows administrators to control access to the scheduling solution and simplifies the process of adding, removing, and changing user access. Modern WLA solutions should also be able to revalidate user status against the directory service continuously, making the necessary rights changes in real time as the directory changes. The capability to track what the user does within the scheduler is a logical benefit of any detailed security authentication feature. Can the solution record the steps of users as they interact with definitions, jobs, and schedules?

Workgroup-based security management is another critical capability because it simplifies security management by allowing administrators to specify security policies for each job or job group and for events and actions owned by a group. In this model, users must be members of a workgroup in order to own a job, event, or action, and they can be tracked by the solution for audit and change-management purposes.

Management of security access down to the individual user level should be a requirement because of user diversity: not all users need access to all scheduling functions. For example, the IT administration team typically has broad access to the tool, but access to certain jobs or certain features within the scheduler may need to be restricted. Other personnel may have the authority to create certain business calendars or jobs, but not the authority to run those jobs in production. In some organizations, business group IT administrators are given limited access to so that they can monitor the progress of jobs they own through self-service portals.

The main features to look for are ease of use and detailed control. Ease of use is necessary for quick authorization of changes for a given user or group in response to changing business needs. Compromising security policies because a particular policy is deemed too difficult to implement is shortsighted. Detailed control, or refinement, is important because of the need to make only certain product features available to certain users and to track "who, what, and when" for any process to the individual user level.

Audit Trails

Many people relate audit trails to security. While there is a strong connection, audit trails have more than one benefit: with audit trails in place, operations personnel can monitor and, when necessary, undo unauthorized changes to the scheduling environment.

For example, even authorized and well-intentioned users can make mistakes when modifying the production schedule. Audit trails provide the means to understand precisely what changes were made to the production environment and who made them. Given the rapidly changing requirements of the modern data center, it is imperative that all changes to the production environment be recorded in detail.

A related topic is the concept of system or scheduling logs. Whereas true audit trails typically focus on "what changes were made by whom and when," logs simply document the results of the processing of a particular job or schedule. Well-organized logs give the user a clear indication of exactly what workload ran, when it ran, and its results.

Reporting and Analytics

A full-featured WLA solution should provide detailed critical-path analytics and historical reporting to business end users, IT executives, scheduling managers, and IT staff. Good reporting on the performance of workload automation environments and the SLAs for business processes is an important evaluation factor.

Real-time visibility and control and a historical view of the scheduling environment are critical requirements for business operation success. Scheduling environments are dynamic; a well-tuned operation is a constantly changing target. Job volumes, processing times, and processing patterns change as the business evolves. It is mandatory to employ business intelligence platforms that provide detailed reporting and support operations forecasting. This capability enables staff to be more efficient and to concentrate on proactive tasks, such as tasks to optimize schedules and increase service level quality. An easy-to-use, powerful, and extensible integrated business intelligence solution is essential to respond to the changing business-processing environment and to user needs.

You should ask these questions about a solution's reporting and analytics capabilities:

- Can the analytics tools provide accurate discovery of entire job streams?
- Do I have the capability to deliver real-time job-run information to business end users?
- Can I accurately predict when a job will be complete, and will I be alerted proactively if resources will affect my capability to meet an SLA?
- Can the solution run automated alerts and remediation functions based on environmental or system failures or constraints?

Rapid access to accurate end-to-end workload reports can help IT proactively detect problems so that service levels are not compromised. Sophisticated analytics can provide insight to support the auditing and compliance processes and supply new strategies to help improve business process efficiency. Ideally, prerequisites for analytics tools should be the delivery of predefined reports and the capability to create custom reports. They should gather the data that IT and business managers need to troubleshoot, trend, and analyze enterprisewide workload performance.

Alert Management and Autorecovery

Because schedulers perform such a vital role by automating critical business processes, they must be able to generate alerts when something unusual happens in the processing environment. Alert management should be flexible enough to handle a wide array of potential events. It should also be extensible so that it can identify and manage events of the user's choosing.

Much like autorecovery functions, alerts functions should respond differently to different types of events. In fact, alert management and autorecovery functions often need to work in conjunction with one another. In a typical scenario, a job failure initiates some type of recovery action. At the same time, notification of the failure may need to be sent to a designated person. This notification may include an email message to a specific user, an XML page to a technician, and a message to a central management console. Additionally, this alert function should allow acknowledgment of the alert so that the scheduler is informed when the appropriate person has received the alert.

As with other features, ease of use and flexibility are extremely important. Some products claim to deliver these types of alert capabilities, but close examination reveals that to do so they require the user to write custom scripts. This requirement defeats the purpose of an automated workload management solution.

Some failures are not meaningful enough to stop subsequent processing, while others will corrupt critical downstream processing and therefore supply inaccurate reports and data analysis. Therefore, the solution should allow operators to create multiple types of recovery scenarios. For instance, in some cases, if an error is deemed severe enough, it may be sufficient for the scheduler simply to stop processing. In other cases, when a specific type of error occurs, the preferred approach may be for the scheduler to back up several steps and rerun those jobs. If these remediation steps are not sufficient, the operator may run a series of related recovery actions, such as restoring the database before attempting to run the jobs again, or these actions could be programmed into the solution, freeing valuable operator time.

The products being evaluated should be able to take distinctly different actions based on the relative severity of each problem encountered. At minimum, the selected solution should allow the user to stop processing, continue, or rerun processing and run a series of recovery actions before moving on to a subsequent step. It the best case, the WLA solution should integrate with help-desk applications and have the capability to connect to ITPA applications that can run diagnostic and remediation workflows to correct system errors. Then, having corrected the system error, the ITPA solution should tell the WLA solution to automatically rerun the workloads in question, with each step logged and appropriate alerts sent at the right times.

Ease of Use

Although "ease of use" is sometimes difficult to describe, most users tend to know it when they experience it. This feature may be viewed as a "soft" requirement, but it should not be underestimated. Staff turnover is frequent, and often the most experienced people are not onsite at the precise moment when a failure occurs or the daily schedule needs modifying.

For most users, the familiar Internet browser interface is the most intuitive of all GUIs. It also has the added benefit of simplifying remote access. Newer products have generally adhered to this type of interface, and in some cases have added graphical capabilities to simplify usability further. Note that tools with intuitive interfaces are not only easier to use, but ultimately produce fewer user errors and enable faster recovery when errors do occur.

A critical step in making a scheduling solution easy to use is eliminating the need to create custom scripts to define jobs and job streams, job dependencies, alerts, events, and other scheduling tasks. Scripts may provide easy quick fixes to immediate problems, but they tend to persist in the solution while awareness of who created and manages them diminishes over time, giving rise to emergency responses when errors occur.

A modern scheduling and batch business process automation solution should be able to deliver flexible and repeatable solutions for these tasks through the use of drop-down menus, point-and-click selections, and drag-and-drop user techniques. Avoiding custom scripting can improve the manageability and quality of service delivered by a job scheduler significantly.

Ease of Installation and Configuration

Given the dynamic nature of the modern data center, with new hardware and applications being added almost continuously, a WLA solution must be simple to implement and reconfigure. Ideally, the solution should be able to reside on standard server operating systems in the data center, and GUIs should be browser based and require little client-side management.

During an evaluation phase, configuration and reconfiguration criteria can be difficult to assess, but potential buyers should observe at least the initial product installation. Can the initial installation be self-service? If not, can staff members at least follow along with the vendor to understand the installation process? How long does the implementation take? Hours? Days? Any product taking more than a day for basic implementation will probably not be any easier to reconfigure when the operating environment changes, and therefore is not a good candidate for buyers who anticipate changes to their environments.

Conclusion

An enterprise-class WLA solution is a core component of an overall data center systems management strategy. With job scheduling and end-to-end workload automation, enterprises can begin to manage the flow of essential business processes proactively, delivering increased value to the business, meeting and exceeding service levels, and lowering operating costs.

The core features of job scheduling are well understood and have stood the test of time; however, the market continues to evolve, requiring new features, continuous innovation, and new application and data support. This evolution guides the move toward end-to-end workload management, complex and scalable data management, cloud-enabled solutions, self-service capabilities, and event-based automation. Each data center must evaluate its business processing strategy continually to determine workload management requirements and whether the current solution continues to meets the organization’s ever-changing requirements.

Cisco Data Center Solutions

Cisco changes the economics of the data center by unifying computing, storage, networking, virtualization, and management resources into a single platform. The result is operational simplicity and business agility, which are essential for modern data center management, cloud computing, and deployment of IT as a service (ITaaS).

The management components of the Cisco® data center architecture provide end-to-end management software solutions. These solutions provide an intelligent, automated approach to IT management. They offer speed and enterprise-class reliability while simplifying deployment and operation of physical and bare-metal, virtual, and cloud infrastructure.

For More Information

- For more information about workload automation solutions, see http://www.cisco.com/go/workloadautomation.
- For more information about solutions for data center applications and big data, see http://www.cisco.com/c/en/us/solutions/data-center-virtualization/data-center-applications/index.html.
- For more information about converged infrastructure and cloud management, see http://www.cisco.com/c/en/us/solutions/data-center-virtualization/unified-management/index.html#~products.

Enterprise Scheduling and Workload Automation Evaluation Checklist

Enterprise Scheduling and Workload Automation Evaluation Checklist

Introduction

When purchasing an enterprise workload scheduling solution, you should gather as much information as possible to enable an objective comparison of competitive products' main capabilities. You should seek customer references, analyst recommendations, and press articles to supplement materials collected directly from the suppliers of the products under review.

This document provides a detailed list of evaluation criteria that you can use to benchmark the features and functions of the solutions that your organization is considering. The checklist provides a tool for assessing thoroughly how well a given product can meet the needs of your enterprise now and in the future. Of course, this checklist is merely a starting point. Any requirements that are highly specific or critical to your operating environment should be added to the list.

Important Considerations and Checklist

In the complex distributed computing enterprises commonplace today, it is difficult to overestimate the crucial role that workload scheduling plays in keeping mission-critical data flowing efficiently. While much attention is focused on high-profile business applications that are visible to a company's end users, the technologies that allow them to perform as designed, transparently and almost instantaneously, are the foundation for successful IT operations.

Given that tens of thousands of jobs routinely pass through a data center daily, a flexible, scalable job scheduler is more important than ever. When evaluating scheduling solutions, you should have complete confidence in both the product and the vendor that you choose. Your due diligence in making this purchase decision will be rewarded with peace of mind.

Company Overview

- Financially stable company with global presence
- Strong technology reputation
- Expertise in data center automation
- Focused on the development, sales, and support of data center automation solutions
- Continuous innovation
- Customer support is 24/7/365
- Recognized by the market in the data center solutions space

General Product Functions

- Proven record of delivering on IT automation and business delivery goals
- Recognized for product ease of use and support
- Job definitions that do not require developer knowledge to configure or support
- Initial installation, configuration, and job processing that can be completed in three hours
- Intuitive web browser interface for ease of use and error-free operation
- N-tiered architecture for outstanding scalability

- Multiplatform support (for example, Microsoft Windows, UNIX, Linux, OpenVMS, IBM OS/400, IBM z/OS, and HP NonStop)
- Proven systems design for easy maintenance, low cost, and quick implementation
- Fault-tolerant architecture for automated system failover and uninterrupted scheduling, with cluster-independent engine and master support for clustered high-availability environments
- Use of a relational database to help ensure performance, reliability, security, and ease of reporting
- Scalable to hundreds of thousands of jobs per master, with thousands of agent connections
- Support for intermaster dependencies for multiple data centers and geographic locations
- Easy definition, processing, and management of FTP, SFTP, and FTPS transactions
- Remote monitoring and management using wireless devices
- Agentless scheduling options for job processing on systems on which scheduling software cannot be installed
- Extensibility to include scheduling of jobs on business applications and data center management solutions
- Extensibility to include scheduling on custom-developed applications
- Support for multidomain Microsoft Active Directory authentication

Comprehensive Set of User Interfaces

- Simple-to-use GUIs
- Anytime, anywhere browser-based login and access to all forms, reports, and job status
- Command-line access to solution's major functional areas
- All jobs indexed and filterable so that only certain types of jobs are visible or can be sorted
- Dashboard features that present the ongoing results of scheduled activity in graphical formats
- Capability to present job streams graphically to make job status and interdependencies easier to understand
- Graphical approaches to definition of jobs and job streams, job dependencies, alerts, events, mobile capabilities, and other scheduling tasks
- Fat client BUI configurations for power users
- Self-service portals allowing self-service and ITaaS delivery of non-mission-critical jobs with detailed control of processes and enhanced security measures

Enterprise Application Support

- Transparent integration with major applications, such as SAP Business Warehouse, Cognos, Informatica, Oracle E-Business Suite, PeopleSoft, JD Edwards EnterpriseOne/World, SAP ERP, BusinessObjects, Hadoop, RHEL, SUSE, OEL, CentOS, Win, Solaris, and Cisco UCS® Manager
- Use of vendor-approved integration points, software development kits (SDKs), and APIs
- Support for other applications, such as business intelligence; extract, transform, load (ETL); data warehousing; database scheduling; and backup applications
- Support for drop-down lists of available jobs, agents, variables, and parameters
- Efficient means to transfer job definitions from development and user-acceptance testing (Dev/UAT) environments to the production system
- Integration adapters that can be installed and activated in one hour

- Support for an application-specific user interface reduces the need to rekey jobs that have already been defined
- Application adapter that does not use a command line with long commands that pass multiple parameters
- Integration that does not require scripting language, but with scripting available to meet complex, custom requirements
- Support for custom applications through API and command-line integration
- Web services interface for the creation of custom extensions, such as for self-service portals
- Support for scheduling web services environments, which include Simple Object Access Protocol (SOAP) - based and representational state transfer (REST) - based methods

Date- and Time-Based Scheduling

- Job scheduling by days, weeks, dates, times, intervals, and events
- Dozens of predefined business calendars
- Job scheduling using a custom fiscal calendar interface
- Job scheduling based on other job statuses, such as Abnormal, Normal, and Error Occurred
- Scheduling management available through command-line interface (CLI) and API
- Support for nested job groups that inherit job characteristics
- Support for workload balancing with multiple options for distributing workloads
- Scheduling based on user-defined exit codes and exit code ranges
- Scheduling of jobs and dependencies by parsing output for specific character strings
- Storage of estimated, actual, minimum, and maximum runtimes for each job
- Detailed scheduling control to account for multiple global time-zone idiosyncrasies

Event-Based Scheduling

- Schedules based on file arrival, file existence, file size, and file stability
- Schedules based on database events
- Schedules based on changing variables (static, dynamic, and user initiated)
- Schedules based on email events

Resource Management and Workloads in the Cloud

- API interfaces with modern resource management tools, such as VMware and Amazon Web Services (AWS)
- Resource requests in private, public, and hybrid cloud environments
- Capability to detect and demand increases in resources by using an IT Process Automation (ITPA) or cloud management solution
- Capability to decommission resources after completing the necessary workloads

Job Processing and Recovery

- Automatic job restart based on job error or system failure
- Insertion of recovery jobs into the schedule automatically for processing
- Support for multiple queues, job priorities, and queue priorities that include priority bumping
- User-defined parameters for job history information
- Support for impromptu processing of jobs as well as predefined schedule
- Support for operator overrides for parameters and dependencies
- Resource management to support both virtual and physical resource models

Security

- Flexible and easy-to-administer access rights
- Dynamic access and ownership rights validation
- Integration with standard employee directory services for robust user authentication and group management
- Predefined and customizable security policies, including role-based security using existing user IDs and passwords
- Password encryption for all agent and application user IDs
- Capability for administrators to grant and deny highly specific access to jobs, agents, and user capabilities and enable view-only and operator-only consoles
- Secure Sockets Layer (SSL) data communications between solution components
- Support for SFTP to eliminate the need to send data, passwords, and FTP commands in the clear or in an unsecure manner
- Encrypted communications between master and agents

Auditing

- Extensive logging that includes all user interactions, job-run statuses, and error messages
- Capability to set error, audit, and diagnostic levels for complete error handling and resolution

Reporting and Analytics

- Out-of-the-box reporting capabilities that deliver information about the entire scheduling environment from a single pane
- Detailed historical and real-time visibility and analytics into the entire job-stream critical path

Alert Management

- Predefined system alert and job event triggers for notification
- Alert notifications, including email, text, paging, console, Simple Network Management Protocol (SNMP), Microsoft Windows Management Instrumentation (WMI), logging, and variable substitution
- Capability to include multiple job variables with any notification action, including job name, agent name, job output, actual runtime, exit code, and pager or cell number
- Management framework integration using two-way communications for alerting and interaction

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Printed in USA C07-704567-01 05/14