Realtime publishers

Best Practices for Workload Automation in the Bimodal Era

sponsored by

ADVANCED SYSTEMS CONCEPTS, INC.

Dan Sullivan

Copyright Statement

Chapter 3: The Financial Impact of Workload Automation

As the past two chapters have illustrated, workload automation is obviously beneficial for IT professionals. But what may be less obvious is just how important it is to the bottom line. Streamlining IT operations in ways that free up IT staff, improve the consistency and reliability of operations, and reduce barriers to innovation are clear advantages workload automation brings to the IT side of business. These also have direct and indirect beneficial impact on both costs and revenue streams of a business.

This final chapter of *Best Practices for Workload Automation in the Bimodal Era* outlines the ways workload automation leads to financial benefits and provides a roadmap for implementing workload automation. Specifically, this chapter will address:

- Advantages of consolidation and coordination
- Faster builds and deployments of the workload
- Resource optimization and IT burden reduction
- Next steps and how to get started with workload automation

It is important to note that these benefits span a multitude of different industries. Whether your organization is a manufacturing business or a financial institution, a healthcare provider or a retail sales operation, workload automation can help reduce operational overhead and improve the quality of services in ways that directly affect both costs and revenues.

Advantages of Consolidation and Coordination

Consider a typical scenario in midsized and large enterprises: Multiple departments across the organization are constantly deploying new services and optimizing the use of long running applications. These deployments often require some form of management and automation. The development department may standardize on one tool, the operations team chooses another, and both teams find themselves supporting legacy custom scripts written by several former employees. This scenario can easily lead to fragmented reporting, inefficient use of software developers, and brittle operations that leave systems administrators hesitant to revise code that is only understood by an ex-employee who left the company years ago. Consolidating these tools using a centralized automation platform is an effective and efficient way to drive coordination and orchestration across the organization and eliminate these all too common inefficiencies. And more importantly, when consolidation isn't an option, automation allows organizations to coordinate with disparate applications and technologies.

When businesses consolidate and centralize workload automation, they often see several benefits:

- Improved coordination and orchestration
- Fewer skills needed to support multiple tools
- Consolidated reporting
- Reduced need for custom scripting
- Decommissioned software

Businesses that have departments and divisions deploying multiple tools can leverage the benefits of consolidation to reduce costs.

Fewer Skills Needed to Support Multiple Tools

Someone unfamiliar with IT might think that IT developers and operators all do pretty much the same thing. This misconception is understandable. Until you've worked in IT for a while, you probably wouldn't realize the wide range of diverse skills that are needed in a modern IT department. For example, enterprises might have personnel that focus on specialties such as:

- Web application design
- Mobile application development
- Database management
- Network operations
- Security and compliance
- User interface design
- Business intelligence
- Operations reporting

Now consider operations support. The staff members all take turns being on call in case a process fails or there is some other problem with scheduled operations. One night the data warehouse expert is on call and receives an alert that a job has failed. The failed job is written in Perl. Unfortunately, the data warehouse expert doesn't have much experience with Perl and did not develop that particular job. Digging into the logs, he sees several messages that give him a clue about the possible root cause but he decides he needs to actually look into the code to understand the problem.

He opens the Perl script and immediately realizes:

- Perl can be a cryptic language
- Perl can use programming patterns different from what one might find in PowerShell
- The job script makes use of specialized libraries and the purpose of some functions is unclear
- Error messages generated by the Perl script are fairly low-level errors in a library with which the developer is unfamiliar

What happens next? The developer starts digging into library documentation and searching online collaboration sites like Stack Overflow for hints and advice on deciphering the meaning of these error messages. A seasoned Perl developer, especially the one who wrote the script, could probably solve the problem in a much shorter period of time. Unfortunately, that person is not available, so a problem that could have been solved in a reasonable amount of time drags on until another developer can learn just enough Perl to debug and correct the problem.

This scenario is repeated over and over and over again in IT shops that use many different tools for automating their environments.

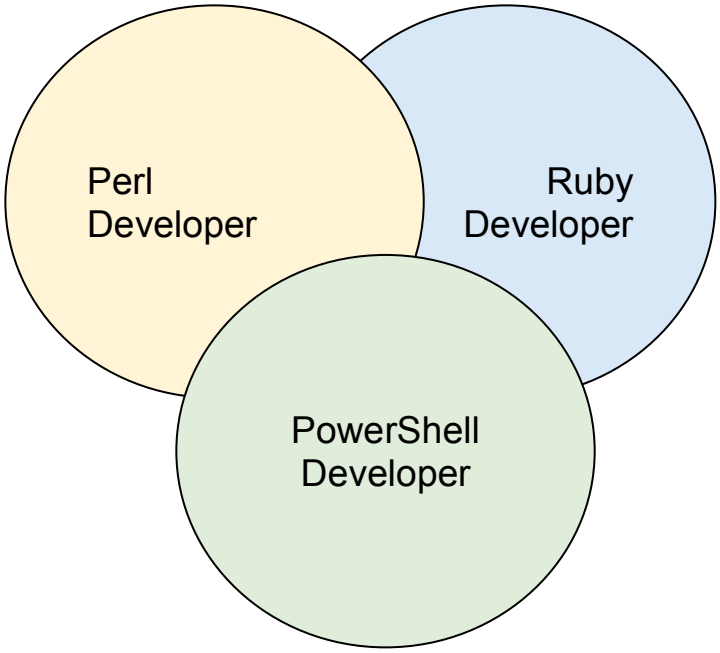

Figure 3.1: Developers have overlapping skills and can sometimes work across application frameworks but doing so often requires deep knowledge of a programming language or tool to solve complex operational problems.

Centralizing through the consolidation and coordination allows IT developers to concentrate their efforts on mastering one solution and benefitting from the ease of orchestration and coordination that comes from using one solution. Instead of having a team with diverse skills but little overlap, this setup has the advantage of a team with overlapping skills, so the business becomes less dependent on the need for a single hero or champion with each of the different tools that might be in use. This kind of arrangement is clearly a benefit to developers. They can go on vacation or work on other projects without the risk of being pulled into an operational emergency because they are the only one with sufficient specialized skills to address an operational emergency.

Consolidated Reporting

We are living in a world where human attention has become a scarce commodity. Never before in history have we had so much information at our fingertips.

Despite this availability of data, businesses are struggling with ways to derive meaningful insight from various sources. As more and more data becomes available for analysis, IT is dealing with an abundance of applications, databases, and systems to manage, in addition to the data dependencies between these systems.

Consolidated reporting is the source of another set of financial benefits. A consolidated reporting system reduces the overhead associated with collecting data for operations, management, and compliance reporting.

The Cost of Fractured Workload Environments

The key to this benefit is that workload administrators do not have to work with multiple data sources to get a comprehensive view of operations. Consider a fractured workload automation environment; you might find that:

- Some custom scripts write logs to the server on which the workload job is running
- Other custom scripts write to a consolidated logging system on the network
- Each custom script has its own pattern for logging different levels of information
- Log messages have a variety of structures that each requires custom code to parse

Now imagine a manager has asked for a status report that shows the results of a logical unit of business work, such as fulfilling an order from the time it is entered into a web application to the time it is shipped and delivered to the customer. One piece of the process writes data to a custom log format from the web application to indicate the status of the original order. The inventory system logs data about changes in inventory levels with respect to the order. The shipping workflow interacts with a third-party service to schedule delivery using the lowest cost carrier. Each of these pieces is written in a different programming language using a different log format.

Overhead Introduced by Fractured Workload Environments

Executives and managers do not want to work their way through separate log reports trying to align pieces of information across log files as if it were a business process jigsaw puzzle. They need a consolidated approach that captures the information across a business process and presents it in a comprehensive report designed to highlight key business operations. Anything that impedes this type of reporting imposes a drag on operations and slows the ability to deliver services and products to customers.

A centralized workload automation tool provides a single source of reporting on business processes. It can also provide a standard set of reports out of the box. This functionality further reduces the demands on IT staff to write reports and associated reporting infrastructure, such as tools to automate report generation, track recipients of reports, and handle distribution of reports. The net result is less time spent by IT professionals on workload automation tasks, allowing them to focus their efforts on higher priority business needs.

Reduced Custom Scripting

Developers and systems administrators can develop a love-hate relationship with scripting. They can love the ability to write code that does exactly what they want. They also appreciate the ability to use the programming language of their choice. In addition, if something goes wrong, they understand the code they have written and can reasonably expect to solve the problem quickly. Unfortunately, scripting comes with a lot of downsides for both individuals and organizations.

Continually Evolving Requirements

Writing custom scripts takes time. It is not uncommon to start with a minimal set of requirements before writing a script. The script can often meet the first and most important needs of a business user. The problem is that business is not static, and neither are requirements. A developer might feel like a script is done and move onto another task when a business user then adds to the original request or the process changes, necessitating an update to the script. Now the script becomes another item on the developer's to-do list.

If such were the case with just one or even two scripts, it would probably be manageable. That is not the norm. Most organizations have hundreds of scripts. These custom scripts grow in number and complexity. If the person that originally wrote a script is not available, someone else might have to modify the script. The second developer may not follow the same coding style or use the same design patterns. After several iterations of different developers adding new code, sometimes on tight deadlines, the script starts to lose its integrity. It becomes more of a patchwork of different interwoven modules than a logically structured script.

The Cost of Ongoing Maintenance

This situation leads to the next problem with custom scripts: they become difficult to maintain. We can think of this as somewhat analogous to the law of entropy in physics, which states that over time, a system becomes more disordered. Custom scripts become more complex as new features are added, often with an eye to finishing quickly instead of following design patterns or maintaining structural integrity.

To fully understand the cost of custom scripting, we need to consider:

- Developer time needed to write the script
- Business users' time working around the limitations of minimal scripts
- Business users' time reporting bugs and feature requests
- Developer time to debug and add new features
- The context switch costs to developers who have to jump from working on one script to another or, worse, from different types of systems administration to scripting

Centralized workload automation helps avoid these issues. Workload automation tools provide a rich library of production-ready Job Steps with built-in logic. This allows for faster workflow creation as developers can simply drag and drop Job Steps, creating dynamic workflows that are easily auditable and flexible to changing business needs. As a result, developers can reuse, rather than rewrite, and reduce the amount of time spent on researching, designing, testing, and modifying scripts.

Design patterns are more obvious and easier to follow in well-designed workload automation tools than in "quick and dirty" scripts we sometimes need to write on the job. In the long run, reducing dependence on custom scripting will reduce development and maintenance costs while improving the quality of workload automation operations and reporting.

Decommissioning Unnecessary Software

Every time we introduce a new piece of software to our organization, we introduce additional overhead in terms of:

- Managing licenses for the software
- Patching and maintaining the software
- Training staff on the software

Managing Licenses for Software

Do you know how many different automation tools are currently in use in your organization? Do you know how many software licenses are available for those tools? Are you using all of your licenses? Could you actually be unknowingly exceeding your number of licenses?

These questions arise with any business software. Tracking licenses and optimizing the use of licenses can be challenging. Different vendors have different licensing models. Some may have per-user licenses while others have server-based licensing. Add the additional considerations of running multiple virtual machines on a single server and the move toward utility computing in the cloud and you begin to see the hidden overhead of using multiple software tools for the same task.

Patching and Maintaining Workload Software

All software must be maintained. Even code we do not write ourselves still has some level of maintenance overhead. This requirement usually means we set up tasks to check for updates and apply those updates on a regular basis. If we are using a piece of software throughout the organization, it is almost certainly worth the effort to standardize and automate the patching process. When departments use specialized tools, such as different automation tools, there may be a tendency to rely on the application administrator to remember to manually update software on a regular schedule. Needless to say, the latter strategy is prone to human error that can lead to unpatched software.

Training Staff on Software

Another problem with multiple applications is that they require multiple people to be trained to use and maintain them. This problem is the same we have identified in previous chapters. IT staff have limited time to develop skills and perform their jobs. The more software we add to the enterprise portfolio, the more stress we put on IT personnel to keep that software running efficiently. This, in turn, requires them to invest time to learn the new software and understand how it integrates with existing applications, programs, and devices.

One of the best ways to reduce the burden around managing software licenses, patching and maintaining software, and training IT staff is to decommission unnecessary software. Why should your organization deploy several software packages when one would suffice?

It is true that standardizing on one tool will require some migration effort. Scripts written for decommissioned tools will need to be reimplemented in the consolidated workload automation platform, but that is a one-time cost. The long-term benefits of a consolidated approach will likely outweigh the short-term costs of reimplementing workload scripts.

There are clear advantages of consolidating and centralizing workload automation:

- Reduced need for diverse and difficult-to-maintain skills for different workload automation tools
- Consolidated reporting
- Reduced dependence on custom scripting
- The ability to decommission unnecessary software

The benefits do not end with these obvious personnel and financial benefits. There are additional benefits that stem from increased ability to deploy workload automation services more quickly and reliably to your organization.

Faster Builds and Deployments

A key advantage of using an established workload automation solution is that automation vendors invest heavily in making their solutions competitive with each other as well as with custom scripting. When evaluating workload automation tools, consider four key features that enable IT personnel to develop and deploy workload automation scripts faster:

- Libraries of tested code that allow for reuse, rather than rewriting
- Workflow application programming interfaces (APIs)
- Self-documenting operations
- Visual scripting tools

Each of these features can contribute significantly to the ability to build and maintain an array of workload automation processes throughout your organization.

Libraries of Tested Code

The wheel has been invented; there is no need to reinvent it. The same can be said for many workload automation functions. Although each business and IT process will have its own particular requirements, they typically share a number of common types of operations:

- Authenticating to services, such as databases and operating systems
- Transferring files
- Performing file integrity checks, such as checksums
- Calling Web services
- Writing process data to log files
- Sending notifications through email, SMS, and other channels

Workload automation platforms provide libraries of tested code to perform these kinds of operations and more. They are especially important for saving developer time. Not only do in-house developers not need to write these scripts from scratch, they do not have to maintain them. Developers can simply reuse these libraries of tested code, instead of rewriting scripts over and over again. Also, workload automation tool vendors can invest time to research the best ways to implement a function, such as authenticating users and processes. This can lead to a more efficient and secure implementation than might be possible if an in-house developer is under time constraints to get a functioning custom script done quickly.

Workflow APIs

Workload automation solutions that support API accessibility promote more connected, flexible environments. With API accessibility, organizations can incorporate methods and properties of any technology or application into the automation solution's content library. As a result, organizations are virtually unlimited in scope: any technology with an API can be easily incorporated into workflows, allowing developers to reuse rather than rewrite scripts using pre-built Job Steps in the content library. As more organizations adopt service-oriented architectures and the use of microservices, workload automation will fit with that design model if APIs are readily available.

Self-Documenting Operations

Documentation is a necessary and valuable part of any software. Good documentation is not always easy to come by. A developer on a tight schedule can get a script up and running without writing documentation, but she cannot avoid writing necessary code. If forced to choose between writing code to implement a feature and writing documentation describing how the implementation works, most developers will choose the former.

Workload automation systems can help mitigate this problem by employing self-documenting operations. This function can be employed through the implementation of libraries, modules, and functions that use naming conventions for functions and parameters that make clear the purpose of the function.

Visual Scripting Tools

Visual scripting tools are another important feature of a workload automation tool (see Figure 3.2).

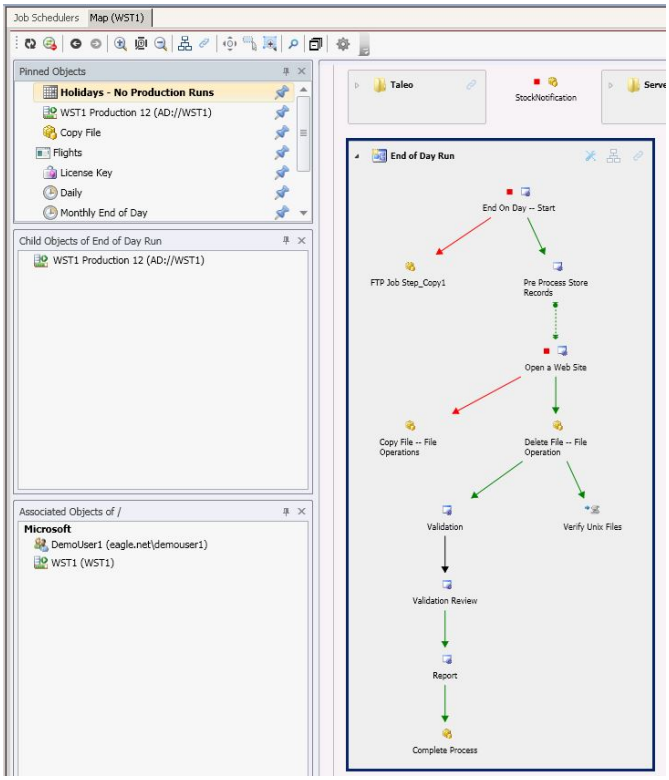

Figure 3.2: Visual scripting tools help increase the speed in which workload scripts are created.

Visual scripting tools can help speed development by supporting drag-and-drop access to libraries and APIs, context-sensitive highlighting, and visual display of information that can help developers debug workload processes.

Resource Optimization and IT Burden Reduction

The widespread adoption of cloud computing is an important factor in resource optimization and IT cost control. One of the advantages of shifting workloads to public clouds is that it allows for just-in-time provisioning of resources. Rather than purchasing servers to run a specific set of workloads on-premise, businesses can rent virtual hardware on an as-needed basis.

This approach helps reduce capital expenditures because companies do not have to purchase servers. It can also help improve efficient use of resources because companies do not have idle servers on-premise. Cloud resources can be simply shut down when they are not needed, enabling companies to avoid paying for unneeded compute resources.

Combining the advantages of on-demand computing with event-based scheduling of workloads can further improve the overall efficiency of workload operations. For example, event-based scheduling allows developers to trigger the start of a workload process after another completes. This setup minimizes unnecessary overhead associated with coordinating workload processes. For example, with event-based scheduling, developers can avoid:

- Running short jobs to check whether another job has finished and resubmitting themselves to keep checking manually
- Starting a complex workload process but leaving it idle while waiting for the preceding job to finish
- Having a human operator monitor processes and manually start workload processes as needed

Modern workload automation tools allow organizations to employ resources only when all job dependencies are satisfied. They also leverage advantages of cloud computing and the associated "pay as you go" model that is typical with cloud computing.

In addition to these obvious advantages, consolidated workloads provide for centralized management and reporting. This functionality can help provide insights into operations and allow systems administrators and application developers to find ways to improve overall performance and optimize multiple workload processes (see Figure 3.3).

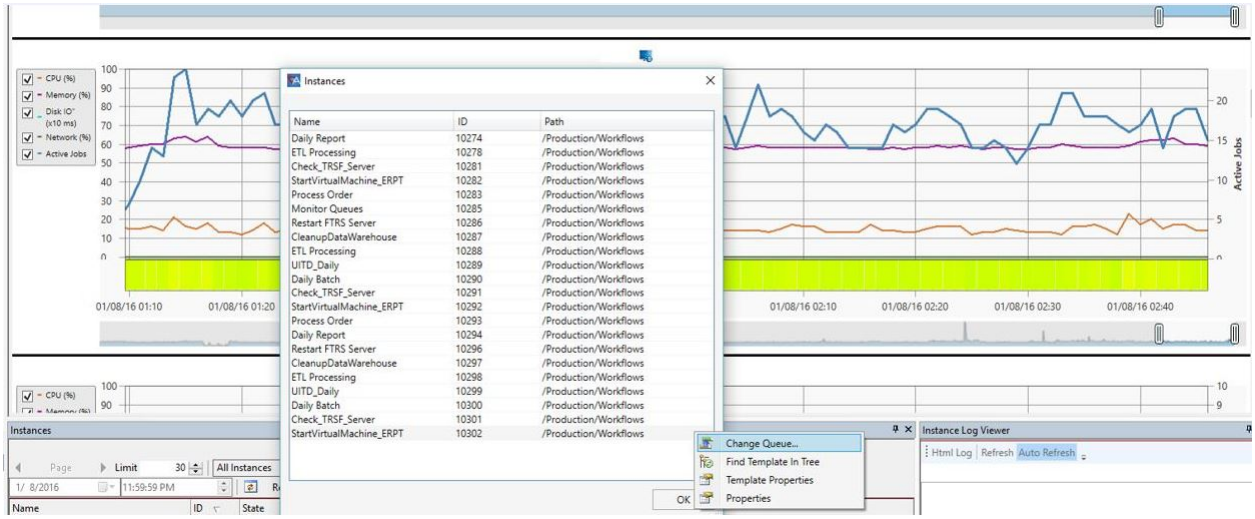

Figure 3.3: Consolidated workload automation reporting tools can help systems administrators and application administrators optimize performance of multiple workload processes.

To summarize, centralized and consolidated workload automation systems reduce IT burden in three specific ways:

- Reducing operational costs
- Improving resource utilization
- Reducing labor costs

The majority of this chapter has focused on the financial benefits of consolidated workload automation. Assuming the discussion here has led you to believe a consolidated workload automation tool may be beneficial for your organization, the question that follows is, how does my company get started?

Next Steps Checklist

The best way to begin the process of deploying a workload automation tool is to begin with an inventory of existing workload processes and tools and move from there in a logical manner to migrating existing processes to the new platform. The following list provides the specific steps to help you realize the benefits of consolidated workload automation solutions:

1. Assess current set of workflow automation tools
2. Inventory custom scripts
3. Assess available skills, needed skills, and skills gap
4. Identify top workflow candidates for event-driven automation
5. Plan for architecture-based automation by replacing silos of automation with a single enterprise workflow platform
6. Create a project schedule for migrating existing workflows to a centralized and consolidated platform

The first step is understanding what workload tools are in place already. This step is important for two reasons. First, some of the most important workload scripts in your operations may be run through these platforms. At some point in the history of your organization, a decision was made to deploy a workload automation tool. That was probably motivated by the need for a common tool to implement multiple workloads.

It is also important to understand licensing costs around existing workloads. You may be able to realize substantial savings by retiring multiple workload automation tools. That kind of cost savings could significantly bolster the business case to implement a consolidated workload automation tool.

Be sure to inventory custom scripts as well. Doing so can be more difficult because there may not be a central, up-to-date repository of information about custom scripts. A two-pronged approach is suggested. First, work with application owners to create a consolidated list of known workload scripts that are running to support ongoing operations. Second, use scripts to scan servers for cron jobs or other batch-scheduled jobs running on each server. In an ideal world, the list of workload scripts known by application owners and systems administrators will match the list generated by the server scan. In practice, there will likely be differences.

In addition to creating an inventory of workload scripts and tools, be sure to inventory the skills available in your staff. Also consider what you expect to need in terms of skills and assess the skills gap, if any. Plan training, if needed, to align with the deployment of the new workload automation system.

This point in the process is a good time to think about the architecture of your future workload automation tool. Think in terms of a consolidated infrastructure. Here is a chance to eliminate silos of workload automation and custom scripts.

Finally, develop a project plan for migrating existing scripts to the new platform. Consider the risk associated with any disruption in service. It often makes sense to migrate low-risk, low-complexity workloads first. This approach allows your staff to ensure the platform is installed and configured correctly without risking a high-impact disruption.

Summary

Centralized workload automation through consolidation and coordination can bring substantial benefits to an organization. These stem from several factors, including the benefits of consolidating and centralizing, improved speeds for building and deploying the workload, optimization of resources, and IT burden reduction. As you plan to deploy a workload automation tool, be sure to consider the next steps outline provided in this chapter to help ensure a smooth and efficient transition to your new consolidated workload automation platform.