WHITE PAPER

Capacity Reservation System for Virtual & Cloud Environments

Andrew Hillier, Co-Founder & CTO, Cirba

Table of Contents

Introduction

The Innovation Gap

Virtually every area of human endeavour that involves the use of shared resources relies on a reservation system to manage the booking of these assets. Hotels, airlines, rental companies and even the smallest of restaurants rely on reservation systems to make sure they can optimize the use of their assets over time in a way that balances customer satisfaction and profitability. Or, as economists would say, strike a balance between supply and demand.

So why is it that modern IT environments are completely lacking when it comes to having a functioning reservation system to control capacity supply and co-ordinate the demands for it? The answer lies in the evolution of IT hosting models, and the fact that the closed nature of physical and early virtual environments has made it possible to survive without one. But this survival has been tenuous (most IT environments are not the model of efficiency, let alone agility), and all this is about to change with the rise of cloud computing, where the consumerization of capacity is making the modeling future capacity requirements and proper forecasting of demand critical to the survival of IT. If internal IT organizations cannot establish and maintain efficient, scalable and agile infrastructure then there are external providers that can.

Why Capacity Reservations are Important

To explore this in more detail, an analogy is useful. For many years, applications were directly hosted on physical servers, and hosting models were relatively inflexible. This is directly analogous to living in a house, where a large capital outlay gives you a place to live for a long time, and as long as you properly maintain it (and perhaps renovate as your kids grow) you will be fine. Some people even live in the same house for their entire life.

With the popularization of virtualization in recent years, the model shifted to more closely resemble living in an apartment. Sharing common resources provides economies of scale, and capital expenditure can be shifted to be more operational (i.e. monthly rent). And, like apartments, moving in and out typically happens more frequently than in houses, but it is still not frictionless, and requires movers. This, combined with the legal and contractual obligations (lease agreements), tends to cause people to stay for a while, and tenancy is typically fairly long. The same is true of mainframe environments, which have always been virtual, but also have a relatively non-volatile tenancy model.

Pausing here for a moment, a very important observation can be made. Using this example, neither houses nor apartment buildings require reservation systems to manage supply and demand. Houses are not commonly shared, and apartments become vacant and are filled again with such low frequency that it is possible for building managers to deal with in a relatively simple way. Unfortunately, the same is not true for clouds.

Cloud environments more closely resemble hotels than houses or apartments, as there are few barriers to coming and going, and capacity can be used for whatever amount of time is desired. Furthermore, hosting internal clouds on converged infrastructure, where large blocks of capacity are typically deployed at once, is a lot like managing a very large hotel, with a "revolving door" of new customer demands to deal with. With the scale and dynamic nature of these environments, properly matching guests to rooms over time is a tremendous challenge, and doing it wrong will anger customers and make poor use of assets, killing profitability. Throw in some weddings and a conference or two (mass onboarding of consumers) and the situation becomes hopeless. Because of this, one wouldn't dream of opening a hotel without a proper reservation system to carefully manage supply and demand.

Unfortunately, this is exactly what many IT organizations are doing. The trend toward internal cloud is directly analogous to shifting from apartments to hotels, but management systems have not been keeping up, and organizations are being caught short. Attempting to manage "new school" infrastructure with "old school" tooling is like managing a hotel without a reservation system, and can lead to utter chaos. Compounding this, adopting one of the many emerging cloud stacks to deal with this may only make it worse, as that breed of solution invariably focuses on enabling immediate requests, not future bookings, and they typically have no ability to model future demands and do the appropriate forecasting.

Because of this, infrastructure teams are left to wildly over-provision capacity in hopes they will have enough to offset any potential future demand, eroding the potential savings and efficiency associated with operating shared infrastructure in the first place. If they under-estimate then the consequences are equally damaging, resulting in performance and SLA compliance issues, or an inability to fulfill the business's requirements.

The fact that IT has survived without a reservation system for so long may seem unusual, but is justifiable given its evolution. But the fact that many organizations are pursuing cloud without one simply cannot be justified, and must be addressed.

Modeling New Demands

Fundamentals of IT Supply and Demand

There are two main influences on data center demand – trends and capacity reservations also referred to as bookings. Trends are the growth (or shrinkage) in demand caused by natural shifts in user activity, organic growth, and business-led changes, such as M&A activity or marketing campaigns, that impact existing IT applications. Bookings are the new demands that are entering (or leaving) the environment that are related to new application deployment, bulk on-boarding activity (such as physical to virtual migrations or data center consolidation), or other project-based activity. Both are important to forecasting capacity requirements, but both require completely different modeling approaches.

Most IT organizations are somewhat competent at trending demand growth (what some would refer to as "old school" capacity management), but most are lacking the tools required to model future capacity bookings, virtual and cloud on-boarding, decommissioning and supply-side changes (such as the addition of new servers or technology refresh). This is a serious problem, because like hotels, the impact of individual demand trends is often dwarfed by the impact of new workloads coming online (and old ones leaving), particularly in cloud environments. This schism is shaking the capacity management world, and requires a complete rethink of how planning is done in modern IT environments.

Characterizing New Demand

There are three primary sources of demand in virtual and cloud environments: bulk on-boarding of existing applications, the planned release of new applications through standard IT service delivery processes, and self-service requests emanating from cloud users. All three require the capacity to be reserved in the target environment in order to guarantee the fulfillment of their needs, but each has a different way of defining this requirement.

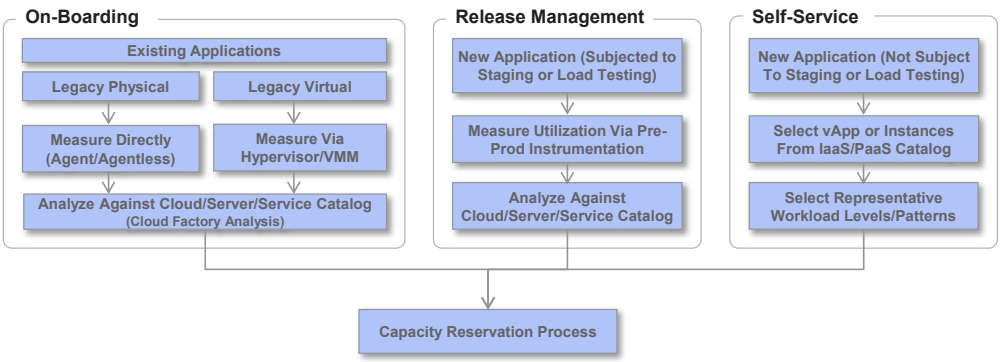

FIGURE 1: Measuring and modeling different types of new demand

Bulk On-Boarding

Bulk on-boarding typically occurs when migration, conversion or consolidation projects move existing applications from legacy infrastructure into virtual or cloud environments. This includes physical-to-virtual and physical-to-cloud migrations, as well as virtual-to-cloud migrations, where early adopters of virtualization are beginning to sunset their older environments. It also includes data center consolidation activity, particularly when it combines facilities migrations with new hosting models, where "waves" of servers are moved into new locations and placed on new virtual or cloud infrastructure as part of the process.

Regardless of the source of the new demand, it must be characterized. This involves either direct measurement of the server activity (via agents or agentless means), acquiring data from hypervisors or virtual machine managers (in the case of virtual workloads), or leveraging data from existing capacity management or performance monitoring systems (if they have sufficient coverage). Some level of configuration information is also required, such as the processor counts and configurations, installed

software details, identification of load balancing groups, etc.. This is needed to understand the workload levels and patterns, as well as the software requirements, so they can be accurately mapped to the cloud capacity models and software configurations (as defined in the cloud catalog, which is covered in more detail below).

Release Management

Most new applications, particularly those that are mission critical in nature, follow prescribed IT service delivery processes in order to make their way into production. The release of such applications is typically well planned, and on their journey to production they pass through a series of pre-production environments for testing, acceptance and staging. The opportunity in cloud environments is to use these steps to gain an understanding of the anticipated demands on IT infrastructure, and to use this to accurately map the requirements into the eventual production environments they will be hosted in. This is a win-win scenario, as the use of cloud infrastructure not only reduces the lead time required to deploy an application (by eliminating hardware procurement from the critical path), but also allows for very precise sizing of the target environment it will run in.

If realistic load testing is not part of the release management process, then virtualization and cloud hosting models also enable other ways to gauge demand. For example, over-provisioned “soaking pools” can be used as part of the release management process to host new applications in order to determine their utilization patterns (using actual production workloads). After the measurements converge (typically by observing an entire business cycle) the application can be accurately sized and moved to a more permanent home. This also allows applications to be routed to the appropriate type of capacity, such as SAN vs NAS-based storage or highly threaded vs “big core” processors.

Self-Service

The new frontier being created by cloud technologies is the ability for end-users to request capacity themselves, rather than to always use IT groups as the intermediary. This may be done as part of a release management process, but more often it is used to support more agile demands, where standard OS builds and software stacks are pieced together to rapidly build new applications or augment existing ones. In these cases, there is typically no legacy or pre-production that can be used to measure application demand levels or patterns, and it is left to the users to estimate what infrastructure they well need.

This estimation is done by selecting cloud instance sizes from a catalog and, if supported by the cloud technology, also selecting a “representative workload” to provide a model of the target utilization. Unfortunately, this process is fraught with inaccuracy, either because the end users do not know what their demands will be, or they know what they will be but do not know how to translate their needs into the arcane language of “Gigahertz and Gigabytes” of IT infrastructure. Users also have a tendency to ask for too much, even if their demands are well known. Clouds that support a significant number of self-service requests often require “resource reclamation” processes that kick in during steady-state operation in order to correct over-provisioning of guest instances.

Analyzing Against Cloud Catalogs

Factory-Based Analysis

With the demand profiles having been characterized through either measurement or estimation, the next step in the reservation process is to analyze these against the supply-side capacity models offered by the cloud technology being used. This is not required for self-service requests, where the users enter their requirements in terms of the cloud catalog in the first place, but is critical to the on-boarding and release management processes.

This "factory-based" analysis must be scalable and repeatable, and should use both quantitative and qualitative analysis in order to answer the following questions:

- Which systems are candidates to host in the cloud?
- Of these, what instance sizes and software stacks do they map to?
- If they are not identical to the models on offer, what remediation steps are needed?
- How should load balancers and application clusters be sized?
- For systems that are not candidates for cloud, what is the optimal alternative hosting model?

By answering these questions, a complete model of new demands can be constructed, that includes not only the anticipated utilization levels, but also the cloud resource allocation requirements and instance configuration details. All of this is required to feed into the capacity reservation process, where the confirmation of available capacity must include both resource utilization as well as allocation limitations.

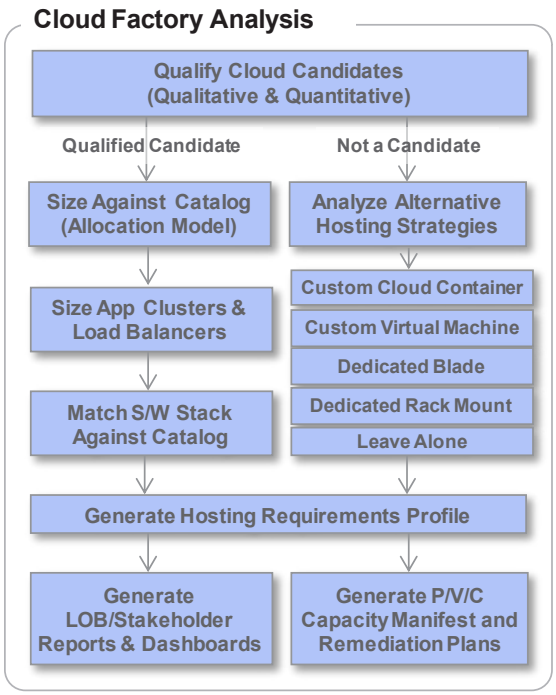

FIGURE 2: Factory-based analysis of cloud hosting requirements and exceptions

The Importance of Policy

The Contract Between Supply and Demand

Before diving into the details of the capacity reservation bookings process, it is important to discuss the role of policies on the management of cloud infrastructure. Policies are becoming increasingly important in the management of shared infrastructure, as they effectively form the contract between supply and demand. By accurately capturing, formalizing and managing to a specific set of business and operational criteria for a specific hosting environment, users can confidently let go of legacy infrastructure knowing that their specific needs will be met and their rights will not be violated.

Cloud Control Policies

The detailed policies that govern the operation of cloud environments are referred to as cloud control policies. Properly-specified cloud control policies cover both quantitative and qualitative criteria, allowing them to represent detailed operational, technical and business requirements. Quantitative criteria include such things as maximum and minimum utilization levels, resource overcommit targets, contention tolerances, and other operational considerations. Qualitative criteria include business rules, technical affinities and anti-affinities, security requirements, process-oriented requirements, etc.. Because these factors may vary from environment to environment (e.g. production vs dev/test) it is common to have several policies active in a given organization, creating pools of capacity that are designed to meet specific application requirements.

This is critical to the booking process, as the ability to place a new demand into a specific environment is heavily governed by the policy that is being used to manage it, and the point at which an environment is deemed to be full is a complex function of supply, demand, trends, bookings and policies. Understanding the policies that apply to new workloads coming into an environment is therefore critical to the capacity forecasting process, as it dictates which environments the workload can go into, whether or not it will fit, and how much capacity it truly requires (which is the basis for reserving capacity).

Hosting Environments

	Production Critical	Production IT	Production Cloud	Production Batch/HPC	Pre-Prod	Dev/Test
Density	Low	Med	Med	Low	Med	High
Performance	High	Med	Med	Very High	Med	Low
Availability	N+2	N+1	N+1	N/A	N/A	N/A
Compliance	Rigorous	Medium	Multi-Tenant	Low to None	None	None
Volatility	Low	Med	High	High	Med	High
Operational Cycles	Business Defined	IT Defined	Unbound	Windowed	Simulated	None
Automation	Approval Based	Semi-Auto	Semi Auto	Fully Auto	Process Defined	Developer Defined

FIGURE 3: High-level comparison of major policy areas across different types of hosting environments

The Capacity Reservation Process

Primary Requirements of a Reservation System

At its most basic level, a capacity reservation system should support the booking of capacity by providing the following capabilities:

- Ability to capture and/or receive demand profiles from the various sources described above
- Ability to assess whether the demand will fit into the target environment at the future date it is scheduled to be deployed (taking into account trends as well as other confirmed bookings)
- If it fits, the ability to "lock" the capacity so it cannot be usurped by another user/application

Although this may sound simple, it actually requires a fairly sophisticated bookings management process, as well as very sophisticated predictive analytics in order to model the future-state scenarios being assessed.

Detailed Capacity Reservation Process

The following diagram provides a detailed process flow for the reservation of capacity:

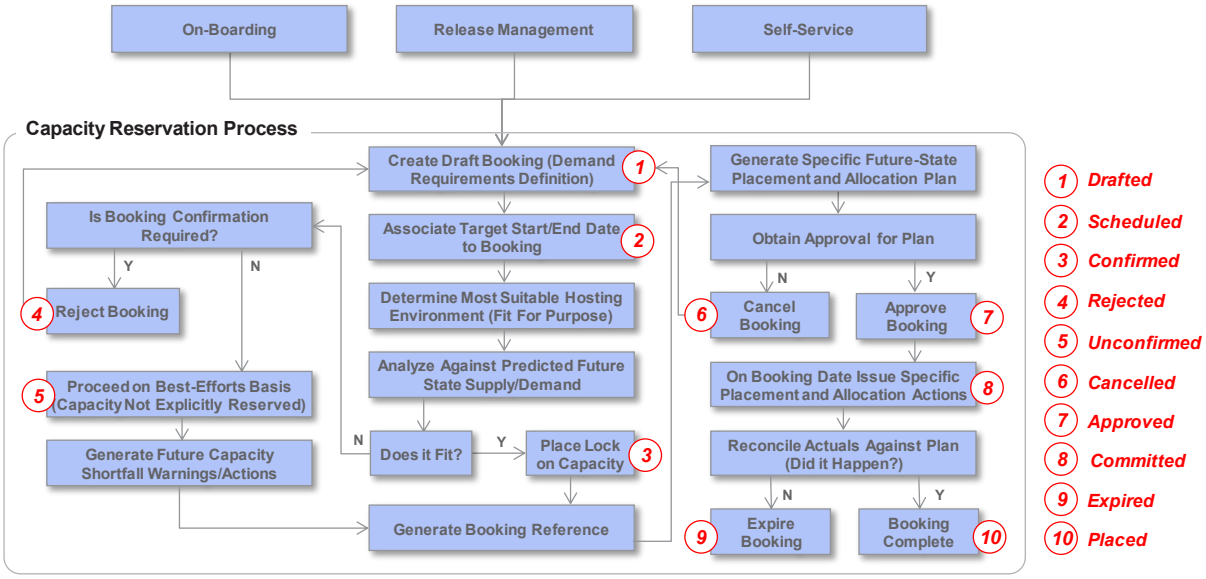

FIGURE 4: Detailed capacity reservation process, showing key flows and resulting states

Throughout the reservation process, a booking goes through several key states:

- **Draft** – the demand requirement has been defined and captured.
- **Scheduled** – a start date (and sometimes an end date) has been associated with the booking.
- **Confirmed** – the demand has been analyzed against the future-state models of the environment and has been deemed to fit. The future state models have been updated to incorporate the demand, so it has priority over future booking requests.
- **Rejected** – the demand does not fit into the target environment.
- **Unconfirmed** – the demand does not fit, but the policy does not require confirmations, so the demand will be entered into the future state model. This effectively means that the booking is granted, but that infrastructure managers will need to add capacity to the environment before the start date in order to not experience a capacity shortfall.
- **Cancelled** – the demand passed the technical hurdles but the action plan to actually make it happen was rejected, meaning it failed to obtain business or ITSM process-level approval.
- **Approved** – the action plan to realize the booking was approved.
- **Committed** – the start date of the booking has arrived and the specific actions to realize it have been "locked and loaded" in the appropriate provisioning, orchestration and/or ticketing systems.
- **Expired** – the action plan was committed but was not executed (for reasons specific to the automated or manual processes being employed), meaning the booking must be either re-created or rescheduled.
- **Placed** – the action plan was executed, and the new instances are fully operational, signifying the fulfillment of the booking.

Although this is slightly more detailed than the process to book hotel capacity, it effectively serves the same purpose by ensuring that applications have the capacity they need when they need it, without forcing infrastructure managers to wildly over-provision their environments to deal with uncertainty.

The Role of Predictive Analytics

The lynchpin of the entire booking process is the ability to confirm that an anticipated demand can actually fit in to the target environment at the desired future point (and all points beyond), and once it is formally booked into that environment, that the capacity is held for it until it is actually deployed. In the flowchart, this is accomplished through the analysis against predicted future state supply and demand.

As described previously, this analysis must take into account existing demands, new bookings, workload trends, capacity supply (and upcoming changes to it), and the control policies governing the environments. And because application workloads carve out complex patterns over time, the confirmation of whether a given set of workloads will safely fit into the available server capacity is non-trivial. It requires looking at many dimensions of data, and assessing all the permutations and combinations of activity that can lead to operational risks. In this sense, it is a lot like playing a complex game of Tetris, where blocks are not only appearing and disappearing, but are growing and shrinking over time, and are subject to complex policies and operational constraints.

To distill the output of this analysis into an easily leveraged form, the concept of an efficiency index is useful. If a virtual or cloud environment has an efficiency index of 1.0, this means that supply and demand are perfectly matched, and based on policy the workload levels and patterns stack up to exactly use the available capacity. An efficiency index of 0.75 means that the workloads could be safely be hosted on three-quarters of the capacity currently deployed, signifying that the environment is over-provisioned and that there is space for new workloads (or, put another way, density can be safely increased). And an efficiency index > 1.0 means that the environment is not only full, but is under-provisioned, and new capacity must be introduced (or demand removed) in order to alleviate the problem.

FIGURE 5: Visualization of efficiency index of five virtual clusters based on each environment's specific policy. In this case, four clusters currently have excess capacity, while one is essentially full

Marrying this concept with future-state analysis, it is possible to compute the efficiency index of an environment at a future point in time based on trends, bookings and policies. This allows the booking process to use be assessed using a much more intuitive criteria: if the introduction of new demand at a future date drives the efficiency index beyond 1.0 for that date (or any date beyond it), the booking will be rejected. Although this is based on some fairly sophisticated analysis, it is conceptually very simple.

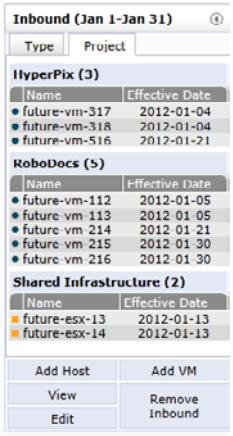

FIGURE 6: Bookings view showing two new applications due to come online (in the blue cluster), as well as 2 new ESX servers being added (to the yellow cluster) in the next 30 days

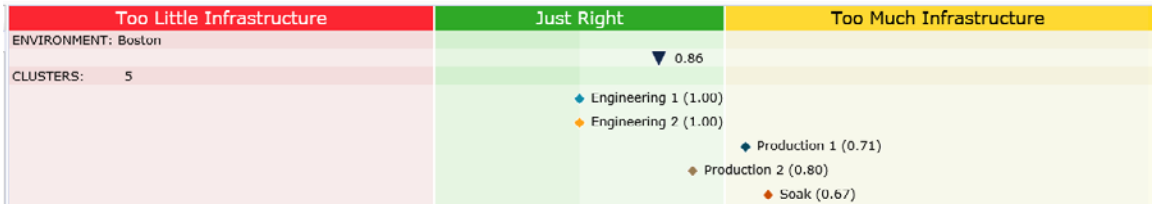

FIGURE 7: Predictive analysis showing 30-day look-ahead for the same environment, accounting for all bookings, demand trends and supply-side changes. Both blue and yellow clusters are forecast to be at optimal density, and the bookings are automatically accepted and confirmed

This predictive analysis is the basis of capacity forecasting in virtual and cloud environments, and "old school" trending-only approaches simply are not capable of representing the complex supply and demand models of cloud environments.

The Impact on IT Operations

The impact of accurate capacity forecasting on both suppliers and consumers of capacity is quite significant. It not only allows supply-side infrastructure managers to right-size their infrastructure (saving millions of dollars), but gives demand-side consumers greater confidence that cloud infrastructure will meet their needs. Proper forward-looking analytics, based on agreed upon policies, allows infrastructure managers to give "official confirmation" to application groups that capacity has been reserved to meet their future needs.

This analysis also forms the basis for new and interesting models, some of which also parallel the hotel booking model. For example, by rewarding advanced bookings with lower costs, and penalizing last-minute bookings with higher costs, behavior can be shifted to promote better planning among users, reducing volatility and increasing efficiency. Just as walking into a hotel lobby and asking for a room at the last minute is both risky and expensive, last-minute cloud requests may eventually be viewed the same way. This is good for everyone, as it helps eliminate unplanned, reactionary operational models.

Conclusion

It would be silly to open a new hotel without a reservation system, and in the not too distant future the same may be said of cloud infrastructure. Given the similarities between hotels and clouds, it is ironic that the use of reservation systems is not more common in IT environments. But this will likely change quickly as internal cloud gains more and more traction, and by drawing parallels to other similar business outside IT, the requirements for, and process of booking capacity will become more and more clear.

About Cirba

Cirba has re-imagined infrastructure control for the software-defined era. We're enabling the world's most successful organizations to scientifically balance infrastructure supply and application demand—creating a demand-driven approach to infrastructure management that maximizes efficiency and cost-savings while reducing risk.

About the Author

Andrew Hillier, Co-founder & CTO, Cirba, Inc.

Andrew Hillier has over 20 years of experience in the creation and implementation of mission-critical software for the world's largest financial institutions and utilities. A co-founder of Cirba, he leads product strategy and defines the overall technology roadmap for the company.

Prior to Cirba, Hillier pioneered a state of the art systems management solution which was acquired by Sun Microsystems and served as the foundation of their flagship systems management product, Sun Management Center. Hillier has also led the development of solutions for major financial institutions, including fixed income, equity, futures & options and interest rate derivatives trading systems, as well as in the fields of covert military surveillance, advanced traffic and train control, and the robotic inspection and repair of nuclear reactors.

Hillier holds a Bachelor of Science degree in computer engineering from The University of New Brunswick.

45 Vogell Road, Suite 600
Richmond Hill, ON
Canada, L4B 3P6

t: +1.866.731.0090
t: +1.905.731.0090
f: +1.905.770.4082
www.cirba.com