Optimizing resource efficiency in Microsoft Azure

By July 2017, Core Services Engineering (CSE, formerly Microsoft IT) plans to have 90 percent of our computing resources hosted in Microsoft Azure. As one of the largest Microsoft Azure customers, we’re constantly evaluating how we use and manage our cloud resources. We’ve established standards and tools for managing our Azure footprint and optimizing our environment. Ensuring that Azure infrastructure is used as efficiently as possible is very important, and increasing efficiency in a big environment means big cost savings. We’ve incorporated people, processes, and tools into our cloud strategy—this has resulted in a 38 percent reduction in cloud spending and an increase in CPU utilization efficiency across our infrastructure as a service (IaaS) environment by over 400 percent, while still aggressively reducing our on-premises footprint.

Investing in Azure at Microsoft

Our goals as an organization include widespread Azure adoption, and to be the first and best customer of Microsoft products and services. We strive to be a recognized industry leader in cloud adoption—both to benefit from the technological innovations in the cloud, and to adopt a product that we trust to host our enterprise IT infrastructure. We also want to help customers realize efficiencies for IT by using the cloud.

We first set out to migrate our development and testing datacenter resources to Azure to avoid large capital expenditures updating datacenter hardware and renewing building leases. As we gained momentum on our cloud journey, new targets were set that included production resources—and this year, we will move over 90 percent of all compute needs into the cloud. Currently, our IaaS computing resources hosted in Azure look like this:

- 20,000 Azure virtual machines
- 110,000 cores utilized
- 1,500 Azure-based applications

As we continue to migrate to Azure, our need for effective management and utilization has become a significant business concern and a focal point for our leadership and engineering teams. Our early adoption of cloud technology was focused on getting solutions and resources migrated from on-premises to Azure. Many of these migrations were performed as “lift and shift”—the on-premises architecture was virtualized and placed in Azure IaaS, with the intent of optimization sometime in the future. The lift and shift migration was fast paced—our Azure resources grew quickly, and our subscription base became very large with many Azure resources being underutilized. We’ve realized the need for optimization and management of our Azure resources, and we understand the business benefits that optimization brings. The requirement for optimization will only become more important in the future as our Azure environment continues to grow. For example:

- We anticipate a 31 percent increase in Azure workloads in the current fiscal year, moving from 65 percent to 90 percent.
- Our on-premises footprint continues to decline. Our physical server presence decreased 30 percent in the last year, and 70 percent in the last two years.
- Azure operating system instances have increased by almost 10,000 over the past two years, whereas on-premises operating system instances have decreased by more than 28,000.
- Migration to platform as a service (PaaS)–based solutions is being heavily promoted for high value on-premises internal solutions and newly designed apps.
Fitting Azure to meet our needs

With a large amount of cloud infrastructure in place, we recognized the need to optimize our Azure resources. To address this, IT created Azure Resource Optimization (ARO), an internally developed combination of tools, processes, and education for Microsoft teams to examine both their total cost of cloud resources, and the amount of underutilized assets. Many different resource types are evaluated to identify cost opportunities, such as IaaS virtual machines, Azure SQL Databases, PaaS Web and Worker roles, and unused Azure Storage. The recommendations provided by ARO include adjusting the SKU size, deleting unused resources, or turning off resources during periods of downtime. The ARO program’s overall goal is to drive proper usage of capacity in Azure by CSE and to reduce the footprint of unused resources by targeting areas such as underutilized servers, misconfigured resources, and other unused resources.

ARO is also dedicated to creating an optimization-centric culture. We began the optimization process by reviewing all of our Azure infrastructure and identifying opportunities for optimization. However, this was simply the initial phase. Having a cultural focus on optimization causes everyone to consider how they use resources. As teams review the ARO dashboard on a weekly basis, the reality of assets becoming pay-by-the-hour instead of locked budget items becomes apparent. This is especially important when our engineers are designing new solutions in Azure and making design decisions that result in solutions that use native scalability. We’ve moved away from the idea of selecting the largest hardware just in case the app might spike due to specific business events in the future. In the on-premises world, teams would acquire servers for a project with one-time capital expenditure budgets, and server utilization wasn’t a concern because the hardware was a fixed cost. In a cloud-first world, we pay by the hour, and that turns our old practices sideways. Each hour counts, and each core counts. After leadership and engineers start seeing the value of CPU utilization, core count, and hours consumed, the true flexibility of the cloud starts to produce business benefits. We can adjust our cloud expenditures in real time, something that was never considered in an on-premises world. Leaders championed the flexibility and efficiency with their teams and drove for wide-scale changes to how we manage our infrastructure. Large cost savings were seen across CSE in a very short period of time.

Figure 1. Azure migration results from CSE
Establishing a lifecycle of optimization

After our ARO dashboard was launched, usage and optimization trends became apparent. Aggregating the total cost savings potential was shocking, as hundreds of servers with SKU sizes too big made for large portions of our daily Azure cost. The problem had been clearly illustrated. After the recommendations are shown, teams can drill down into the performance data to understand which size SKUs to select, when to turn off virtual machines that aren’t in use, and which unused virtual machines to delete. After application teams see the data and validate the recommendations, making the changes is easy. API calls are used to adjust the SKU size or delete the resources from a simple front-end web console. Due to the speed and flexibility of the cloud, teams can watch their cost drop in hours after making the changes, and the dashboard updates to reflect the latest performance data and cost analysis. Watching the benefit of reducing core counts and resource types in near real time provides a great benefit to teams that are trying to stay within their budgets. As a result of ARO efforts, the following patterns of behavior emerged, and we have adopted them as our optimization model:

1. Identify the optimization opportunity.
2. Examine and validate the data behind the recommendation.
3. Use tools or API based utilities to implement savings.
4. Assess results and restart the process again.

These stages are represented in the following graphic.

![Figure 2. The Azure lifecycle of optimization](image)

We’ve established a governance model to contribute to Azure cost optimization that consists of several groups working together. Here are some of the most important groups and how they are involved in the process:

- We take the point of view of a customer with Microsoft Azure. In doing this, we better reflect the customer/provider relationship and work with the Azure product group to test and improve Microsoft Azure as a product.
- Our CIO and the CIO’s direct reports provided a focus on adoption and optimization, in addition to sponsorship for our governance and Azure adoption in general. This sponsorship provides support for our processes and helps drive Azure adoption by all business groups across the organization.
Our IT engineering teams contribute in several ways that facilitate Azure optimization:

- Cloud team leaders hold weekly progress review meetings that are focused on cloud adoption and cost optimization.
- A PaaS working group was established to drive conversations around enterprise PaaS architectures.
- Several large training summits have been created to provide education about cloud technologies, the future of Azure, optimization, Azure Resource Manager automation and templates, and other topics. The Engineering Fundamentals Summit and Microsoft Cloud Summit helped address the effort of accelerating CSE adoption of cloud technology and best practices.

### Generating optimization recommendations

Identifying poorly utilized servers is the starting point. The ARO team monitors on-premises datacenters and Azure virtual servers daily, using performance counters from System Center Operations Manager (SCOM). Processor, memory, and hard drive data is gathered, and then the team uses industry-standard P95 values to determine whether specific assets are underutilized. We categorize all servers into five performance categories: frozen, cold, warm, hot, and on fire. Teams often use hardware that’s much larger than needed, which leads to very low utilization. Part of the ARO program is designed to educate engineering organizations about the cost associated with picking server sizes that are too large.

Underutilized servers are often abandoned because of budget cuts, project cancellations, reorganizations, or obsolete hardware. Underutilized servers that are on-premises are prime candidates to move to the cloud—where the elasticity benefits allow us to correctly size them to their business value, upgrade them to the latest CPUs, or turn them off during hours when they’re not in use. The team is migrating operating system instances and SQL databases to Azure at a fast pace to vacate datacenters. Analyzing cold or frozen Azure servers ensures that these servers are either resized down or turned off when they’re not in use.

### Collecting data

Collecting data to identify poorly utilized IaaS virtual machines and Azure SQL Databases is the starting point. For IaaS (and on-premises) virtual machines, SCOM collects local performance data such as CPU, memory, and disk I/O. For Azure SQL Databases, we use the Database Transaction Unit measurement directly from Azure—it’s a combination of performance monitors for usage. For other resource types, we collect metrics via Azure APIs.

### Generating performance analytics results

The performance analytics data we collected was used to identify opportunities for optimization. Our key evaluation processes looked something like this:

- Establish a timeframe to collect performance data for a recommendation (3 months for virtual machines, 14 days for Azure SQL Databases).
- Create a recommendation that’s based on the most heavily utilized component of the server (disk I/O, memory, CPU) with industry-standard P95 values to ensure that the recommendation doesn’t harm the software running on the virtual machine.
- Give the ability for the engineer to review the performance metrics to validate that the recommendation makes sense.
- Show the cost difference between the current configuration and the recommended configuration.

### Enabling optimization with our right-sizing toolset

Our toolset for executing the changes required in the lifecycle of optimization involve both graphical and automated solutions that interact with our Azure environment:

- **ARO dashboard.** Our internally developed dashboard is where it all starts. Aggregating a simple cost view with all of the recommendations across IaaS and PaaS resources shows how much of current spending could be...
removed without business impact. Drilling down into the recommendations provides the specific line-by-line recommended changes, and links to the tools in this list to execute the change. Views into specific subscriptions and servers are also available in this dashboard to help with asset and subscription management.

- **Cloud Cruiser.** Cloud Cruiser is a third-party software as a service (SaaS) application that gives us very valuable financial information and reports about our Azure usage and spending as real time as possible. Using Cloud Cruiser, we can examine and aggregate financial data across multiple global Azure subscriptions, which is crucial. Our Azure environment spans a large number of subscriptions across the business—this gives us the “real-time” visibility that’s required to better manage and control.

- **Snooze.** To support the concept that non-production cloud servers need to be online only when employees are actively working on them, a suite of tools was created to return those compute hours and save significant costs. In some cases, non-production environments can be turned off, or de-allocated, over 70 percent of the time, which is a direct 70 percent cost reduction. Snooze options include:
  - **On demand snooze.** This has a simple API-driven GUI, where a user can type a server name and click Snooze, and the server is deallocated—instantly stopping the cost of that IaaS virtual machine. Additionally, any user can start a virtual machine in this tool.
  - **Scheduled snooze.** This is a calendar-based scheduling utility that teams can use to select what days and times that they want their servers to stop and start, again saving significant cost.
  - **Megasnooze.** This is an automation solution that reviews SCOM data and snoozes any server that is idle for more than 48 hours. This ensures that teams that don’t have healthy snooze practices still save considerable money.

- **Resize.** The ability to quickly change the size of a server in Azure is the enabling technology behind most cloud optimization. With a simple reboot, dropping the size and cost of a virtual machine by more than half, at large scale, allows us to quickly use the flexibility of the cloud to recapture IT spending. Resize options include:
  - **On-demand resize.** This has a simple API-driven GUI, where a user can type a server name and select either the recommended SKU, or one of their own choosing, and resize the server. A user can also select to do the resize at a scheduled date and time, combining both the on-demand and scheduler functions into one UI.
  - **Automated resize.** Simple API automation combined with recommendations is just a script away. When certain divisions within the organization request it, we can automatically resize all virtual machines for them in a scheduled fashion, such as over a weekend. We used the same code to resize all on-premises virtual machines in CSE over four weekends—out of 10,000+ resizes, not one incident or app team escalation occurred.

### Seeing business results

Right-sizing Azure with our processes and tools has reduced our cost and massively increased utilization across our Azure environment, resulting in the following cost and operations benefits:

- A 38 percent reduction in cloud spending from optimization activities such as:
  - 9,000+ Azure resize requests.
  - Almost 30,000 Azure virtual machine snooze requests.
  - 7.7 million cumulative virtual machine hours in snoozed state where we weren’t being charged.
- An increase of almost 400 percent in CPU utilization. In six months, CSE has moved from 4.5 percent average CPU utilization to 16 percent average CPU utilization across our Azure IaaS instances.
- A reduction in our IT infrastructure budget. We were given a target to reduce our IT infrastructure budget across CSE as part of the push for Azure adoption, and by budget reporting time, we will have achieved that goal.
- A decrease in the number of operating system instances. We’ve decreased the overall number of operating system instances by more than 20,000, which greatly reduces infrastructure costs.
- A more agile development and deployment process. Moving to Azure has enabled a more agile development environment. We’ve also used GitHub for code-sharing across CSE, which has increased code reuse and
transparency. These development processes also provide additional benefits, such as incorporating our security review cycle into the engineering process, rather than managing it separately.

- An optimization-centered culture. Having a cultural focus on optimization causes everyone to consider how they use resources. This encompasses the optimization of existing solutions, the development of new solutions, and how our teams approach the cloud in general.

By right-sizing Azure, our cloud team has been able to provide an efficient platform for delivering our solutions while continuing to optimize costs as our cloud environment grows and changes monthly. As we continue to rely more heavily on Azure, we’ll continue driving business value by examining new ways to use every last processor for CSE.

CSE is in a unique position while we quickly adopt new Azure technology as our first and best customer. Each new cloud management and optimization scenario we address directly feeds back into the Azure product group, which often turn into features. The goal for these solutions across CSE is not just to provide world-class cloud management internally, but to enable Microsoft customers to also benefit from these experiences. When we look forward toward a PaaS-first future, more management scenarios will be uncovered and incubated. We also see the optimization concepts applying to technologies, such as service fabric and containers. CSE continues to strive to be a cloud leader—and share our methodologies along the way with customers and the Azure product group.

For more information

Microsoft IT Showcase

microsoft.com/ITShowcase

© 2017 Microsoft Corporation. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.