Modernizing “Heartland Escapes” to a Cloud-Hosted Infrastructure

Aidan Polivka

CS630 – Modern Operating Systems

Colorado Technical University

October 9, 2023

Table of Contents

Project Outline

“Heartland Escapes” is a bookstore located in Nebraska. They have been in Lincoln for quite a few years, opening their first store in 2010 and their second store in 2019. The store has recently seen a lot of success through a new marketing scheme using popular social media platforms like Instagram and Tik Tok. Much of their popularity can be attributed to the regular events they host, like the “school’s out reading program”, author meet-and-greets, and Halloween scary story readings. Because of their recent spike in consumer interest, they plan on expanding their business to neighboring cities, one location in Omaha and another in Grand Island.

Since their store first opened in 2009, “Heartland Escapes” hasn’t had any major upgrades to their technological hardware or software besides minor increases in disk space, ram, and a processor upgrade. Their current system is hosted on the premises of their first store location and consists of a windows 2008 server machine with an Intel Core i7-8700 processor. The Intel Core i7-8700 64-bit processor operates with 6 cores and 12 threads (Intel® CoreTM I7-8700 Processor (12M Cache, up to 4.60 GHz) Product Specifications, n.d.). Its Clock rate is 3.2 GHz, its L2 Cache is 256 KB for each core, and its L3 Cache is 12 MB. The server works with 8 gigabytes of RAM and has 8 TB of Hard Disk space, 3.5 TB being occupied. This server hosts a home-grown point-of-sale system, a home-grown inventory system, and a public website all written in .NET Framework 4.8 web forms, along with a Microsoft SQL Server containing an inventory database and an accounting database. A network of 8 machines between the two locations consume the Point-of-Sale web application and Inventory service, and the website interacts with the inventory service so that users can search for books in each store location from the comfort and convenience of their own homes. With the addition of two new stores and the added complexity of an ever-expanding consumer base, “Heartland Escapes” is looking to upgrade their system to take advantage of the scalability, monitoring, security, and management simplicity of a cloud hosted system.

Seeing as their current system is very Microsoft heavy and they expect continuous growth, my proposal to “Heartland Escapes” is to migrate their infrastructure to Azure and to keep their on-premises server to store a physical backup of their cloud system. I think migrating to the cloud would be highly beneficial for their expectation of seeing continuous growth, and cloud services offer a lot of opportunities for scaling up hardware for periods of increased traffic rather than needing to always have CPUs available. This is would easily resolve the domain need created by “Heartland Escapes” periodic promotional events. Azure is the cloud platform that I’m most familiar with, and it’s also a Microsoft product which would be brand consistent with their .NET applications and MS SQL Server database management system.

I think the configuration in Azure should be separated into two resource groups, one for data and infrastructure and the other for applications, with all resources being in the us central region. Starting with the application storage container, there would be three app services. One for the Point-of-Sale web application, one for the Inventory Service, and one for the public website. Azure App Services are containers with a configurable base operating system, and in this case, I think all services would run

on a windows system for simplicity since deploying to a different operating system could potentially lead to unforeseen issues. The app services would live under an app service plan using the Premium v3 P1V3 hardware tier. This tier has two vCPUs and 8 GB of RAM per instance and can scale up to 30 instances. This would be a starting point to scale up or down from. In the data and infrastructure group, there would be an Azure SQL Server hosting both databases, and there would be a virtual network (vNet) that manages firewall rules and security concerns between the secure parts of the system, and a vNet Gateway would exist so that users could access the system via VPN. The public website would be the only aspect of the system outside of the vNet. The vNet would only be accessible via gateway, and the website’s public IP address would have access as well.

OS Processor and Core

Migration Benefits

There are many benefits to migrating to Azure. On the surface it looks like the new hardware is less capable than the existing hardware, but that isn’t true at all. Each service instance has access to 2 virtual CPUs, and there are 30 available instances, meaning the system now has access to 60 virtual cores where the whole system originally shared 6 cores! The applications should not need that much processing power either, and Azure allows administrators to easily scale the application resources up and down. So as the demand for this system grows, the hardware can grow with it naturally at the click of a few buttons. The scalability of Azure App Service instances is so advanced that you can create rules for scaling that dictate whether another instance of the app service should be created based on date, CPU percentage thresholds, memory thresholds, and minimum and maximum threshold limits (Ed Baynash, 2023).

Because the system is now distributed into individual azure components, the load on the system is localized to individual services rather than the entire system itself. So, if the database is performing a large batch operation that’s taking up resources, the deployed services aren’t going to feel a slow down (unless they’re accessing the database). Not only is the load better distributed in this way, but again the scaling aspect applies here. Each service could be configured to have a maximum of 10 instances, and to increase or decrease the instance number by one based on average CPU Percentage metrics or RAM metrics. With that configuration it would take a lot of traffic to overwhelm any of the three services.

An additional benefit is the increased reliability of the system. These services are now decoupled from a single machine and are running on an operating system managed and upgraded by Microsoft. This means that there is no longer the single point of failure from being hosted by an on premises machine, and if there is a failure on an instance of an app service then another instance can pick up the slack. This also means that any security updates for the underlying operating system are not the responsibility of the individual maintaining the system, but the responsibility of Microsoft. This deferred responsibility is a positive in this case, because Microsoft is going to ensure that their PaaS services are as secure as possible so that they aren’t liable for any breaches. Microsoft can make these upgrades with very limited down time. Speaking of down time, the P1V3 service tier boasts a 99.95% yearly availability, which could arguably be more available than the on-premises system depending on what “Heartland Escapes” machine maintenance looked like.

Distributed Systems and CPU Scaling

This upgrade is by all definitions distributed and virtual. What I mean by this is, the individual applications that were originally hosted on that single machine are now distributed into their own containers with exclusive access to a pool of processing, memory, and security resources. Azure being used as a Platform as a Service allows for this to be done very quickly and easily through their Azure Portal UI. These distributions are virtual as well. As stated in the project outline, Azure App Services are really containers, running on the base operating system defined by their App Service Plan which is a virtual host of the containers. At the heart of every cloud service is really a massive server farm with thousands of processors, even more cores, terabytes of RAM, and petabytes of disk space distributed into many virtual machines.

Because of this, there is really a lack of control around the specific processor used for your deployed Azure services. What is available to a user of Azure is the ability to upgrade the number of virtual CPUs. There isn’t much information online about how many cores are available in an Azure vCPU, they use a different unit of measurement for performance since multiple vCPUs can span across the same physical core. You can increase the processing power depending on the tier of your App Service Plan, but I haven’t been able to find a direct comparison of the Azure Compute Unit (ACU) to existing processors, or how that ACU is calculated (Micah McKittrick, 2022). So, at my level of understanding, to “upgrade a CPU” you really need to upgrade the App Service Plan that hosts your App Services. Because of how obfuscated the user of the cloud service is from the details of the individual vCPU, the decision of whether to upgrade the App Service Plan tier (effectively upgrading the vCPU) is dependent upon cost of the upgrade, how much additional performance is expected from the upgrade, whether that performance upgrade is crucial to the design of the system, and bureaucratic approval. There isn’t much additional need for migration or security concerns since that’s all baked into Azure as a service. Since Azure has advance logging and performance reporting, testing the upgraded plan would mainly consist of monitoring.

Scheduling Algorithms

OS Concurrency Mechanism

OS Security Risks and Mitigation Strategy

Future Considerations Using Emerging Technology

References

Intel® CoreTM i7-8700 Processor (12M Cache, up to 4.60 GHz) Product Specifications. (n.d.). Ark.intel.com. https://ark.intel.com/content/www/us/en/ark/products/126686/intel-core-i7-8700-processor-12m-cache-up-to-4-60-ghz.html

Ed Baynash et. al. (2023, April 10). Get started with autoscale in Azure - Azure Monitor. Learn.microsoft.com. https://learn.microsoft.com/en-us/azure/azure-monitor/autoscale/autoscale-get-started?toc=%2Fazure%2Fapp-service%2Ftoc.json

Micah McKittrick et. al. (2022, April 29). Overview of the Azure Compute Unit - Azure Virtual Machines. Learn.microsoft.com. https://learn.microsoft.com/en-us/azure/virtual-machines/acu