White Paper

Job Scheduling: The Right Tool for a Tough Economy

EMEA HEADQUARTERS

Tour Franklin
92042
Paris La Défense Cedex
France

+33 [0] 1 47 73 12 12

info@orsyp.com
www.orsyp.com

AMERICAS HEADQUARTERS

54 Middlesex Turnpike
Bedford, MA
01730
USA

+1 [781] 276 4600

info_usa@orsyp.com
www.orsyp.com

TABLE OF CONTENT

Executive Summary

The technology boom of the nineties is abating. Though businesses continue to invest in Information Technology, new economic conditions are forcing them to take stock of what they bought, how well it works, and what lessons they learned.

Much of the technology hype turned out to be just that. Yet much of the new technology investment made during the past decade improved business productivity and contributed to profit growth. Now business needs to separate the technological wheat from the chaff. It is time to discard what doesn't work and take what works and run with it. And whether times are good, bad or uncertain it always pays to invest in proven technologies that reduce costs and contribute to the bottom line. One such technology is the unglamorous yet highly necessary job scheduler. The right job scheduling solution reduces costs and contributes to the bottom line by improving system and business productivity.

In today's computing environment your tools need to be truly cross platform. Globalization of business, mergers and acquisitions, and a profusion of products have led to a truly heterogeneous world. A cross-platform solution means more than just linking some of the more common mainframes and Unix systems and Windows platforms. A cross-platform solution means supporting all Unix, mainframe and NT platforms and their various versions. A solution that doesn't support old OS versions and obscure legacy platforms is only a partial solution.

This White Paper addresses the issues facing IT: tight budgets, limited resources, and pressure to add to business profitability, and explains how job scheduling helps IT successfully—and cost-effectively —address all of these challenges.

1. Introduction: The IT Issues Businesses Face

Businesses are always looking to reduce costs and improve productivity. Traditionally viewed as a cost centre, IT's role has evolved to become critical to the functioning of business. It is now required that IT not simply support the business but contribute to revenue generation and profitability.

IT needs to balance the increasing costs of technology—new hardware, new software, and people to implement and administer them—with a business-minded outlook. It is up to IT to use new and old technologies to support the business and to look at strategies and tools that make existing infrastructure more efficient and help contribute to the bottom line.

As IT looks to improve efficiencies, computer operations take on a pivotal role. In the growth period of IT spending four specific areas received the bulk of investment: applications, networks, systems and middleware. All these rely upon the computer operations group to integrate and keep things working.

All too often computer operations were an afterthought in application and infrastructure acquisition and the business strategy behind this investment. Now it falls to the operations group to sort out and manage all the disparate pieces of the networked-ERPed-CRMed-SCMed enterprise.

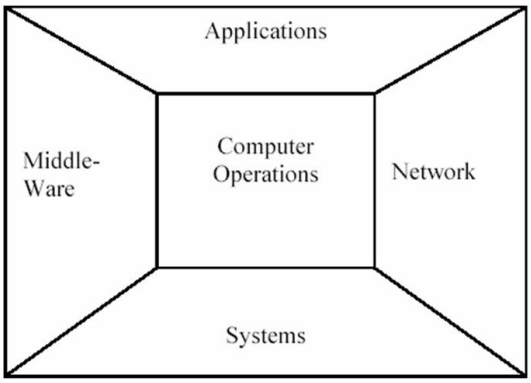

Figure 1. Computer operations is at centre of the enterprise IT growth model

With that responsibility comes renewed emphasis on integrating the operations group into the broader strategic and business planning activities of the enterprise. This means that they fall under the same business pressures that trickle down from the enterprise.

This relatively new view of IT as a business-minded entity adds complexity to the traditional IT model. Now the major issues IT faces—budget limitations, limits on personnel hiring and retention—are expanded to include generating revenue for the business. IT's bottom-line contribution comes from its traditional strength—improving technological efficiencies. And it can expand further through introducing new initiatives that help contribute to the business's competitiveness.

But while IT spending increased dramatically in the past decade IT budgets have not been transformed into blank checks. Even though IT has become integral to the success of business, budgets still limit IT departments.

1.1 Budget Pressure: Businesses Are Scrutinizing IT Budgets

IT investment boomed during the past decade, but today the rate of increase in technology spending is decidedly slowing. Gartner Group reports that the "IT budget consumes between 5 percent and 7 percent of revenue in most enterprises, and operations labour accounts for 18 percent of the budget." As businesses seek to reduce costs they will certainly take a look at IT budgets. In fact, several surveys indicate that IT budget growth will be down sharply. For example, International Data Corporation reports that IT spending will grow at about 7%, down from 11% from the previous year. If economic conditions worsen the growth rate could slow to 5%. Sources as varied as Gartner Group and Computerworld magazine report similar numbers.

1.1.1 Personnel Constraints: Talent Will Continue To Be Expensive and Difficult To Obtain

Despite anticipated slowdowns in IT spending the personnel to implement and manage new and existing systems will continue to be in short supply for the foreseeable future. According to the Bureau of Labor Statistics *Occupational Outlook Handbook* the top five fastest growing occupations over the coming decade are all computer related. The computer industry will also experience the highest wage and

salary growth of all occupational categories. According to the Information Technology Association of America (ITAA) despite a large reduction in demand for IT workers there will still be almost half a million more jobs than workers to fill them. And the ITAA reports that technical support people remain most in demand by IT and non-IT companies alike-one-fourth of all new positions over the next 12 months.

1.1.2 Technology Must Contribute To the Bottom Line

New technologies are constantly being introduced or exploited to help business improve their competitiveness and profitability. From the earth-shattering rise of the internet to the use of mundane technologies like email, businesses are implementing new technologies that make individual workers, departments and whole businesses more nimble, competitive and profitable.

New technology adoption has led to the growth of IT's role and size in business. These new technologies brought with them benefits—and some drawbacks. Benefits include reduced costs, increased productivity, quicker time-to-market, and better access to information. Drawbacks include complex software packages that do not deliver advertised benefits, increasing manpower requirements to oversee burgeoning environments, and growing instability of environments made up of a diverse array of software and hardware packages that don't always work well together.

Because IT is integral to every aspect of the business it is no longer enough to promise benefits from technology implementation—IT must produce them. And with shrinking budgets IT must still deliver even if it uses only what's at hand. Ripping out existing infrastructure and starting from scratch isn't an option, hiring an army of consultants to make everything work together is prohibitively expensive and time consuming. Fortunately there are effective solutions available that can help maximize IT efficiency and reduce costs.

2. Job Scheduling: The Right Tool

While IT vendors like to tout sexy—and expensive—solutions to enterprise IT problems there exists one basic and critical tool that can significantly reduce costs, improve IT efficiency and increase revenues. This tool is job scheduling.

2.1 What Is Job Scheduling?

Job scheduling, also known as batch processing, can be defined as follows:

- In a computer, a batch job is a program that is assigned to the computer to run without further user interaction. Examples of batch jobs in a PC are a printing request or an analysis of a Web site log. In larger commercial computers or servers, batch jobs are usually initiated by a system user. Some are defined to run automatically at a certain time.
- In some computer systems, batch jobs are said to run in the background and interactive programs run in the foreground. In general, interactive programs are given priority over batch programs, which run during the time intervals when the interactive programs are waiting for user requests.

2.2 Why Job Scheduling Is More Critical Than Ever?

Since the first programmers stood in line, punch cards in hand, at room-sized computers, computer tasks have been prioritized and scheduled. The number of computer jobs has grown apace with the growth in computing power and the number of applications. Large businesses routinely execute thousands of jobs per hour, every hour of the day, every day of the year. And as enterprises grow, these numbers keep increasing.

With the enormous increase in computer jobs it is more difficult, time consuming and costly for human beings to queue up and launch jobs manually—yet thousands of businesses do just that! Actual programmers engage in a task that can and should be automated. Highly paid professionals spend hours each day performing the extremely simple, mundane task of job scheduling while complex, important projects remain undone for lack of personnel.

In the centralized, controlled “glass house” mainframe world, robust and efficient job schedulers evolved. They were capable of handling the tightly controlled demands of businesses that used them. They leveraged all that is good about mainframes. But they also leveraged the fundamental mainframe paradigm of centralized, top-down, rigid control. And their use is limited to their proprietary environment.

As the computing world became more distributed and heterogeneous these main frame schedulers were unable to meet the demands placed on them. Because they are unable to scale and communicate outside of their environment businesses today are forced to rely more and more on patchwork job scheduling solutions that require large amounts of human intervention to make them work correctly.

Over time several solutions emerged to address the problem of job scheduling in distributed environments. While many of these solutions brought some relief to IT departments, the challenge continues. The sheer number of platforms that need to be supported, the wide array of operating systems versions still in use, and the constant change in IT environments create imposing challenges that any job scheduler must meet in order to fulfil its promise.

2.3 Automating Distributed Job Scheduling: Creating Value

With the increase in jobs in all businesses and the need to have these jobs run more quickly it makes sense and pays dividends to automate job scheduling. Automating job scheduling yields several tangible benefits:

2.3.1 Reduced Costs

As businesses grow and add computing power the number of jobs grows. As the number grows manually launching jobs becomes more time consuming and costly in terms of highly paid, skilled people performing a monotonous, repetitive task. Automating job scheduling reduces personnel costs while freeing up those human resources for more important and more profitable projects.

2.3.2 Improved Productivity

Automating job scheduling means that jobs are launched on time. And when one job ends another starts automatically. Plus, automation improves efficiency and reduces potential for human error.

2.3.3 *More efficient Use of Resources*

Job scheduling optimizes resources allowing more work to be accomplished. Existing computing and human resources are used more efficiently. A properly functioning job scheduling solution also allows new resources to be added or existing resources to be reconfigured with minimal impact on IT operations.

3. Return on Investment with a Job Scheduler

In this section ORSYP provides a simple method of calculating return on investment when implementing a job scheduler. For demonstration purposes ORSYP uses its job scheduler, Dollar Universe, for comparison and in examples. (For a detailed description of Dollar Universe, see the next section.)

Savings with a job scheduler depends on the number of managed systems and applications integrated into the solution. Savings generally occur quickly once a good job scheduler is implemented. And there are qualitative and quantitative savings to be considered.

3.1 Qualitative Savings

While an ROI calculation focuses primarily on hard dollar savings qualitative savings that accompany a job scheduler should not be overlooked. Good job scheduling implementations improve IT Operations quality in a number of ways. With Dollar Universe these include:

- Decreased errors reduce manual interventions;
- Reduced risk due to Dollar Universe's robustness and reliability (better solution for strategic applications needs);
- Improved network and system performance and reliability helps maintain service levels defined in SLA;
- Fewer maintenance interventions frees IT Operations staff to work on alternative tasks;
- Faster execution of job updates lets operations make changes more quickly.

3.2 Quantitative Savings

Job schedulers improve productivity. IT departments typically use 46% of their budget to manage their systems. This is the area where job schedulers can save budget dollars. In the specific case of Dollar Universe, concrete savings occur by:

- Reducing staff assigned to automation and monitoring tasks;
- Reducing training and upgrading costs by using the same tool throughout the managed environment;

- Reducing server/network resources consumption, due to Dollar Universe's internal technology (Event-driven engine, cooperative architecture, etc.);
- Reducing development costs associated with specific applications needs due to Dollar Universe's extended range of automation features;
- Reducing customization costs due to Dollar Universe's cross-platform, multi-systems integration.

3.3 Financial Evaluation

Direct and measurable savings come primarily from automation and monitoring features in a job scheduler. Payback areas include: job submission, job monitoring, and problem resolution. For the purposes of this exercise ORSYP will compare a totally manual environment with Dollar Universe. This evaluation takes a conservative view and is an estimate only. The following data will be used:

- Job Submission: 200 jobs per day and per person can be launched manually;
- Monitoring: 400 jobs per day, per person can be monitored manually (with Dollar Universe, 5 days/year are enough to monitor a complete production regardless of the number of jobs);
- Problem resolution: 20 aborted jobs per day, per person can be analyzed and fixed (50 with Dollar Universe due to progressive diagnostic features and real time intervention capability).

Assumptions: There is a manual operation in use today. If some aspects of automation are in place manual numbers would decrease.

In the calculations below we use the following values:

- 240 = number of operations days/year,
- 215 = the number of jobs/day,
- 5 = number of similar environments to automate,
- 3% = manual incident rate (with manual operations),
- 1% = Dollar Universe incident rate (Dollar Universe eliminates operations errors but some application errors may occur).

3.3.1 Job Submission

To determine savings in the job submission area we first calculate the man-days necessary to run the jobs manually. This would be:

(Operations days/year) x (jobs per/day) x (environments)/200 jobs/day/person = man-days

or

240 days/year x 215 jobs/day x 5 environments/200 jobs/day/person = 1290 man-days

Since using Dollar Universe would automate all these job submissions the total time devoted to this task would be saved.

Task	Manual	With Dollar Universe	Net savings
Submission	1290 man-days/year	0 man-days/year	1290 man-days/year

3.3.2 Job Monitoring

The job monitoring calculation is similar to job submission. First we calculate the man-days necessary to manually monitor the job workload. This would be:

(Operations days/year) x (jobs per/day) x (environments)/400 jobs/day/person = man-days

or

240 days/year x 215 jobs/day x 5 environments/400 jobs/day/person = 645 man-days

Using Dollar Universe requires minimal job monitoring and would only require about 5 man-days/ year. The savings are calculated below.

Task	Manual	With Dollar Universe	Net savings
Monitoring	645 man-days/year	5 man-days/year	640 man-days/year

3.3.3 Problem Resolution

When it comes to problem resolution a similar calculation is used. Only now incident rates and resolution capacity are added in. With a job scheduler such as Dollar Universe, fewer incidents occur and they are resolved faster. Therefore we calculate for the manual environment and Dollar Universe and then find the difference.

(Operations days/year) x (jobs per/day) x (environments) x (incident rate)/resolutions/day = man-days

or

3.3.3.1 Manual

240 days/year x 215 jobs/day x 5 environments x .03 incident rate/20 resolutions/day = 387 man-days

3.3.3.2 Dollar Universe

240 days/year x 215 jobs/day x 5 environments x .01 incident rate/50 resolutions/day = 51.6 man-days

Now we find the difference:

Task	Manual	With Dollar Universe	Net savings
Diagnosis/Fixing	387 man-days/year	57 man-days/year	330 man-days/year

3.3.4 Combined Savings

Now the three small tables are combined and added together. Here is the result:

Task	Manual	With Dollar Universe	Net savings
Submission	1290 man-days/year	0 man-days/year	1290 man-days/year
Monitoring	645 man-days/year	5 man-days/year	640 man-days/year
Diagnosis/Fixing	387 man-days/year	57 man-days/year	330 man-days/year
Total	2322 man-days/year	62 man-days/year	2260 man-days/year

If one man-year = 260 man-days

Then 2260 man-days/year = 8.69 man-years

We then calculate an ROI dollar value based on an assumption that an operator would cost an organization $35,000/year.

8.69 man-years x $35,000/year = $304,150 ROI in year one of a Dollar Universe implementation

In addition to these significant savings companies also benefit from indirect savings related to applications availability which translates into increased corporate capability to run its business.

Proven availability improvement is usually between 1 and 2%.

From this ROI it becomes clear that there are significant monetary benefits to be gained from implementing an automated job scheduling solution. It also becomes clear that beyond the direct

In addition to the bottom line through cost savings there are the benefits of freeing up system resources allowing them to be used more productively. And the freeing up of expensive, talented human resources that can be put to better use on more important IT projects. Taken together, automating job scheduling offers significant benefits to enterprises of every size.

Financial Services Industry Finds Upgrading Schedulers Will Pay Large Dividends

In every industry IT is under pressure to make businesses perform more efficiently while adding to the bottom line. Nowhere is this more apparent than in financial services. For example, regulatory requirements make IT integral to the continued success of the business. And batch job scheduling has become the critical component to IT success.

The U.S. Securities and Exchange Commission is asking that all stock trades be cleared on “trade day plus one” (T+1) by 2004. This requirement “will force a switch from Wall Street's traditional batch processing systems to a real-time processing network that never crashes,” according to the Computerworld article, “Bigger Than Y2K.” The article goes on to note that while upgrading to comply with T+1 will cost about $8 billion dollars the industry will realize savings of about $2.7 billion dollars a year. Not only this, but the industry will be have lower costs, lower error rates, and higher productivity while gaining the ability to handle greater transaction volume.

4. Job Schedulers: How They Work

Job scheduling is one of the most important components in a production-computing environment. Job schedulers do a number of things. They initiate and help manage long, complex jobs such as payroll runs and inventory reports. They also launch and monitor applications.

Most computer environments use some kind of job scheduler. But not all job schedulers are equal. With the advent of large distributed computing environments, very few job schedulers have scaled to meet the challenges of enterprise computing. Mainframe schedulers enjoy a reputation for power and robustness, but are limited to working on mainframes. Unix schedulers have a reputation for being severely limited in functions, but have cross-platform abilities which mainframe schedulers lack.

4.1 Manual Scheduling

When beginning to manage batch workloads in open systems environments, most companies launch their first jobs using manual methods. This is understandable and appropriate. However, this approach quickly breaks down when the number of machines and batch jobs increases.

For example, Unix and NT systems provide job launchers-*Cron* in Unix and *winAT* in NT. These tools allow users to launch jobs at specific times on specific dates. These commands provide a basis for scheduling, yet on their own do not deliver a solution for complex scheduling requirements. They rely on operators manually submitting jobs from a workstation. This approach is costly, and potentially unreliable and error prone.

4.2 Distributed Scheduling

In distributed systems Cron and winAT provide simple job launching capability. They offer the ability to start a batch job at a specific time, based upon an adequate set of time and date matching criteria. Both perform simple job scheduling tasks such as kicking off a backup every Sunday at 2 a.m., or cleaning out/tmp each night at midnight. However, they are limited.

The biggest weakness of these tools is their inability to monitor and to correlate the execution of one job with the results of another. If a backup job fails, cron/winAT doesn’t know that it should suspend the job that updates the tape catalogs or deletes yesterday's old files. If the backup finishes early, cron/winAT can't move up jobs that are to be executed upon completion of the backup.

Also, cron/winAT can only start jobs that are time-dependent. This makes it impossible to create a job that runs when a file disappears or when a system resource hits a certain threshold.

And job launching configuration files are difficult to maintain. Even minor changes to a job's start time are time consuming and error-prone. And there are no layered tools to make job creation easier. Briefly, cron/winAT are simple job launching tools designed for simple, low volume environments. They lack almost all critical features required for complex, large systems.

To make up for this deficiency many administrators create their own job management systems. They use cron/winAT to initiate a job controller and create scripts that detect

failure conditions, initiate other jobs, and provide some degree of checkpoint/restart capabilities.

While these solutions often work adequately for small job streams, they rarely scale to handle job loads of complex network environments. They also lack sophisticated user interfaces and reporting tools that allow users to keep audit trails of job streams.

More importantly, home-grown job schedulers quickly turn into full-time programming commitments. As dependence increases on the tool, more and more features are added. The result is usually a varied mix of scripts, programs, and Unix utilities that only a few people actually understand—a situation prone to problems.

4.3 Mainframe Scheduling

Mainframe job scheduling is the complete opposite of native Unix job scheduling. Mainframe tools provide robust scheduling capabilities that handle huge, complex job streams with ease. Mainframe schedulers group jobs into collections, treating the collection as a single entity whose execution, success, or failure can be tracked and used to trigger other jobs or collections of jobs. Users start jobs and job collections using time triggers or other criteria such as creation of a file, mounting a tape, or the shutdown of a database. The job scheduler is aware of almost all activity within the system and can respond accordingly.

Using screen-oriented user interfaces, system operators can track the status of jobs, noting which are running long and which are completing. Using this interface, operators can suspend jobs, delay execution, restart jobs, and track schedule slippage. It is possible to alert an operator if a job exceeds a maximum run time, or if a job failed to start due to unmet execution criteria.

Mainframe schedulers also offer good reporting tools. They create execution logs and report job failure and success. Analyzing these reports over a period of time lets users see trends, such as accounting job streams that take longer and longer or backup jobs that begin to press against the limits of backup windows.

4.4 Some Aspects of a Good Job Scheduler

Whether NT, Unix, mainframe or something else, there are fundamental capabilities that a job scheduler should contain.

A good scheduler supports non-temporal job triggers such as file creation or system alerts. Users must be able to suspend a job stream, slip a schedule to another time of day, and cancel a single instance of a job without affecting its overall schedule. There should be no limit to the number of jobs that can be created, and the system should be as easy to use with 10 jobs as it is with 10,000.

And the job scheduler should be not only a technical asset but a business asset. It should reduce costs, increase productivity, and maximize efficiency so that IT can fulfil its mission of adding value to the business.

5. Distributed Job Scheduling Methods: A Comparison

Several competing job scheduling architectures have emerged for heterogeneous, distributed environments: Cooperative, master/agent and variations on master/agent

that include master/submaster/agent and console/master/agent. Because of the similarities between master/agent and its variations and the fact that they share many of the same characteristics this paper will compare the cooperative and master/agent architectures.

5.1 Master/Agent Architecture

The traditional architecture for job scheduling solutions is the master/agent architecture. Schedulers using this model generally evolved using mainframe concepts. This architecture involves putting a full implementation of the job scheduler on one server, the "master," and putting agents on a series of other servers, the "agents."

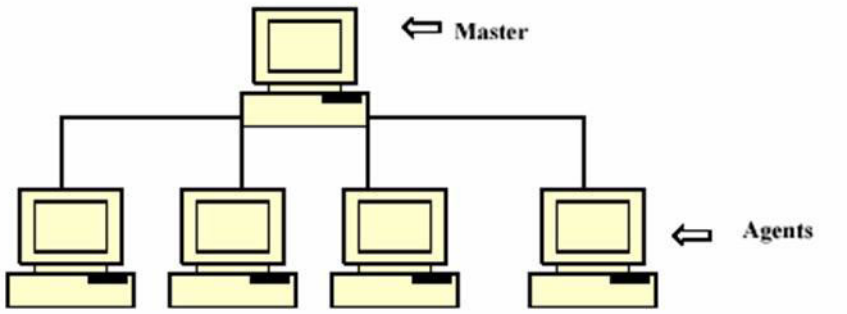

Figure 2. Master/Agent architecture presents a variety of benefits and limitations

In the master/agent configuration, jobs are set up, scheduled and administered from the master server. The actual work is done on the agents. The agents communicate with the master throughout the job run as the master passes parameters and other critical data to the agent. Jobs may be partitioned among agents. As the job is passed from server to server communication must be maintained between agents and master. This makes network availability critical to successful completion of jobs.

5.1.1 Benefits

The master/agent central administration allows tight control over jobs. But that comes at a cost-a central top-down, rather inflexible tree structure.

5.1.2 Drawbacks

5.1.2.1 Dependence on Network and Central Server Availability

The most significant limitation of master/agent systems is the requirement for the master and agents to remain in sync. When the network or central server is interrupted, how long will it take to reconstruct your activity? The well-known volatility of distributed networks is an important consideration when considering schedulers based on master/agent architecture.

5.1.2.2 Performance Degradation

A second area of concern is performance. In master/agent environments communication continually flows between the master and each of the agents. As the

workload increases the network traffic increases. As the traffic increases the potential for overload expands.

5.1.2.3 Scalability Issues

Another aspect to consider is scalability. A master can only support a limited number of agents and this depends on the number of jobs to be run. Creating a new master or “instance” creates a new and separate administration. The more instances you create, the more management you need. And when you create a new instance you need to literally recreate *all* jobs. The process can take days, weeks or even months and the process itself can lead to errors and failures at any point along the way. Plus, while the new instances can be managed by the same administrator, within reason, the inability to administer the entire job scheduling environment from a single point increases complexity, and the likelihood of confusion and errors.

This lack of scalability can impact your overall costs drastically. When you create a new master you by definition need to add new hardware at the master and agent levels. In a large enterprise this could quickly grow to a million-dollar problem.

5.2 Cooperative Architecture

Developed specifically for distributed environments Cooperative Architecture leverages the combined computing power of networks. In cooperative architecture environments a full copy of the job scheduler is implemented on every server in the network. With this approach once a server is given parameters for a job it can run independently.

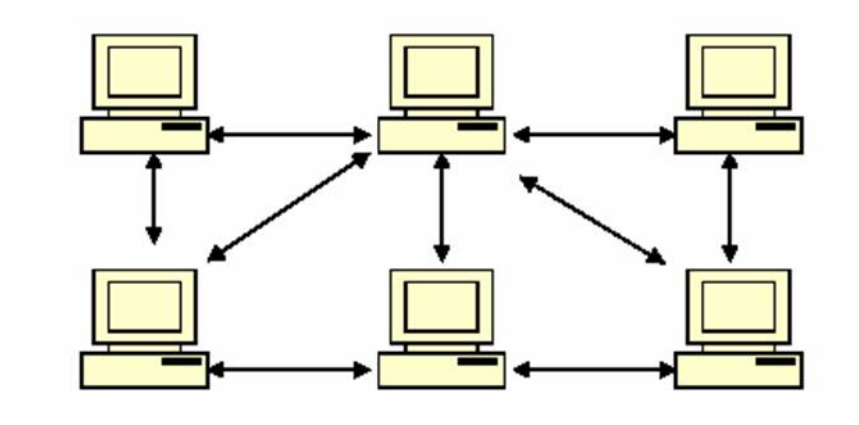

Figure 3. Cooperative Architecture improves performance and scales to the size of your network

Each server runs jobs independently of all others. Communication occurs for coordination and updates. It effectively uses network resources to combine mainframe-like robustness with distributed flexibility.

Administration in cooperative environments is flexible. You can manage your job scheduling from either a central point or at the local level.

5.2.1 Benefits

Since cooperative architecture was created for distributed environments it has many inherent benefits:

5.2.1.1 Fault Tolerance

With a full working copy of the software on every server, network downtime has diminished impact. Jobs continue to run even during network outages. The same applies to individual servers. If one server crashes, all other servers in the network continue their jobs. Any interdependent jobs are held until the crashed server resumes activity.

5.2.1.2 Performance

Since jobs run locally, network communications and overhead decrease. This translates into improved network and system performance.

5.2.1.3 Scalability

In a cooperative environment scaling is limited to the size of your network (for an example of this scalability, Dollar Universe could easily handle 1000 jobs running on 500 servers for a total of 500,000 jobs. But there literally is no limitation.). And replicating jobs is straightforward. Based on logical views of jobs and the environment even the most complex jobs can be replicated in minutes.

Another distinct advantage is more efficient use of hardware resources. Typically, in cooperative architecture your total job scheduling overhead is about 1% of CPU resources on each server in the network. In master/agent profiles you need a dedicated server for the job scheduler itself plus a back-up server in case the master fails. This is in addition to resources used on each server. And because of the limits on scalability, each time you expand to a new master configuration you need to add hardware and software for the job scheduling server.

6. Adding Value with Dollar Universe

Dollar Universe, from ORSYP, provides true cross-platform enterprise job scheduling. It gives users the tools necessary to manage complex distributed computing environments. Dollar Universe delivers unmatched availability and flexibility. And it connects to every available legacy platform.

Based on a Cooperative Architecture, Dollar Universe is designed for distributed open systems batch management. It takes advantage of distributed topology to give fault-tolerant reliability and limitless scalability.

Dollar Universe is designed to be easy to use and easy to manage. From administration interfaces for both Unix and NT to an object oriented architecture that makes updates quick and simple Dollar Universe has the features that boost productivity from individual administrators to the organization as a whole.

6.1 Cooperative Architecture: Foundation for Success

Cooperative Architecture is the foundation of Dollar Universe. The principle is straightforward.equip each server with full automation capability. With Dollar Universe each server in the system receives a full copy of the software. This insures high performance and continuous service regardless of the size or complexity of the network.

6.2 Multi Platform

In our Cooperative Architecture environment we connect to every available platform. From Unix to NT to Mainframes and everything in between Dollar Universe provides complete integration. Because of this, Dollar Universe delivers more efficient integration with less overhead on more platforms than any other scheduler.

6.3 High Performance

Cooperative Architecture spreads scheduling and launching workloads across all the nodes in the network. This effectively removes performance and execution bottlenecks found in traditional master/agent models. Cooperative Architecture increases reaction times and performance of automation functions. This is especially true in high volume operations.

6.4 Fault Tolerant

Because Dollar Universe fully automates the job scheduling process, scheduled jobs continue normal local processing even if the network is unavailable (which is crucial in distributed environments). And if a local node fails it has no impact on the overall network.

6.5 Versatile and Easy To Manage

Administration and monitoring of Dollar Universe is easy and flexible. Cooperative Architecture's client-server communications allows local and/or remote management of administration and supervisory functions. Users can choose a method that makes the most sense for their organization. This versatility lets Dollar Universe meet the requirements of the most demanding business.

6.6 Centralized Management

The user defines individual jobs, processing streams and associated scheduling data centrally and then distributes them to target machines. This is particularly useful when the configuration requires the same jobs to run in parallel on more than one machine. Users can also adapt centrally defined parameters to specific local requirements.

6.6.1 Local Management.

If local servers host different applications, users can implement job definition and scheduling locally.

6.6.2 *Blended Management*

Local operations may be monitored locally. However, users can upload local job status to a central monitoring node. This allows centralized problem resolution. A central operator can connect to a local node, diagnose problems, and take the steps necessary to recover in real time.

6.7 Management Units-Powerful Concept Defines Best of Breed

At the heart of Dollar Universe lies the Management Unit. A Management Unit is a logical execution environment. It's a concept that defines Dollar Universe as best-of-breed job scheduler for distributed and replicated environments.

Management Units lets users build and manage operations processes using a logical view of their environment without direct reference to the physical configuration. This is powerful in distributed environments where hardware configurations change frequently. And processes built this way adapt automatically to changes in the logical or physical configuration.

6.8 How Management Units Work

- Dollar Universe uses an object approach. In distributed environments objects (calendars, rules, jobs, schedules, etc.) may be common to multiple servers.
- Objects are attached to Management Units (logical execution environments) not to hardware. Management Units are classified by Type. Once an object is defined, in a single operation it can be attached to all Management Units of the same Type, regardless of how numerous. This is a particularly effective way of managing a replicated environment. N identical environments may be managed as one.
- The object-oriented approach minimizes the number of parameters that need to be defined and maintained. This speeds up implementation and changes. In addition it makes organizational and technical change easier. Users can instantly reuse previously defined objects for new Management Units.
- When users distribute Management Unit parameters across the network they can use generic addressing (e.g. send procedures and schedules to all Management Units of the type "Shop").
- Users can also use generic and hierarchical addressing to enable triggering and job synchronization (e.g. run the same procedure on all Management Units of the type "Shop," or on all the "Shops" depending on the Management Unit "Region").
- Changes in the number of shops, their physical attachment to a machine, or their organization in regions will not affect the definition of operations processes involved. Organizational changes occur in the administration tables. Operations processes adapt automatically.

6.9 Event-Driven Engine

For better operations responsiveness Dollar Universe is event-driven. This feature is available for scheduling as well as sequencing purposes:

- Dollar Universe eliminates the rigid processes of the "operations plan" (generation of the day's production timetable) in favour of "dynamic job scheduling." There is no preparation phase and no loading of an operations plan. In high volume operations, there is no dead period during which automation is not available because of scheduling computation. This means that there is no limit on number of jobs per day.
- Dollar Universe launches and sequences each job depending on real-time events (time, start or end of other jobs, resource status change, user intervention, etc.). The scheduler does not periodically scan for expected events, the actual events themselves inform the scheduler via the "event-handler" function.

The event-driven engine provides:

- "Just in time" sequencing and scheduling, optimizing the execution time of processing streams by eliminating the gaps due to cyclical processes. This results in sizeable reductions in the duration of operations;
- Best possible performances in terms of CPU overhead, due to the absence of cyclical processes and associated wasteful loops;
- Highly responsive corrective action capacity since operators' interventions are interpreted in real time.

6.10 Job Description.Flexible and Easy To Use

Dollar Universe lets users define all types of processing streams no matter how complex. The drag and drop option in the intuitive graphic interface makes job definition child's play.

An elementary job, or Uproc, is an execution procedure. It is associated with a definition of its functional constraints (dependencies, incompatibilities, physical and logical resources, etc.). Uprocs can be scheduled individually or Uprocs can be scheduled together in a sequence template called a Session. Then users can simply schedule the Session header. When this is done Uprocs within the Session are submitted according to the defined sequence. This includes automatic error processing. More importantly Session allows constraints of a more technical nature (e.g. restore database) to be expressed above the applications logic reflected at the Uproc level.

6.11 Fast, Powerful Scheduling

Dollar Universe's powerful scheduling algorithms satisfy the most demanding requirements. Moreover, Uprocs and Sessions simulation allow users to verify scheduling algorithms prior to activation.

Dollar Universe comes with a standard set of scheduling rules. Users can also create scheduling rules for special cases (e.g. 8th workday before the end of a 23-day period except for Tuesdays and Thursdays). The same rule (e.g. every workday) will adjust to the different calendars.

A launch window determines the earliest and latest launch times, day by day. Uprocs and Sessions may have multiple launches per day.

6.12 Implementation and Deployment

Dollar Universe provides a creative development and implementation environment. It offers four parallel environments, or Areas, with totally separate data and services. These four Areas enable progressive tuning of applications processes throughout their life cycle without data or logical conflicts. From development, through integration, acceptance and finally deployment into the Production Area it is easy to test and transfer processes between Areas without risking pollution of production data. Parallel Areas significantly diminish the number of errors encountered during applications deployment.

In each Area, Dollar Universe tracks management of objects by version. This makes it easy to change and restore objects.

6.13 Efficient Workload Balancing

To efficiently spread the workload over time and across configurations, Dollar Universe features a load balancing function built upon batch queue management. This regulates the number of running jobs and optimizes system resources. This function optimizes resource consumption by:

- Limiting the number of simultaneous running jobs in the queue;
- Balancing load between different machines because of a logical queue that may point to one or several physical queues resident on remote machines;
- Managing dynamic priorities within a batch queue-submission priority is automatically incremented every cycle. Jobs submitted with a low priority will eventually gain sufficient priority to be processed before other jobs with an initially higher priority. This means that all jobs will be processed despite differences in priority;
- Defining execution priorities assigned per queue at the level of the operating system itself.

6.14 Operations Monitoring

Dollar Universe supplies operators with both a graphic and a character and command mode Job Monitor. The Job Monitor gives operators a clear view of past, present and future. It lets them filter displays according to standard (job status, execution date, job name, etc.) and user defined criteria. Several levels of information facilitate operator access to data required to diagnose problems before taking steps to recover.

Conclusion

The pressure on IT to produce promised savings and efficiencies from new technologies they implement will only increase. In an era of fiscal belt tightening these pressures increase even more. While the computing world becomes more distributed, more heterogeneous, and more complex, IT departments are asked to do more with less.

This paper shows that automating job scheduling can alleviate some of these pressures while adding value to the business. And it gives you a means (the ROI calculator) to measure that value in concrete terms.

However, not all solutions are equal. In your selection of a job scheduler, choose carefully. Find one that gives you solid ROI. Also find one that meets your present requirements but has the robustness and scalability to meet future needs.

About ORSYP

ORSYP Software provides cutting-edge job scheduling and job management solutions empowering IT organizations to automate end-to-end IT and business processes centrally and with distributed intelligence. ORSYP enables IT departments to create efficiencies, reduce costs, enhance service delivery, and mitigate operational risks in today's heterogeneous, complex, and dynamic IT environments. Over the years ORSYP has helped modernize IT infrastructures with new implementations and migrations that are business-friendly and cost-effective.