Business & IT Analysis

quocirca

Business Report
for IBM

Provisioning and Orchestration

a report commissioned by

IBM

April 04

Created: April 04

Version: 1.0
Final

Author: Jon Collins

1 Executive Summary

Two aspects of delivery of IT on demand are the capability to provision functionality, for example to turn on specific application features or storage capacity for a given user, and to orchestrate, that is to ensure the IT infrastructure is working optimally to deliver that functionality. There are a number of reasons why it is worth dealing with these two capabilities in the round, not least that they both work together to assure a certain, predictable service level to the user.

This paper is to give CIOs, IT managers and decision makers insights into employing provisioning and orchestration technologies in the enterprise IT infrastructure. These technologies are part and parcel of the on-demand infrastructure, by which we mean an IT architecture organised to deliver applications and functionality in a way that best meets the needs of the business.

This paper presents an overview of provisioning and orchestration, describing what underlies each capability, how they relate to management of enterprise IT and what is required elsewhere in the architecture to make them work. While many organisations may be ready to deploy such capabilities, many others will need to organise themselves, both in terms of business readiness and the readiness of the IT architecture. This paper also presents a number of success factors, to provide a starting point for businesses that are looking to deploy provisioning and orchestration.

Finally, this paper describes where provisioning and orchestration are going in the future.

2 Contents

3 Why Provisioning and Orchestration?

The concepts of provisioning and orchestration are nothing new in computing. Indeed, nor should they be, as what we want to achieve with information technology should not be that complicated. Before looking at where these two concepts have come from, it is perhaps worth defining them in the simplest of terms.

Provisioning is a term for enabling access to a clearly defined service.
This may be as straightforward as allocating storage ("ten terabytes, please") or network bandwidth – most home users are "provisioned" with a certain connection to their ISP. It may be more complex – access to applications, or even specific functionality within an application, can be provisioned. The flipside of provisioning is costing, that is to associate a provisioned service with how much it costs, and to be able to charge accordingly, if necessary. This is as true inside the enterprise as it is when a service is bought in from outside.

Orchestration is the art of using computer resources as efficiently as possible.
As with provisioning, resources include storage, processing and bandwidth, but also application functionality. Ideally orchestration should take place automatically, but there is a place for manual reallocation of resources. There are overlaps between orchestration and provisioning: indeed, it could be argued that provisioning is a primitive form of orchestration.

Given these informal definitions, it should be obvious that provisioning and orchestration have been around since computers were first foisted onto an unsuspecting? world. The very first operating systems included orchestration features, though they were not referred to in that way, at the time. Instead, time slicing, swapping and paging were invented largely to maximise the effectiveness of the then-woefully inadequate computing power available. Meanwhile, as it was inconceivable that any but the largest companies would afford computers of their own, provisioning mechanisms (using the term "time sharing") were devised to allocate and charge computer access between different businesses. Computers were run by computer bureaus, set up in the late sixties and early seventies, furnishing computer facilities and charging for access time alone. Initially, there was no fixed storage, nor were there any applications: companies would run their own programs on their own data, each stored as a series of punched cards or reels of magnetic tape.

This model operated successfully until computers became cheap enough for companies to buy for themselves. Initially, it seemed obvious to enterprises that in-house computers offered a more cost effective model than buying in computer power. This was not, however, a given; it was further complicated by the arrival of personal computers and workstations, which could be networked together to offer what was ultimately global access to processing power. This story is well told; to this day, however, nobody has been able to say conclusively that business productivity has increased significantly because of computers.

By breaking the barriers of computing, the complexity of the distributed computing model made it impossible to offer provisioned access to computers; it also meant that orchestration technologies, however improved by new developments in multiprocessing technology, were woefully inadequate to cope. This gave us the situation we have today: most organisations have computers of all shapes and sizes, all with different capabilities and different costs, and all running different software. Project-based deployments over the years mean that the payroll system may still be running on a creaking old mainframe, while user files are stored on a Unix server and email is delivered by a state of the art, rack-mounted system. The ultimate issue is one of waste: all those partially loaded servers offering standalone, siloed applications; all those state of the art desktops offering email clients, word processing and Web browsing software; all those half-full disk drives. The waste implies inefficiency; unused processing implies that money has been used that could ultimately have been spent elsewhere.

Until a couple of years ago, waste was less of an issue in IT. Acronyms like Total Cost of Ownership (TCO) were largely a weapon of the big-iron suppliers against the insidious onslaught of Wintel, but large IT budgets, allocated with only tenuous guarantees of return, were the norm. When the dot-com bubble burst however, such waste was seen as no longer acceptable. Today, the IT market is getting back on its feet again, but it is a different market. The main difference is one of justification – of course, IT purchases have always required to be justified, but not to the extent they are today. As such, TCO has been moved to the backburner and replaced by Return On Investment (ROI) calculations. However, while some IT deployments can be justified in financial cost/benefit terms, many require a broader view of the risks and rewards – the Total Value Proposition. This is true in particular for provisioning and orchestration, which have many benefits (such as security, policy compliance, IT responsiveness) that are difficult to express in financial terms. These benefits are discussed later in this paper: given the increasing need for justification, it is no coincidence that provisioning and orchestration capabilities have risen to the fore, and are being addressed in some way by the majority of major IT vendors.

Over the past thirty years, technology has changed dramatically but our basic needs have not. To maximise the efficient use of IT, we need some way to go back to basics and deploy provisioning and orchestration capabilities, while recognising that we are in the midst of the highly complex, extremely demanding landscape that we are faced with today.

Let's look at them one by one.

4 Provisioning Today

At the most fundamental level, organisations require to allocate IT resources to users, projects and departments. By "resources" we mean a number of provisioning dimensions:

- processing time, measured as CPU instructions
- storage, measured in bytes
- network bandwidth, measured in bits per second.

In addition, there are the application functions, each of which involves software making use of the above dimensions. It is only possible to provision a commodity however, so all such resources must be commoditised and delivered according to three criteria:

- WHAT we are provisioning
- HOW we are provisioning it
- HOW MUCH is it costing

This leads to the first problem faced by provisioning, indeed, the main reason we stopped being able to do it in the first place: the complexity of enterprise IT environments. In most organisations, IT is a morass, added to and modified over the decades until the underlying structure is barely recognisable – indeed, if there is a structure at all. It is impossible to provision a mess.

We therefore need strategies to commoditise the mess. Assuming there is no green field site, there are two options, consolidation and virtualisation.

- **Consolidation** is essentially replacing a disparate set of IT systems (servers, storage or network devices) with a new hardware that can be partitioned to do all the old jobs just as effectively. For many organisations, this is not an option – if migration was so easy, they would have moved away from the old platforms years ago.

- The second alternative, **virtualisation**, involves adding a management layer onto the assets to be provisioned, so they can be managed as a single unit. Consolidated assets can also be virtualised, so that a single mainframe computer (for example) can be seen as a number of virtual server systems, each running its own operating system and applications.

There is a place for both approaches. In some cases, it may be possible to replace a number of servers or storage subsystems with a single box that can do the job as effectively. A virtual layer can then be applied over the top of both consolidated and other IT assets, enabling the whole to be managed.

Once a provisionable architecture exists, it is possible to serve out resources from it. In the simplest case, this involves an IT operations staff member using a provisioning software package to allocate said resources, such as reservation of storage, allocation of network bandwidth and priority and allocation of application functionality. Provisioning can occur when an individual joins the company or leaves the company, or any point in between. At this point services can be allowed or disabled. It could even be something as simple as whether a user has access to the Colour Printer.

Things can become more complicated than that. For example, there may be provisioning policies in place that need to be applied. Also, and as mentioned, if we want to provision effectively, we need to know what we have and how much it is costing, and we need some mechanism to allocate it to the people and applications that need it. We may require to provision certain functionality to different users by way of a portal, which begs the question – what is an application anyway, but a set of functions made available for a given user to perform his daily activities?

Finally, and most importantly, provisioning is not just about allocation, but also de-allocation and re-allocation. We want to be able to turn something off, so freeing up the resources being utilised for other functionality to use. This is particularly useful in circumstances where application workloads are cyclical, such as payroll, finance, ordering and so on. Software provisioning can also be used for security reasons, when linked in with identity management of users or by treating software patches and security updates as application components to be rolled out. It may be that given application functionality should be made accessible or not, depending on user roles and security considerations.

A number of computer structures are structured to be appropriate to provisioning. In servers, there is the blade architecture, in which multiple servers plug into a single chassis. Secondly, mainframe hardware has long been designed to be sliced and diced. Third, there are highly multiprocessing computers. These architectures enable certain service guarantees – for example, there is minimal risk of downtime. Rack mounted storage systems are also provisioning-friendly. Considering the cost implications and to make best use of provisioning, organisations need to have servers and storage of different kinds and different quality levels. Different users, for example, may be satisfied with different types and levels of service – some users may be satisfied with a daily backup, while others require data guarantees up to the second. Each has an associated cost, so again by building costing into the equation, users gain this visibility and can buy in capabilities accordingly.

Certain functionality is system-specific, for example data security, accessibility or backup/restore. Provisioning can be manually controlled up to a point, but there is a point where automation comes in. This can involve the scripted provisioning of resources for a new user, or the reallocation of short-term resource to a project. As intelligence is built into

provisioning however, and automation becomes more dynamic, it becomes more complex than provisioning itself. When this line is crossed, we call this orchestration.

5 Orchestration Today

If provisioning is about resource allocation, orchestration is about resource optimisation. This happens at execution time: while provisioning enables you to decide who uses what, it is up to orchestration to make sure that IT resources are used optimally. To work properly, orchestration software needs to know what IT assets exist, and the IT assets need to be "orchestratable" – that is, it must be possible to allocate jobs to them in real time. Orchestration also needs to have some idea of the capabilities of its assets – for example available storage capacity, or server workload capabilities.

Orchestration might be about managing IT resources, but to work best, it needs to take account of the needs of its "customers", that is the applications, the people, and the business processes it supports. It can do this by means of defined policies, for example allowing for overrides and time of day re-allocation, which enable resources to be prioritised according to different uses. Therefore, orchestration works on classes of service so a more dynamic spread can be given. The better the understanding, the better that orchestration can work. Therefore orchestration software should be able to work with thin client and application delivery software, web servicing and cluster management software.

Orchestration needs a clear picture of the loading of the systems, so that it can ensure there is sufficient processing capacity to handle peak workloads. It therefore needs to gather information about what is available, and how much it is used. It can then predict resource requirements and allocate resources effectively. It also works best when linked into IT performance monitoring software.

Although provisioning can take place manually, orchestration can only really happen automatically. The smaller the granularity of the tasks to be orchestrated, the more efficient use can be made of resources. However, the management overhead increases therefore a balance needs to be struck. This is something that Grid computing incorporates in the form of "marshalling".

The term Information Lifecycle Management has been coined to describe storage orchestration – what this means is ensuring that the type of storage used is applicable to the needs of the information. For example, recently accessed information about key customers can be placed on the faster, more accessible disks to ensure it can be accessed efficiently. Such disks are expensive however, and data that is not so relevant can be shifted onto slower, cheaper disks.

Orchestration can also encompass reliability issues – a well-orchestrated server farm is resilient against failure of a server. Like with provisioning, while a consolidated infrastructure isn't essential for orchestration, it certainly helps.

6 Provisioning and Orchestration Success

Provisioning is upward facing, being about delivery of the service to a customer, whereas orchestration is downward facing, ensuring that the platform is doing everything it can to maximise the benefits and minimise the costs. If done right, provisioning and orchestration can work together to enable IT infrastructure and applications to be used more efficiently. We can have provisioning without orchestration, as long as we are happy to provide only a dubious level of efficiency. We can have orchestration without provisioning, but this suggests

the totalitarian state view of IT, where everyone has the same needs – again, in the real world, it is efficiency that suffers.

Provisioning should go back to back with the orchestrated delivery of service, to enable IT delivery in the most efficient, cost effective way. Efficiency benefits and effectiveness map directly onto the removal of waste and the delivery of the right resources: by delivering provisioning with orchestration, we get the benefits of both, for example:

Efficiency benefits:

- Better allocation and utilisation of existing and new IT resources
- More predictive management, enabling future workloads to be anticipated
- Reduction of certain security risks
- Better, more reliable guarantees of service quality

Effectiveness benefits:

- Project-based management rather than fire fighting
- A stronger starting point for IT contract negotiation
- Greater visibility on, and allocation of costs
- New financing models. Provisioning and orchestration open the door to models such as “pay as you grow”

These benefits do not come out of the box however. While provisioning and orchestration software can help, it is not a silver bullet. Here are some of the things you need to think about when deploying provisioning and orchestration software, to maximise its potential.

- Hardware and software need to be compatible. This should not be a problem for off the shelf environments.
- Policies need to be defined and implemented for the allocation and deallocation of IT resources
- Operational Management processes need to be in place, across the board. For example, one of the biggest security weaknesses in many organisations is the inability of the HR department to tell the IT department when somebody leaves.
- Provisioning and orchestration work best within a comprehensive framework of management tools. This implies some kind of integrated, centralised or distributed control approach.
- Standards need to be applied. Use best practice procedures for the operational aspects of provisioning across servers, operating systems, middleware, applications and network devices. And storage!
- This also requires usability – needs to be provisioned in a way that it is appropriate to be used. At the application level, this corresponds to usability, which is a tenet of user interface design, and doesn't apply down the stack (is this true? Standards?)
- There needs to be changes in the way applications are deployed. In the past, near-complete projects have been given to the ops guys.

Deployment of provisioning and orchestration is a two-stage process. First, the organization needs to be able to roll out the product across its enterprise. Second, there is user provisioning – how can functionality be switched on or switched off for a given user.

Provisioning of the basic resources first, then provisioning of application functionality. The customer is the line of business, and this suggests a sea change in how they use IT. There is also nothing to stop provisioning to business partners, and certainly orchestration should have a positive effect on all users of IT.

7 Provisioning and Orchestration Futures

To be able to deliver IT resources as and when we need than, and to have the IT orchestrating itself to provide the best possible service, these are admirable goals. However, it is probably fair to say that the majority of mechanisms to deliver these capabilities still have a way to go. To illustrate this, let us consider a number of benefits that provisioning and orchestration could have when they are fully realised.

By themselves, and as discussed in this paper, provisioning and orchestration can be seen as drivers towards the more efficient use of IT. In fact, they are enablers of much more than this. We have seen how they need a number of policies and processes in place to be adopted successfully, and it is these processes that will be enhanced. We are not at the point where provisioning is seen as an essential part of the infrastructure. When it is, then "provisionability" will become an essential feature of new applications and hardware, and this will drive the adoption of role-based operational processes in IT. Coupled with this, it is the inclusion of charging, and its association with role-based policy management, that will change the landscape of IT for ever by providing a concrete linkage is made between business processes and the IT resources required to support them.

Meanwhile, the role of orchestration can grow until it reaches the very edges of the network. For example, orchestration software could co-ordinate with custom-designed load balancing appliances to ensure information is transported and processed as efficiently as possible. Better integration is possible at the application layer as well: at the moment, it is up to the application designer to decide where different parts of the application will execute. Truly comprehensive orchestration would be capable of making these decisions dynamically, at execution time, taking the load from the application programmer and reducing development and deployment times. The opportunities offered are vast – for example, all those idle desktops could be being used at weekends to generate the weekly sales reports, or tasks could even be allocated to them when they are sitting idle, or at times of peak demand. There are screensavers for SETI – why shouldn't there be screensavers for corporate data mining?

Application use of provisioning and orchestration is still in its infancy, but over time more and more applications will become enabled to work well in a p-and-o environment. Common application components, such as those supported by operating systems and application servers, will run as multiple instances each of which is aware of what the others are doing. The service oriented architecture advocated by both Microsoft and Java camps will further catalyse the drive towards application software making the best use of an orchestrated infrastructure. Similarly, provisioning and orchestration software will be integrated into applications, both IT-related such as billing engines and non-IT, for example Human Resources automation.

Provisioning and orchestration can grow outward from the enterprise, as applications, systems software and the underlying physical infrastructure are implemented and managed by different service providers. In this case, organisations will need an integrated view not only of their own environments, but also of third party services. Once again, visibility on charging will be the catalyst to make this work.

It is clear that provisioning of the future will be different in practice than provisioning of the past, however it will get businesses back to what they wanted in the first place, which is to get what they pay for as effectively as possible. Now, that's a future worth working towards.