Job Scheduling on Windows
White Paper

Custom Research Note

Author: Dr. Mike Gilbert, Legacy Directions
Technical Review: Michael Dee Hester, Microsoft
Published: December 2006
For the latest information, please see http://www.microsoft.com/mainframe
Analyst Dr Mike Gilbert of Legacy Directions Limited wrote this custom research note for Microsoft Corporation. Interested readers should contact the author at mike.gilbert@legacydirections.com to arrange further discussion or an interview.
# Contents

Introduction ................................................................................................................................. 1

A brief review of the market ........................................................................................................ 2

Job scheduling explained ............................................................................................................ 3
  - First-generation job scheduling: batch processing ................................................................. 3
  - Second-generation job scheduling: workload management .................................................. 3
  - Third-generation job scheduling: workload automation ....................................................... 4

Mainframe-based job scheduling products ................................................................................. 6
  - BMC CONTROL-M for z/OS .................................................................................................. 6
  - CA Unicenter CA-7 Workload Automation, ................................................................. 6
  - CA ESP Workload Automation for z/OS .......................................................................... 6
  - IBM Tivoli Workload Scheduler for z/OS, ........................................................................ 6

Distributed-based job scheduling products .................................................................................. 8
  - AppWorx ............................................................................................................................ 8
  - ASCI ActiveBatch .............................................................................................................. 8
  - ASG-OpsCentral .............................................................................................................. 8
  - BMC CONTROL-M for Distributed Systems ................................................................. 8
  - CA Unicenter AutoSys Workload Automation ............................................................... 8
  - CA dSeries Workload Automation .................................................................................. 9
  - IBM Tivoli Workload Scheduler .................................................................................... 9
  - ORSYP Dollar Universe ................................................................................................. 9
  - Redwood Cronacle .......................................................................................................... 9
  - SMA OpCon/xps .............................................................................................................. 10
  - Tidal Enterprise Scheduler ............................................................................................. 10
  - UC4:global .................................................................................................................... 10
  - Vinzant Global ECS ...................................................................................................... 10

Analysis of job scheduling on Windows .................................................................................... 12
  - Technology ...................................................................................................................... 12
  - Market forces .................................................................................................................. 13

JCL on Windows ......................................................................................................................... 15
  - JCL emulation products ................................................................................................... 16
  - Using a job scheduler with JCL on Windows .................................................................... 16

Windows PowerShell .................................................................................................................. 18

Conclusion .................................................................................................................................. 19

Appendix 1. First-Generation Job Scheduling: Batch Processing ............................................ 20
Appendix 2. Second-Generation Job Scheduling: Workload Management ............................... 21
Appendix 3. Third-Generation Job Scheduling: Workload Automation .................................... 22
**Introduction**

In the traditional data-processing industry, we had come to view IT processes as either online real-time processes or scheduled batch processes. As individual users, we experienced online access through graphical client/server tools and online transaction-processing (OLTP) monitors. However, much of the computing activity was scheduled to run at certain times of day using batch scheduling systems.

Today, we inhabit a world in which everything we need seems to be available online, so we might suppose that no one uses batch scheduling anymore – right?

Wrong. The market for batch job scheduling software is growing at about 6 percent compound annual growth rate (CAGR). This growth is driven by specialist systems management vendors who alongside BMC Software, CA, and IBM are competing in a vibrant market for workload automation, which is the next generation of job scheduling software. Job scheduling and workload automation are at the very heart of today’s real-time enterprises.

Technology advancements in Microsoft® Windows Server® and in the underlying multicore, multiprocessor architectures being developed by Intel and Advanced Micro Devices have combined to provide a Windows Server platform that is capable of handling workloads equivalent to a 4000-MIPS mainframe computer *. Enterprise-level organizations that install the Windows Server platform need job scheduling tools to manage such large processing capacity effectively.

The purpose of this paper is to compare mainframe-based and distributed-based job schedulers and to assess the readiness of Microsoft Windows-based job scheduling technology to manage workloads at the enterprise level. This paper can also help IT organizations that plan to migrate mainframe workloads to the Windows Server platform as part of a legacy modernization initiative.

The paper opens with a review of the market for job scheduling, followed by a brief outline of its growth path. Detailed comparisons of a selection of both mainframe-based and distributed-based job scheduling products provide the basis for analyzing job scheduling capabilities on each type of platform. This paper also looks at two products on the market that offer JCL emulation on the Windows Server platform. It concludes with a glimpse at what Microsoft is doing with scripting and task scheduling.

---

* Independent research based on data published by the Transaction Processing Performance Council (http://www.tpc.org) assumes a 16-processor server and SAN configuration.
A brief review of the market

Job scheduling software is included within the larger systems management software sector. According to industry analysts, the market for job scheduling software licenses in 2005 was $540 million. Job scheduling software sales are growing at approximately 6 percent CAGR, which gives a predicted market size of $720 million in 2010.

The market for mainframe-based job scheduling software is relatively stable. Analysts estimate that there are between 16,000 and 18,000 IBM mainframe computers in production use at approximately half that number of customer sites. This market is dominated by the traditional job scheduler vendors BMC, CA, and IBM. Small growth is achieved by upgrades and add-on sales for scheduling ‘agents’ which can incorporate workload running on new distributed platforms into the overall scheduling mix.

Growth in the market is driven by the emergence of job schedulers based on distributed platforms where platform sales continue to rise. This growth, plus the continued migration of applications from older platforms such as VSE, MPE, Tandem NonStop, VMS, and GCOS, are fuelling investment by many vendors in new features for rich event-based automation of IT operations on contemporary platforms.

In May 2006, CA acquired Cybermation to gain their ESP job schedulers. This purchase augments CA’s current CA-7 and AutoSys solutions with an injection of modern workload automation technology. From a competitive standpoint, this will also help CA increase its market share in the mainframe arena by building on the success of the Cybermation ESP: mSeries solution. In July 2006, ASG acquired Diversified Software Systems, which delivers JCL automation software products and services, to strengthen their mainframe-based workload management solutions.

There is renewed industry interest in migrating traditional mainframe workloads to distributed platforms with system software which provides a compatible operating environment for IBM mainframe applications. JCL emulators help make this transition a smooth one. Fujitsu Software expanded its COBOL migration products in April 2006 with its release of NeoBatch, which supports JCL execution on the Windows platform. Micro Focus has addressed COBOL migration and support for JCL on distributed platforms with the Micro Focus Server product line for mainframe migration. Both Fujitsu and Micro Focus provide integration with job schedulers.
Job scheduling explained

This section charts the progress of job scheduling through three generations: batch processing, workload management, and workload automation. (Note that different vendors use different terms to describe their products and features.) At the most basic level, batch processing is the scheduling of non-interactive jobs to optimize use of the resources of a single computer. Later workload management improvements distributed the job scheduling across clusters of servers, and offered calendar scheduling features. Recently, the integration of Web-based applications and the scheduling of jobs based on real-time events have defined workload automation.

First-generation job scheduling: batch processing

Scheduling computing tasks by means of batch processing remains the backbone of most mainframe-based IT operations. Batch processing schedules jobs to optimize use of costly computing resources. One of the primary goals of early batch job scheduling was to keep CPU usage as close to 100 percent as possible, night and day. Today however, operations managers prefer to keep processing capacity in reserve to handle peak demands.

Present-day job schedulers offer these batch processing features:

- Automatic restart and recovery
- File management
- Integration with security systems
- Operator alerts
- Scheduler failover
- Service classes
- Spooling devices
- Scheduler throughput
- Workload failover

This is the feature baseline for job scheduling products. More advanced features for second-generation workload management and third-generation workload automation build on this baseline. (Appendix 1 describes these features in more detail.)

Second-generation job scheduling: workload management

The business demands for information processing continually increase in volume and complexity. These needs have placed great burdens on operations staff. Organizations have diverse application workloads to process, which may include packaged applications, several different platforms, and significant integration of operations across business functions.
This change has driven a second generation of job scheduling features, called workload management. Workload management provides the capability of managing jobs that are spread across many diverse platforms from a central point of control. It provides functions that define processing priorities by business deadlines (calendar scheduling) and by cross-functional dependencies.

Workload management, which builds on the basic batch processing features, is achieved through richer scripting features, more sophisticated scheduling engines, and multi-system, cross-platform workload balancing capabilities. In summary, these additional features are:

- Cross-platform support
- Cyclical scheduling
- Deadline scheduling
- Inter-dependent jobs
- Dynamic resource balancing
- External task monitor
- Multiple calendars and time zones
- On-demand scheduling
- Scheduling of packaged applications
- Scheduling of Web applications
- Single point of control
- Workload analysis

Appendix 2 describes these workload management features in more detail.

**Third-generation job scheduling: workload automation**

The most recent generation of job scheduling has been driven by the broader integration needs that have arisen from Internet-based business activities. To compete in their own markets, organizations have been forced to rethink the way they do business. Customers want self-service applications, and expect such applications to provide real-time, integrated access to personalized information. For these scenarios, any product and departmental silos must be hidden behind Web applications that present the company, products, and services to the outside world.

To address these needs, IT organizations are building complex, real-time, automated business processes using a patchwork of existing and packaged applications. A Web-based product order may trigger several dependent applications to complete the ordering process. Examples include order tracking, billing applications, product assembly and packaging instructions, shipping notifications, and inventory management. A workflow of jobs that use existing applications is often built with batch integration technology that is provided by job scheduling software.

Another innovation in job scheduling products is event-driven schedules. This represents a clear move by batch tool vendors into the online camp. Jobs may be
scheduled to respond in real time to business events such as the web-based product order described above. A process percolates through a chain of activities—a workflow—using events such as file creation, e-mail arrival, new log entries, or console messages to trigger the next step in a sequence automatically.

Dynamic business models are driving the need for more dynamic workload scheduling. Job scheduling products are evolving to provide "live" process management features to meet online response deadlines, features such as critical path monitoring and dynamic resource adjustment. Workload planning and forecasting are essential to ensure that the business processes will perform during times of heavy customer demand, such as the Christmas shopping period. This third generation of job scheduling software is workload automation.

Workload automation, which builds further on the workload management and batch processing features described earlier, permits IT departments to define service level agreements (SLAs) for critical business services and to monitor performance against them. These additional workload automation features are:

- Conditional dependencies
- Critical-process monitoring
- Dynamic schedules
- Event-based automation
- Graphical workflow definition
- Mobile access
- Programmable scheduler API
- Virtualization
- Workload forecasting
- Workload planning

Appendix 3 describes these workload automation features in more detail.

Application integration software is moving to address the same set of automation needs, but from an online perspective. TIBCO and webMethods products, and the Microsoft Biztalk™ server, are examples. The line between batch and online processing is blurring as technology converges on solutions to address the current thirst for process automation.

These technologies come from different starting points and are typically used by different members of IT departments. Application integration software is used by developers to construct composite hard-wired applications spanning multiple platforms, across organizations and between businesses. Job scheduling products are used by operations staff to maximize operational efficiency to meet service deadlines. There are indications that job scheduler vendors are continuing this trend toward convergence as they find new customers: application development teams and departmental users who are looking for ways to put together new business processes quickly. This trend is in turn driving new feature requirements such as the ability to run multiple instances of a schedule side by side with no resource or data-flow contention.
Mainframe-based job scheduling products

Four job scheduling product families that are hosted on the IBM mainframe are available from BMC, CA, and IBM. CA now offers two product families because of its acquisition of Cybermation.

BMC CONTROL-M for z/OS

BMC Control-M enterprise job scheduling underpins BMC’s Operations Management product line. BMC Control-M for z/OS was designed and constructed from the outset as a multiplatform job scheduler. Job scheduling agents support more than 20 platforms including OS/400, Tandem NonStop, Unisys 2200, Unix, Windows, and z/OS. BMC Control-M for z/OS includes options to interface with SAP, Oracle, and PeopleSoft. A complimentary product, BMC Batch Impact Manager, ensures that processes complete on time.

Each scheduling server in a Control-M implementation defines a scheduling region. Multiple regions can be federated to manage cross-region workload dependencies. Recently, BMC introduced a feature called “agent-less” scheduling, which allows users to schedule workloads on platforms that do not require scheduling software to be installed. As an example, this feature can be used to simplify operations across a large number of desktop computers.

CA Unicenter CA-7 Workload Automation,

CA positions job scheduling within CA Workload Automation, which is part of the CA Enterprise Systems Management (ESM) product line. CA Workload Automation includes a number of complementary Unicenter job management products and the recently acquired Cybermation ESP product line.

At the heart of CA Workload Automation for z/OS are Unicenter CA-7 Workload Automation and Unicenter CA-11 Restart and Tracking. Unicenter Workload Control Center (formerly Unicenter Enterprise Job Manager) is a management portal interface that provides the central control for enterprise-wide job scheduling operations.

CA-7, developed to supplement early JES2 installations, has over 1900 customers worldwide. It provides many workload management and workload automation features. CA-7 has a powerful and mature scheduling engine built on z/OS to take advantage of the Workload Manager (WLM) component of z/OS, but lacks the rich event-based architecture of CA ESP (see below). CA has plans to combine the strengths of these products in future releases.

CA-7 supports SAP, Oracle, and PeopleSoft. CA's Universal Job Management Agent can manage custom applications on Unix, Linux, Windows AS/400, OpenVMS, and Tandem platforms.
CA ESP Workload Automation for z/OS

CA ESP Workload Automation (formerly Cybermation ESP: mSeries) for mainframe-based environments has over 200 customers. CA ESP can initiate jobs from a rich array of event sources and can schedule multiple occurrences of event-triggered jobs. The primary scheduling servers can failover to another server in a Parallel Sysplex. The CA ESP server runs on z/OS but can manage workloads running on a wide range of platforms through ESP agents for z/OS, OS/400, OpenVMS, Windows, Unix, and Linux. CA ESP also includes agents to support SAP, Oracle, and PeopleSoft applications. The CA ESP server uses CA ESP Encore to manage z/OS job recovery.

IBM Tivoli Workload Scheduler for z/OS,

IBM Tivoli Workload Scheduler for z/OS, based on the original Tivoli Operations Planning and Control (OPC) product, provides mainframe-based workload management with capabilities to manage workload on Linux, OS/400, Unix, and Windows via scheduling agents. When a job fails, IBM Tivoli Workload Scheduler for z/OS handles automatic dataset cleanup to restart the job. The central scheduler provides automatic recovery from system failure by using a hot standby architecture. Users can configure alternate workstations to automatically reroute workload in case of primary workstation unavailability. The Workload Scheduler balances workload by integration with the IBM Workload Manager (WLM) component of z/OS.

IBM technology uses an open interface and can be used with other scheduling engines. Fault-tolerant agents carry a local schedule plan to continue workload processing in the event of network or scheduling server failure. The related product IBM Tivoli Workload Scheduler for Applications integrates workload scheduling with Oracle, PeopleSoft, and SAP business applications.
**Distributed-based job scheduling products**

In comparison with mainframe-based job schedulers, a much broader choice of job scheduling products are hosted on distributed platforms. CA and IBM have products for these platforms, and BMC’s product can be hosted on the mainframe or on a distributed platform. Additional vendors have developed distributed platform solutions, and use agents to provide full cross-platform workload management on Windows, Unix, Linux and z/OS from a central distributed platform server. These vendors are typically well-established single-product companies who focus on systems management tools. These products are described briefly below.

**AppWorx**

AppWorx Corporation, formed in 1990 provides job scheduling software and consulting services and has 500 customers worldwide. AppWorx software manages workloads on Linux, OpenVMS, OS/400, Unix, and Windows platforms. AppWorx also supports Oracle, PeopleSoft, Retek, SAP NetWeaver, and SunGard Banner. Its AppMaster product provides a central point of control, and a graphical analysis package provides operations forecasts and reports.

**ASCI ActiveBatch**


**ASG-OpsCentral**

ASG-OpsCentral is part of ASG’s Operations Management suite. ASG-OpsCentral provides central workload management on Windows or Linux. It can schedule jobs on z/OS through ASG-Zeke and on Linux, Unix, and Windows through ASG-Zena (which includes support for SAP, PeopleSoft, and Oracle applications). The full suite consists of many products which together provide job scheduling functionality, including JCL tools as well as workload analysis and planning. Automated restart management is also provided for z/OS.

**BMC CONTROL-M for Distributed Systems**

Although BMC Control-M incorporates a central scheduling server and remote agents to control satellite workloads, the server engines that run on Unix, Windows, or z/OS are functionally the same. By using the same server on both the mainframe and distributed
platforms, BMC reports that there is no difference between mainframe-based and distributed-based scheduling. Platform differences appear in the way agents manage local execution features such as job scripting (for example JCL or command files), workload balancing, job restart (which must be scripted on distributed platforms), and application-to-application data flows (pipes in Unix).

**CA Unicenter AutoSys Workload Automation**

CA offers Unicenter AutoSys Workload Automation for job scheduling on distributed systems. The Unicenter Workload Control Center also supports AutoSys and provides centralized administrative control for workload scheduling operations.

AutoSys, developed originally for the Unix platform, has over 2000 customers worldwide. AutoSys includes forecasting, rich event-based automation, and dynamic workload management. As AutoSys has evolved to schedule jobs on many platforms, it includes a built-in cross-platform scripting language. Developers specify job recovery actions in application coding and job scripts; AutoSys manages automatic recovery of workflows in the event of a system failure.

AutoSys supports SAP, Oracle, and PeopleSoft applications, and custom applications on Unix, Linux, and Windows platforms. Currently there is no z/OS workload agent, but CA plans to adopt the z/OS agent from CA ESP for this purpose.

**CA dSeries Workload Automation**

CA dSeries Workload Automation (formerly Cybermation ESP: dSeries) for distributed-based environments has a small number of customers to date. CA dSeries can also initiate jobs from a rich array of event sources. The CA dSeries server runs on Linux, Windows, and Unix and can manage workloads on other platforms and integrate with packaged applications using the same agent technology as CA ESP. CA reports that CA dSeries is an easy–to-use product with a small footprint. The primary scheduling servers can failover to another server in a Windows cluster. There is no job recovery processing other than that provided explicitly by applications.

**IBM Tivoli Workload Scheduler**

IBM Tivoli Workload Scheduler (TWS), based on the original Tivoli Maestro product, provides distributed-based workload management for Unix, Linux, and Windows platforms. It has the capability to manage z/OS, OS/400, and other platform workloads via agents. A TWS network contains at least one domain, the master domain, which is the management hub. Additional domains can be used to divide a network into locally managed groups. Automatic recovery is provided by domain managers using a hot standby architecture. Fault-tolerant scheduling agents carry a local schedule plan to continue workload processing in the event of network or scheduling server failure.

The TWS for Applications component integrates workload scheduling with Oracle, PeopleSoft, and SAP business applications. IBM Tivoli Dynamic Workload Broker (TDWB) provides virtualization and load balancing in distributed environments and
dynamically discovers newly provisioned resources. TDWB integrates with IBM Enterprise Workload Manager to provide more granular service-class and resource-utilization information for load balancing. IBM technology uses an open interface and can be used with other scheduling engines.

Many IBM customers use a heterogeneous workload environment. The z/OS and distributed scheduler engines can be accessed from a central job scheduling and operations console to provide end-to-end workload automation.

**ORSYP Dollar Universe**

ORSYP S.A. based in Paris, France, was established in 1986 to provide IT automation software and related consulting services. Dollar Universe is a cross-platform job scheduler with 950 customers worldwide that supports Linux, MPE, OpenVMS, OS/400, Unix, and Windows. Options are provided for integration with SAP, Oracle, PeopleSoft, JD Edwards EnterpriseOne and Microsoft Business Solutions—Axapta® (now Microsoft Dynamics™ AX). Central operations are managed via a GUI or Web client; integration is provided for HP, BMC, IBM, and CA systems management tools.

**Redwood Cronacle**

Redwood Software, founded in 1993, offers Cronacle for job scheduling. Redwood lists 3100 customers, which includes clients of their document management tools. Cronacle supports workloads on OpenVMS, OS/400, UNIX, Windows, and z/OS, and integrates with SAP, PeopleSoft, Oracle, and others. The Redwood Explorer interface provides a Windows-based central console and a Web-based client. Cronacle features failover support and dynamic load balancing. Redwood also provides an integrated version of Cronacle for SAP NetWeaver called SAP Central Job Scheduling.

**SMA OpCon/xps**

SMA (Software and Management Associates) was established in 1980 to provide job scheduling software for Unisys platforms. The company created OpCon/xps in the mid-1990’s to respond to the emerging client/server market on Windows and Unix platforms. OpCon/xps now provides cross-platform automation solutions on a broad range of mainframe, Linux, Unix, and Windows platforms.

OpCon/xps has about 130 customers, most of whom automate cross-platform applications on many systems, including traditional IBM and Unisys mainframes and distributed platforms. OpCon/xps runs on a single dedicated Windows server with a failover capability and can control workload on mainframes and other servers through agents. SMA has successfully transferred the majority of its original customer base to the new product. Many have migrated; SMA provides a Unisys agent for those who remain. A production server can be shut down for servicing without stopping jobs scheduled for other servers. The z/OS agent includes a JES2 restart capability.
**Tidal Enterprise Scheduler**

Tidal Software formed in 1979 provides products and services for cross-platform job scheduling and application performance management. Tidal Enterprise Scheduler automates workloads on and across Linux, MPE, OpenVMS, OS/400, Tandem, Unix, Windows, and z/OS. Tidal also integrates with SAP, PeopleSoft, Oracle, Informatica, JD Edwards, Tivoli Storage Manager, Veritas, Business Objects, and Lawson, and provides centralized access from a single graphical console.

Tidal Enterprise Scheduler is used by over 400 organizations, typically to automate IT processes across several diverse platforms. The scheduling server, which runs on Unix or Windows, has a high-availability failover option. Alternatively, the scheduler can use Unix or Windows clusters. Application restart must be handled by application coding conventions. On z/OS, Tidal offer an agent that manages task initiators, or a gateway option for use with JES2. Integration with Tidal’s performance management software provides the basis for critical process monitoring.

**UC4:global**

UC4 Software was formed in 1985 to provide Siemens mainframe scheduling software. In 1996, UC4 Software released their first cross-platform job scheduler, now called UC4:global. UC4:global, is used by over 750 customers worldwide, supports Bull GCOS 8, Linux, MPE, OS/400, OpenVMS, Siemens BS2000, Tandem, Unix, Windows, z/OS platforms. UC4:global supports SAP, Oracle, PeopleSoft, and Seibel applications.

UC4:global uses agents on remote platforms called executors. It has redundant scheduling servers to handle failover conditions automatically. Rollback is provided for z/OS JCL. File version management (similar to z/OS GDGs) on Windows simplifies recovery in the event that prior copies of flat files need to be restored. It includes a rich event-based architecture, calendar scheduling, and a central point of control and visual designer for cross-platform operations. Schedules can be stored in Oracle, DB2, or SQL Server.

**Vinzant Global ECS**

Vinzant Software has been providing job scheduling software since 1987 and lists almost 100 customers on their website, primarily from the U.S. Its Global ECS software incorporates a Windows-based administration client and server software. Agents are available to manage workloads on Linux, NetWare, Unix, and Windows platforms. A software development kit provides integration APIs for C, C++, and Microsoft Visual Basic®.
Analysis of job scheduling on Windows

An IT organization that is considering a strategy for moving from a mainframe to the Windows platform must understand the implications of moving its scheduling operations. This section presents comparative research concerning job scheduling solutions that can be hosted on the Windows platform to support enterprise-scale workloads.

Job schedulers employ a central server and a database; these two components govern the core scheduling capabilities. Product vendors have chosen either mainframes or distributed platforms to host the central server component and they use agents to reach other platforms. Some vendors permit a combination of both mainframe and distributed servers, with a central console. No matter what the configuration, the scheduling servers have platform-specific capabilities that affect overall job scheduling capability.

The questions that are addressed in this research focus on two areas: technology and market forces.

- **Technology.** What, if any, are the material differences in the strength and maturity of the features provided by tool vendors on the two classes of host platform, mainframe and distributed?

- **Market forces.** What, if any, are the material differences in market forces that are driving feature maturity and innovation of scheduling products on the two classes of host platform?

To answer these questions, we have researched several of the products listed in the previous section. In addition to conducting Web-based research, we have interviewed vendors and gathered detailed product and feature-level information about a selection of key mainframe-based and distributed-based products.

The products selected for detailed research are shown in Table 1. Research has been confined to their capabilities on the mainframe and Windows platforms. However, distributed-based products generally provide equivalent capabilities across Linux, Unix, and Windows platforms.

<table>
<thead>
<tr>
<th>Mainframe-based Schedulers</th>
<th>Distributed-based Schedulers</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC CONTROL-M for z/OS</td>
<td>BMC CONTROL-M for Distributed Systems</td>
</tr>
<tr>
<td>CA Unicenter CA-7 Workload Automation</td>
<td>CA Unicenter AutoSys Workload Automation</td>
</tr>
<tr>
<td>CA ESP Workload Automation for z/OS</td>
<td>CA dSeries Workload Automation</td>
</tr>
<tr>
<td>IBM Tivoli Workload Scheduler for z/OS</td>
<td>IBM Tivoli Workload Scheduler</td>
</tr>
<tr>
<td></td>
<td>SMA OpCon/xps</td>
</tr>
<tr>
<td></td>
<td>Tidal Enterprise Scheduler</td>
</tr>
<tr>
<td></td>
<td>UC4:global</td>
</tr>
</tbody>
</table>

This research was used to build an aggregate picture, so as to differentiate the capabilities that are available for mainframe-based and distributed-based operations.
respectively. This provides the basis for analysing job scheduling solutions hosted on the Windows platform using mainframe solutions as a reference point.

Note This paper does not provide guidance on product selection, nor does it provide commentary on the intrinsic value of specific features.

Technology

The feature descriptions for batch processing, workload management, and workload automation were used as the basis for collecting and aggregating the relative strength and maturity of each feature on the two platforms. The assessment is limited to scenarios where mainframe-based schedulers are used primarily to schedule mainframe workload, and Windows-based schedulers are used primarily to schedule distributed workload.

From a technical standpoint, this analysis shows that there is little to choose between mainframe-based and Windows-based job scheduling products. The following points arise from observations of consistently strong features by host platform.

Strengths of mainframe-based job schedulers

Mainframe workload that is scheduled by mainframe-based job schedulers runs under the control of JES2 or JES3, and is thus scripted using JCL. JES2 and JES3 are therefore important components of the overall mainframe job scheduling environment.

- **Service classes.** Service classes were built into JES2 and JES3 from the outset, to provide a mechanism for prioritizing batch processing based on resource requirements and job priority. Although similar mechanisms have been built into distributed-based products, service classification has been a lower priority.

- **JCL restart and recovery.** Because JCL and the JES engines support multi-step processes, they incorporate a sophisticated checkpoint and restart capability that is not found in the equivalent scripting languages that are used on distributed platforms for workload execution.

Strengths of distributed-based job schedulers

Distributed platform workload that is scheduled by distributed-based job schedulers may be scripted using a proprietary scripting language, or by platform native command shell languages. Because there is no standard scripting language, this summary makes no assumptions about scripting features.

- **Event-based automation.** Distributed-based schedulers offer a richer variety of built-in event types, including storage threshold detection, database events, file arrival/update, Web server events, user logon/logoff, mailbox events, and network events.

- **Graphical tools.** All job schedulers use Windows GUI tools or Web browsers to provide a central point of control for schedule definitions and operations. However, you can expect richer tools for workload analysis, graphical workflow definition, forecasting, and planning functions in schedulers designed for the Windows platform.

- **Scheduling Web applications.** Job schedulers on the Windows platform provide greater opportunities to integrate J2EE and Microsoft .NET workloads. Some schedulers can schedule Web services.

These relative strengths give a broad indication of the differences you may expect from schedulers running on the two classes of host platform. These differences are slight,
and variable by product; they are outweighed by the majority of features that show equivalent strength on mainframe and distributed platforms.

Market forces

It is clear that the mainframe and Windows platforms come to job scheduling from very different starting points. Batch processing was central to early mainframe operations, because these machines were designed to handle varied workloads on behalf of a large number of users who had to wait for their output. Windows was introduced as the operating system for the personal computer, where the emphasis has been on ease of use and immediate access to personal information.

Cost

Users look for ways to optimize the use of shared IT resources to minimize operational costs and to avoid costs associated with capacity upgrades. Job scheduling provides a means balance workloads across existing servers and to shift non-essential workloads away from peak on-line processing periods. Traditional batch programs have been developed for the mainframe as a way to allow work to be accumulated for overnight processing. This in turn has driven the creation of rich batch processing capabilities on mainframe platforms. On distributed platforms, cost is still a major driver, but platform costs are lower, thus reducing the pressure for 100 percent utilization at all times. The dominant concern for many organizations today is the cost of integrating and automating complex on-line systems and processes. Workload management and automation address this concern by reducing the need for costly real-time integration services and software and by simplifying operational procedures.

The Internet

The Internet is one of the key driving forces for software innovation. The Internet is “always on” so there is little need for overnight batch activity when Web, application, and database servers must be responsive to users around the globe in all time zones. Self-service Web applications create a need for rich real-time integration of back-office applications to fully automate the service that is provided to customers.

Regulatory compliance

To be compliant with new regulations, particularly in the healthcare and financial services sectors, IT organizations must ensure that key processes are fully automated, monitored, and logged for future audits. Job schedulers are fulfilling a need for coordinated cross-platform integration between servers, packaged applications, and core applications to ensure that business activities can be traced through the various IT components that support them.

Growing complexity

Systems administration to optimize resource utilization becomes more complex with the increasing use of server farms, blades, clustering, virtualization, and storage area networks, which in turn demands greater sophistication in workload management and automation features. Job scheduling can now be extended to optimizing and improving
the use of Windows Server on HP Integrity, Unisys ES7000, and Fujitsu PRIMEQUEST servers.

**Agility**
The current drive for “agile” IT means there is high demand for software that can be used to achieve simple but rapid integration across a variety of applications and platforms. Job scheduling vendors have responded to this demand by adapting batch integration techniques for use in real-time integration (event-driven scheduling). The key difference between job schedulers and application integration software for real-time integration is that the former offers a quick way to assemble a business process without the cost and effort associated with software development.

To summarize, there is a great deal of overlap in the market forces on the mainframe and distributed platforms, but these drivers also help to explain how job scheduling has evolved differently on these platforms.

The arrival and adoption of job schedulers on Windows has brought with it a rich and mature capability for more traditional batch processing and workload management. These job schedulers are modelled on or have been ported from traditional mainframe tools and thus exhibit the characteristics that are required to move large batch workloads to the Windows platform.

It is clear that the need for workload automation—real-time integration—is driving innovation in the job schedulers that run on distributed and mainframe platforms. There is less innovation in batch processing except where extremely high workload volumes require the use of multi-system computing architectures to handle high and highly variable capacity demands.
JCL on Windows

JCL, the job scripting language of IBM mainframes, is used for jobs that are to be run under the control of a scheduler or one of the JES engines. JCL provides the detailed definitions of job steps, dataset definitions, output spooling, and program execution.

Before we look in detail at JCL emulation products, it is worth considering the benefits and drawbacks of preserving JCL during a legacy modernization initiative. This will help you decide when it makes sense to use such products and when it does not.

During assessment, you may consider techniques that leave application code largely unchanged on the mainframe, such as data access and legacy extension. However, if you are considering an application rewrite, package replacement, application migration, or re-engineering options you will need to consider the implications for existing JCL. In Table 2 below, we give a broad outline of these techniques and provide some guidance on benefits, drawbacks, and implications for reusing or replacing JCL.

Table 2. Legacy Modernization Implications for JCL

<table>
<thead>
<tr>
<th>Technique</th>
<th>Description</th>
<th>Benefits</th>
<th>Drawbacks</th>
<th>JCL Implications</th>
</tr>
</thead>
</table>
| Application rewrite    | An application is replaced by a new version that is written using new programming languages, databases, and tools | • Freedom to choose new architectural standards (languages, databases, and tools) | • High cost of development  
• Hidden costs  
• High risk of delay or failure  
• Fragmented skills and tools | Existing JCL is very unlikely to be usable, so must be replaced with new Windows batch scripts |
| Package replacement    | An application suite is replaced by a commercial package that provides equivalent functionality with different operational requirements | • Application skills and ongoing investment are provided by the vendor  
• Rapid introduction of new processes | • High cost of customization  
• Little business differentiation or flexibility  
• Increased overhead from orphaned applications | Existing JCL is very unlikely to be usable, so must be replaced with new Windows batch scripts |
| Application migration  | An application is migrated with minimal change to a new platform, so as to provide near-identical operational requirements | • Lower operational costs on new platforms  
• Access to contemporary skills and technologies  
• Low business impact of change | • Application is not modernized  
• May not satisfy new business requirements  
• Potential additional cost to reintegrate with mainframe | Preserving existing JCL is very likely to be a requirement, to ensure minimal application changes and low migration costs |
### Technique | Description | Benefits | Drawbacks | JCL Implications
--- | --- | --- | --- | ---
Application re-engineering | An application is invasively converted to improve its structure, so as to comply with new architectural standards or business needs | • Greater control over costs and risks  
• Selective modernization of key assets such as SOA and databases | • Unforeseen manual effort  
• Generated code may be difficult to maintain  
• No general-purpose conversion tools | Depending on the nature of re-engineering, JCL may be not reusable – for example, DB changes may invalidate JCL

If you are not preserving the JCL, you will use the native scripting environment provided by the platform to define the details of the processes to be run. Some job schedulers provide a platform-independent scripting language.

### JCL emulation products

**Fujitsu NeoBatch and NeoSort**

Fujitsu NeoBatch and NeoSort are part of the NetCOBOL family of legacy revitalization products. Fujitsu NetCOBOL also supports mainframe CICS/COBOL applications on Windows platforms with NeoKicks, as part of their mainframe-to-Windows migration offering. The NetCOBOL family includes support for modernizing mainframe applications during their migration to the Windows platform. NeoBatch provides the option of running the JCL without changes, or of converting the JCL to JScript. By converting to JScript, users can extend their jobs to take advantage of the Microsoft .NET Framework. Otherwise, JCL and JScript versions perform identically. Early release software was first made available in September 2005. NeoBatch and NeoSort were released for general availability in April 2006.

Fujitsu cites a small number of customers who are using this technology to migrate mainframe JCL. Fujitsu is also working with a number of vendors to provide job scheduler integration.

**Micro Focus Mainframe Server**

Micro Focus sells legacy development and deployment tools. For IBM mainframe customers, Micro Focus tools provide a compatible environment for developing and testing COBOL-based online transaction and batch processing applications. Two years ago, Micro Focus added production CICS support to their distributed products (Net Express for Windows and Server Express for Unix and Linux) to support mainframe migrations. Recently, Micro Focus extended this effort with a beta program to support JCL batch processing and JCL conversion for distributed platforms. The JCL support was released in October 2006 for general availability in Micro Focus Studio for Mainframe Migrations (for development) and Micro Focus Server for Mainframe Migrations (for deployment).
Although this is a first release, the technology for JCL support has been in use by mainframe developers since Micro Focus Mainframe Express was first released in 1998. Micro Focus is currently working with a number of vendors to integrate their JCL execution engine with job schedulers.

Using a job scheduler with JCL on Windows

Table 3 shows a summary view of the level of JCL support provided by either Fujitsu Computer Systems or Micro Focus for each Windows job scheduler.

Table 3. JCL Support for Windows Job Schedulers

<table>
<thead>
<tr>
<th>Windows Job Scheduler</th>
<th>JCL Emulator Support from Fujitsu or Micro Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppWorx</td>
<td>Untested</td>
</tr>
<tr>
<td>ASCI ActiveBatch</td>
<td>Tested (basic support)</td>
</tr>
<tr>
<td>ASG-OpsCentral</td>
<td>Untested</td>
</tr>
<tr>
<td>BMC Control-M for Distributed Systems</td>
<td>Tested (basic support)</td>
</tr>
<tr>
<td>CA dSeries Workload Automation</td>
<td>Tested (advanced support)</td>
</tr>
<tr>
<td>CA Unicenter AutoSys Workload Automation</td>
<td>Tested (basic support)</td>
</tr>
<tr>
<td>IBM Tivoli Workload Scheduler</td>
<td>Untested</td>
</tr>
<tr>
<td>ORSYP Dollar Universe</td>
<td>Untested</td>
</tr>
<tr>
<td>Redwood Cronacle</td>
<td>Untested</td>
</tr>
<tr>
<td>SMA OpCon/xps</td>
<td>Untested</td>
</tr>
<tr>
<td>Tidal Enterprise Scheduler</td>
<td>Tested (basic support)</td>
</tr>
<tr>
<td>UC4:global</td>
<td>Tested (basic support)</td>
</tr>
<tr>
<td>Vinzant Global ECS</td>
<td>Untested</td>
</tr>
</tbody>
</table>

Both Fujitsu and Micro Focus provide a simple batch command-line interface that provides basic support for a job scheduler to start JCL processes and either wait for a completion code or cancel the job.

Advanced support provides additional capabilities such as variable substitution, remote execution, asynchronous operations, and the ability to start and stop service classes or the entire JCL subsystem. In some of the product combinations shown as “Tested (basic support),” support may include a small subset of these advanced capabilities. This is not shown, to keep the table simple.

JCL statement support

Fujitsu and Micro Focus both claim full syntax support for JCL statements, but execution support is not provided for IBM’s JES2 or JES3 control statements. In general, this stance provides an acceptable solution, because the JES2 and JES3 control statements are either irrelevant in the new environment, or superseded by facilities that are provided by the job scheduler.
There are four base JCL statements that are not fully supported by these two JCL emulators. These are the CNTL, ENDCNTL, OUTPUT, and XMIT statements.

The JCL execution support provided by Fujitsu and Micro Focus is equivalent to JES2. This includes support for features such as service classes, PDS naming, generation data groups, remote job entry, output spooling, and tracing. They do not provide full support for checkpoint restart or advanced features such as SMS managed data, integration with other JES engines, or workload balancing.

**Support for common utility programs**

The IBM z/OS operating system ships with a number of important data management and other utility programs that are frequently used in job steps to prepare data for processing by the application program.

The most important of these programs is the DFSORT utility, which is used to sort records in a sequential file. Both JCL emulators include IBM compatible SORT programs, and both provide equivalents to the most commonly used utility programs.
Windows PowerShell

Microsoft recognizes that task scheduling and task automation support in Windows XP, Windows Vista, and Windows Server are critical to the enterprise customer. To address this need and provide a sound foundation for third-party vendors, Microsoft has created Windows PowerShell™.


Windows PowerShell consists of an integrated command line shell and scripting language. The scripting language includes application command-line syntax to provide a more natural and consistent verb-noun style of command invocation. Windows PowerShell allows IT professionals to pipe native .NET objects between commands. It also has access to the entire .NET Framework for application and data manipulation, and supports functions of text-based variables of traditional command interpreters. Windows PowerShell’s error-handling features simplify script writing and testing; for example, the WhatIf parameter indicates what would have happened had the script been run without this parameter, and the Confirm parameter requires users to approve commands before execution.

At the same time, Microsoft has created a new Task Scheduler service for Windows Vista‡. Task Scheduler adds support for scheduling tasks in response to events that appear in the Windows event log; chaining tasks for complex operations; and specifying multiple conditions that must be satisfied for a task to be run, including events, date and time, network availability, and whether in battery mode or on AC power. Each scheduled task can be assigned the security context of an authorized user, with passwords stored in the Credentials Manager.

Windows PowerShell and Task Scheduler are two key components of a job scheduling solution. Windows PowerShell is similar to (and can be converted to) process-based languages like C# and Microsoft Visual Basic .NET. Task Scheduler is a central server component that oversees a broad range of server operations. These Windows components are planned to merge over time to combine the process-based and server-based models.

† See http://www.microsoft.com/technet/scriptcenter/hubs/msh.mspx
‡ You can find more details at http://www.microsoft.com/technet/windowsvista/mgmntops/taskschd.mspx
Conclusion

Job scheduling is a vibrant market, and job scheduling tools are likely to be found at the heart of enterprises large and small, regardless of their core platforms. Job scheduling has found a new purpose in providing a relatively simple and agile way to stitch together the myriad of applications and platforms that are required to automate new business processes in real time. Whereas early job scheduling was necessary to optimize the use of costly IT equipment, it is now used to optimize critical business processes and to manage service level agreements.

This report set out to assess the impact of moving scheduling operations from a mainframe to the Windows platform. Below are the major conclusions that can be drawn from this research.

- Organizations use job scheduling tools on Windows for real-time process integration more often than for traditional batch processing. Internet applications, regulatory compliance, and multiplatform administration are driving the need for process automation in organizations that have spread their workloads across many Windows or other distributed platforms.

- Organizations that operate heavy and complex workloads on the mainframe are more likely to operate traditional batch processing environments to optimize use of mainframe resources.

- Only 3 vendors offer mainframe-based job schedulers. These vendors compete in a stable market of 8,000 to 9,000 mainframe customers. At least 13 vendors offer Windows-based job schedulers and compete in the growing Windows Server market. There are estimated to be between 7,000 and 10,000 organizations that use distributed-based job schedulers already.

- Job scheduling technology for the Windows platform is as mature as job scheduling technology for the mainframe platform, and it is equally capable of handling enterprise-scale workloads. The principle differences arise from the use of native platform features such as JCL.

- Both mainframe- and Windows-based job schedulers have strong support for integrating workloads across many applications and platforms, including leading packaged applications. Both have sophisticated failover and checkpoint/restart mechanisms to recover from system failures. Similar features are available on both platforms for workload balancing, planning and forecasting, calendar and workflow schedules, critical process monitoring, and many other features.

- Windows-based job schedulers have stronger support for event-based automation, a greater number of graphical tools, and integrated support for new application workloads that are based on Web technologies (primarily Java, .NET, and Web services).

- There are two vendors with early-market products that support both mainframe COBOL applications and JCL migrated to the Windows platform. These products provide a basis for low-cost workload migration where preservation of existing investment is a priority. Both vendors provide the ability to integrate with third-party job schedulers to provide advanced scheduling capabilities beyond JCL.

- Microsoft has enhanced the task scheduler that is built into Windows Vista, and has added a free download called Windows PowerShell for advanced scripting. These will provide important native platform capabilities to supplement and underpin the features of future Windows-based job schedulers.

In summary, if you are considering moving workloads to the Windows platform, there is a broad choice of job scheduling tools that can take on the task of workload management and automation. These tools provide the same features and functions as their mainframe counterparts.
However, to plan this move, you should consider carefully whether you are planning to preserve your current IT batch operations and development practices unchanged, or whether your move heralds a change to embrace the new opportunities of event-based automation. If the former, then consider adopting migration tools that preserve the operational characteristics of the mainframe (such as JCL emulation). If the latter, now may be a good time to consider adopting the advanced capabilities of a job scheduler to help you handle complex workload automation.
## Appendix 1. First-Generation Job Scheduling: Batch Processing

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic restart and recovery</td>
<td>The scheduler restarts after a system failure and recovers jobs and job steps that are currently running with automatic rollback and roll-forward of transactional resources to ensure application integrity.</td>
</tr>
<tr>
<td>File management</td>
<td>Define and schedule file management jobs, which includes sorting, renaming, labelling, copying, comparing, platform conversion (code page, byte order, and format, as for VSAM to C-ISAM), and batch editing.</td>
</tr>
<tr>
<td>Integration with security systems</td>
<td>Security integration includes operator authentication, access controls, job authentication, and access controls for resources that are required by jobs, including integration with SAF, RACF, ACF2, and third-party Web security products.</td>
</tr>
<tr>
<td>Operator alerts</td>
<td>Operators and users are notified of problems or other conditions that require intervention in such a way that jobs may be manually restarted or cancelled.</td>
</tr>
<tr>
<td>Scheduler failover</td>
<td>The scheduling system manages system failures by self-replication to standby or pooled machine resources, or through redundant scheduling servers.</td>
</tr>
<tr>
<td>Service classes</td>
<td>Jobs are scheduled based on service classifications (to which hardware and software resources are assigned) to control resource contention by restricting the number of concurrent jobs in each service class.</td>
</tr>
<tr>
<td>Spooling devices</td>
<td>Spooling devices minimize scheduling delays by moving the wait time for slow devices that are used for job input and output (such as readers and printers) to a separate processing queue.</td>
</tr>
<tr>
<td>Scheduler throughput</td>
<td>The scheduler manages very high-volume workloads (ideally by scaling linearly with the raw processing capability of the execution nodes) by minimizing network and other scheduling latency.</td>
</tr>
<tr>
<td>Workload failover</td>
<td>The scheduler manages workload server failures by redirecting workload and resubmitting failed jobs to standby or pooled machine resources.</td>
</tr>
</tbody>
</table>
## Appendix 2. Second-Generation Job Scheduling: Workload Management

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-platform support</td>
<td>The job scheduler manages jobs and schedules on multiple platforms, where predecessor and successor jobs (in a dependent workflow) may run on different platforms.</td>
</tr>
<tr>
<td>Cyclical scheduling</td>
<td>Job definition includes calendar schedules to run jobs on a regular, predetermined cycle such as daily, weekly, monthly, yearly, or (for example) the second Tuesday in the month.</td>
</tr>
<tr>
<td>Deadline scheduling</td>
<td>Job definition includes calendar schedules to specify a date and time for job completion up to one year or more in the future.</td>
</tr>
<tr>
<td>Inter-dependent jobs</td>
<td>Job execution is conditional on successful completion of other jobs (predecessors and successors), which provides simple static workflow definitions that are contained within a group (or network) of job definitions.</td>
</tr>
<tr>
<td>Dynamic resource balancing</td>
<td>The scheduler determines the job mix on each processing node dynamically, to balance the use of critical hardware and software resources including servers, CPUs, memory, IO subsystems, applications and databases.</td>
</tr>
<tr>
<td>External task monitor</td>
<td>The scheduler monitors external tasks (jobs that are run under other subsystems, operator-started tasks, and packaged applications) and uses completion events to trigger dependent jobs or schedules.</td>
</tr>
<tr>
<td>Multiple calendars and time zones</td>
<td>Job schedules are based on one of several customizable calendars (different public holidays, different time zones, local business deadlines).</td>
</tr>
<tr>
<td>On-demand scheduling</td>
<td>Users and operators submit individual jobs, which will be scheduled alongside those jobs already scheduled to run, and which can act as predecessors to trigger dependent jobs.</td>
</tr>
<tr>
<td>Scheduling packaged applications</td>
<td>Jobs that run under the control of packages such as SAP, Oracle, and PeopleSoft are defined, scheduled, and monitored from the central scheduling console and integrated into cross-application workflows.</td>
</tr>
<tr>
<td>Scheduling Web applications</td>
<td>Schedule, monitor, and control Java-based or .NET-based workload or Web services with integrated security and recovery management.</td>
</tr>
<tr>
<td>Single point of control</td>
<td>An operator monitors and manages workloads that run on multiple nodes of a heterogeneous processing network from a central console.</td>
</tr>
<tr>
<td>Workload analysis</td>
<td>Numerous analysis tools are available, such as queue displays, performance displays, schedule displays, schedule trace-back, configurable reports, historical reports (for auditing), and summaries.</td>
</tr>
</tbody>
</table>
### Appendix 3. Third-Generation Job Scheduling: Workload Automation

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conditional dependencies</td>
<td>The scheduler adjusts predecessor requirements based on dynamic conditions such as program return codes, console output messages, and job termination codes (simple support for conditional branching in workflows).</td>
</tr>
<tr>
<td>Critical-process monitoring</td>
<td>A group of related jobs that contribute to a business objective (such as an SLA for payroll) is monitored, and ahead-of-time alerts are issued in the event that the objective may be missed.</td>
</tr>
<tr>
<td>Dynamic schedules</td>
<td>A feature that can automatically alter schedules, resources, and workload mix based on dynamic conditions (such as changing resource configuration to improve customer response times without intervention).</td>
</tr>
<tr>
<td>Event-based automation</td>
<td>Job scheduling may be based on predefined events including console messages, file existence, file size and content, file system status (such as disk full), operating system and network events, and user-programmed event-sensing routines.</td>
</tr>
<tr>
<td>Graphical workflow definition</td>
<td>Graphical tools define a workflow of relationships between dependent jobs in a group that includes dynamic branching on job completion based on conditional tests such as completion code, message to console, and file existence.</td>
</tr>
<tr>
<td>Mobile access</td>
<td>Facilities are provided for off-site mobile access and management of workload, to recover from an outage, capacity problems, or deadline issues.</td>
</tr>
<tr>
<td>Programmable scheduler API</td>
<td>System programmers can use a documented API or write exit routines to provide more advanced features to create and modify schedule and resource definitions, and to control schedule execution engines.</td>
</tr>
<tr>
<td>Virtualization</td>
<td>Define multiple logical scheduling systems or partitions that share a uniform hardware platform, to enable separation of monitoring, access controls, auditing, and billing for multiple clients or departments.</td>
</tr>
<tr>
<td>Workload forecasting</td>
<td>Forecast future workloads based on schedules that have already been defined, including reports on resource utilization, load balancing, time windows, and business function SLAs (such as payroll) under both normal and failure conditions.</td>
</tr>
<tr>
<td>Workload planning</td>
<td>These tools assess the impact of new workloads or workload changes on capacity and performance. (Such tools are often based on simulation.)</td>
</tr>
</tbody>
</table>