The use of Microsoft Azure for high performance cloud computing – A case study

Henrik Alderborn
Abstract

BM System is an Uppsala company that develops intelligence management systems. They are currently using a third party provider of virtual machines. With the increased use of cloud computing technology over the last decades, BM System wants to explore if there are cloud computing solutions that could be possible to use for increased capacity and flexibility. Since the company is using Microsoft products for other applications, the cloud computing software Microsoft Azure is a primary choice to have a closer look at.

The aim of this thesis was to choose a Microsoft Azure service that would be beneficial for BM System to use as a complement to the current on-premise solution, and to evaluate the performance, price and functionality of the chosen Azure service.

The thesis shows that Azure Batch may be a good service for BM System to use. Tests showed that a virtual machine of the FSv2-series would be the best choice. Features to chain virtual Microsoft Azure machines after one another and use an automatic scaler of computing power within Azure Batch was also shown to be successful.
Acknowledgements

I want to thank my supervisors Johan Pettersson and Mikael Östlund at BM System for all their support and advice that they have given me during the work with my bachelor’s thesis. I have appreciated their engagement, daily support, positive attitude and valuable feedback.

I would also like to thank Qi Lin at Uppsala University for taking on the task of reviewing my thesis and giving me constructive comments.

Finally, I want to express my appreciation to BM System for letting me do my thesis work at the company and for making necessary equipment available for me. My sincere thanks!
1 Introduction

Cloud computing has made large strides in the last decade, both in terms of technology and number of customers. Cloud computing includes everything that is a hosted service being delivered over the internet, such as data storage and virtual machines for customers to use [1]. This is called cloud computing technology and involves how to combine cloud technologies into a joint service, a cloud service. Several big companies like Amazon, Google and Microsoft have during recent years focused a lot of resources toward their cloud services, and they now offer plenty of different services to both large and small companies, or even private individuals, often with an affordable price as well.

1.1 BM System

BM System is a software company located in Uppsala [2]. It was founded in 1991 and is a leading company in developing intelligence management systems. The company has extensive expertise in GPS-based guiding systems that are provided to customers around Sweden as well as abroad. Two of the biggest systems are "BM Road Service System" and "BM FetchPlanner System", with the former being a system to handle winter road maintenance (i.e. optimal routing for snow plowing), the latter being a system for handling garbage collection. Both of these systems contain large optimizing problems. Resolving these issues require a large amount of computing power. The time for completion of an optimization problem can span everywhere between 5 minutes and multiple days. Some have strict time limits, usually one hour.

2 Background

With cloud computing being a major part of this thesis, the following section covers the basics of cloud computing.

2.1 Cloud Computing

Cloud computing is a rather general term and simply refers to the access and use of software and data over the internet [3]. More specifically the term cloud refers to different types of distributed computing like a cluster of servers, network or software. The term computing refers to the ability for users to use the cloud as a service.

There are four different cloud deployment models [4]:

1. Public clouds are available for everyone and are managed by a company - often with a commercial aspect in mind.

2. Private clouds are exclusively used by a single company or an organization. The infrastructure can be hosted by a third-party company or by the
company itself. Private clouds often provides better security and higher customization than a general public cloud.

3. Hybrid clouds are a mix of both public and private clouds. Sensitive data can for example be placed on the private cloud while scaleable computing power can be used on a public cloud.

4. Community clouds are used for a limited set of organizations with some common interest, like a set of different banks.

Services provided by the cloud computing companies are often divided into three categories or service models. These three service models are defined as the SPI-model, see figure 1. SPI is a compound of the first letter in software, platform and infrastructure.

- **Infrastructure as a service (IaaS)** - The cloud company provides resources for storage, computing and network on-demand. Instead of buying computing resources and installing them on premise, the customer rents it [5, p. 19]. The customer has to be responsible for all kinds of configurations that are needed. The greatest benefit with this model is that the customer is in full control of everything that is running on these resources.

- **Platform as a service (PaaS)** - With a PaaS model the provider offers not just access to the hardware but also various softwares that the customer needs to develop and maintain an application.[5, p. 20]

- **Software as a service (SaaS)** - A whole software solution is provided for the user, ready to be used. The main difference between PaaS and SaaS is that PaaS is a service for application development while SaaS is a service that is already developed and ready to use [6].

![SPI Model](image)

**Figure 1: SPI Model**
There are plenty of benefits with cloud solutions for a company:

- **Elasticity or scalability** meaning that the computing power provided by the computing provider can change on demand [5, p. 10]. For a consumer this means that you can always get enough computing power, whenever needed. The computer power can change in just a couple of minutes, which is perfect when the stress on the product for example increases during peak hours. This also allows companies with different workloads during different parts of the year to lower the computing power during downtime and set it to normal power during active periods. Elasticity also easily enables the possibility to increase the business and you do not have to worry to overbuy or underbuy hardware, which might be the case when going on premise.

- **Cost** - The main benefit of scalability with cloud services is the capability to save money. You don’t have to pay for computer power that you don’t need. This is especially useful for new companies that do not want to buy all the necessary hardware as well as having the cost of setting everything up and maintaining it [1].

- **Maintenance** - With a cloud service the provider of the service has a responsibility to handle and maintain the hardware. With more application oriented services (PaaS and SaaS) the provider also has the responsibility to maintain and update the software that the consumer is using [1].

- **Availability** - A cloud computing service often has higher uptime than a similar service on premise.

A popular billing method with cloud computing is pay-as-you-go. With pay-as-you-go you only pay for what you use. This principle goes hand in hand with the previously mentioned advantages of cloud computing.

There are however also disadvantages that one has to take into account when using cloud computing services.

- **Security** - A huge number of users using resources and a lot of sensitive data on the cloud may well increase the assiduity of hacking attacks. As a consumer of a cloud service, there is not much you can do to increase the security [5, p. 175]. Almost all responsibility lies at the cloud providers to manage the security.

- **Configuration** - Depending on the service, there are less room for low level configuration of for example the hardware or the operating system.
There are plenty of companies that provide cloud computing services. Some of these are:

- Microsoft Azure
- Amazon Web Services (AWS)
- Google Cloud Platform

Microsoft Azure will be the cloud provider evaluated in this thesis, because BM System is already using many of Microsoft’s products. BM System also has some previous experience with Microsoft Azure.

3 Microsoft Azure

Microsoft Azure is a cloud computing service with around 100 different services that includes everything from general storage and virtual machines to AI functionality and internet of things (IoT) services. Azure offers a pay-as-you-go approach which means that you only pay for the time you use a service. There exist deals with 1 or 3 years reservation which can be cheaper if you in advance know your planned usage. Each Azure resource used, is associated with a region in the world (e.g. Sweden Central).

Azure Storage and Azure Batch are two Azure services that are important in this thesis. Both Storage and Batch contain concepts and terminology that are important to understand when using these services. The basics of these services are covered in the following sections.

3.1 Azure Storage

Azure storage is Azure’s cloud solution for data storage. Within Azure storage are several different products that are available for the customer. One of these are Azure disk storage, which provides the customer with a virtual hard drive. The customer can for example choose a size of the hard drive and which hardware type (e.g. HDD-drive or SSD-drive).

Another product is the Azure blob storage which is a more abstract type of storage compared to the disk storage. The user doesn’t specify specific hardware and size. Azure blob storage can store everything from regular textfiles and JSON files to schemaless storage of structured data. The data file itself is often referred to as a blob.

One important concept with storage is resiliency. A container of data is always stored at a data center located in the region your storage account is assigned to. Data can be very sensitive and must be able to survive a data center crash. The data may also need to be available at all times. Azure storage has four different levels of redundancy which determines how many copies of the data that is stored but also where it is stored (see figure 2). With an increase in resiliency comes a higher fee.
- Locally redundant storage (LRS) - Three copies of any piece of data are stored on a data center located in your region.
- Zone-redundant storage (ZRS) - Three copies of any piece of data are stored in three different data centers.
- Geo-redundant storage (GRS) - In addition to a LRS redundancy, additional three copies are stored in another region (at one data center).
- Geo-zone-redundant storage (GZRS) - In addition to a ZRS redundancy, additional three copies are stored in another region (at one data center).

The LRS data can become inaccessible or even lost if a data center crashes or are outaged in any way. Data stored with ZRS is safe and accessible even if one or two data centers are down, but not if a whole region is down. GRS and GZRS provide guaranties to access the data even if the region is unavailable. If GRS or GZRS are chosen as the redundancy option, the data in another region can only get accessed to read if the primary region is outaged. An additional addition to GRS and GZRS is the read-access (RA) option, which enables reading of the data in another region.

![Illustration of the four different redundancy options available in Azure storage.](image)

### 3.2 Azure Batch

Azure batch is a service that handles and connects resources available for users on Microsoft Azure. The batch account has two compulsory resources which are always used. The first is an Azure storage account and the second are virtual machines (VM). The storage account is set up when creating the Batch account, but the VMs are specified when you want a job to be done. There is possible to include other Azure services like network in the Batch account.

The general infrastructure when using the Batch service is to have a Batch
client, located on a premise, which is the application that communicates with the Batch service and gives directions. It is also possible to monitor the state of the execution from the Batch client. To communicate with the Batch service you can use either direct REST API calls or an Azure Batch API. If one prefers a graphical interface, it is possible to communicate to the Batch service from the Azure portal.

Azure Batch uses virtual machines to execute its workloads. A VM is called a node and multiple nodes form a pool. Batch uses application packages which contain the source code that will be installed and run on all nodes within a pool. These application packages are often uploaded to the Azure storage account beforehand by the user. Which application packages that should be installed on the nodes are specified in the Batch client when creating a pool.

For the user to assign workloads to the Batch client to execute on the pool (nodes) a job has to be created. The job is pointing to a pool, and the job contains tasks. The tasks are constructed to run a program or script with its own input. The input is usually a subset of data to be computed on the pool. The input data can be uploaded beforehand or provided by the Batch client when creating the tasks. With multiple tasks that cover the entire set of input data, an effective and parallel workload can be achieved. Each task is pointing onto a node in the pool and for max efficiency one task can be pointing to its own node. Tasks can be assigned to the same node if the node effectively can handle multiple tasks in parallel.

With multiple nodes available, connected and optimized for parallel programming, Azure Batch can be used for high-performance computing. Azure Batch works especially well for intrinsically parallel problems, which are problems that can be efficiently solved in parallel in separate tasks.

### 3.2.1 General workflow of Azure Batch

A general workload with Azure Batch can look like this:

1. The first step is to upload the files that the nodes will use. This can be files containing data or whole applications such as exe files.
2. The second step is to create a pool of compute nodes. This step includes specifying properties of the type of nodes. A job with tasks is also created.
3. Download files as well as applications from Azure storage. Install the applications that needs to be installed.
4. Use the Azure portal or the Batch client to monitor how the tasks are doing.
5. Upload output from the tasks to Azure storage.
6. Download output from Azure storage.
Figure 3: A general workflow of Azure Batch [9].
4 Objectives and problem formulation

There is an increasing demand for scalability and flexibility in BM System’s products, in terms of computing power demand as well as flexible deployment. Utilizing cloud computing to achieve this demand has been discussed internally at BM System.

The aims of this bachelor’s thesis were therefore:

1. To select one Microsoft Azure service that would suit the demands at BM System.
2. To investigate and evaluate the performance, price and functionality of the selected cloud computing service.

There are several important concepts that were examined and evaluated in this thesis: performance and time analysis, scalability, pricing and log management.

- Performance and time analysis - Some of BM System’s services have requirements and promises to be finished and delivered within one hour. When using a cloud service there are often several steps in the workflow that might take time. The thesis evaluated which steps in the workflow that took the most amount of time (e.g. starting up a service) and if it was possible to optimize the performance of the selected Azure service.

- Scalability - One of the main reasons to use a cloud service for dynamic workloads is the ability to use precisely the amount of hardware that is needed. Nothing more, nothing less. This concept goes hand in hand with the pay-as-you-go concept that Azure provides. Which features are available in the service when it comes to scalability?

- Pricing - The cost of a service is very important. How much does the service cost and are there any options to reduce the cost in any way?

- Log management - Logging is very useful to generate when running an application and crucial when maintaining an entire system. Which services and features for log management are implemented with Azure?

5 Method

5.1 Evaluating which Azure services to use

Microsoft Azure offers plenty of different services, and picking the right service can be difficult. BM System’s workload includes optimization problems that require quite a lot of computing power. Some optimization problems needs to be done within one hour as well. Microsoft Azure has an article that acts like a guideline to which service that would fit the best [10]. It is important to know when choosing a service how you want to integrate an already existing
application. There are two main options. The first is to "shift and lift", which means that you migrate your existing code without any refactoring or cloud optimization. The other option is to refactor the existing code for cloud computing. To refactor and optimize the code for cloud computing is preferable, if you have the time and resources. The second parameter is the category of the workload that will run on Azure (i.e. "high performance computing" or "short lived processes"). Because BM System’s workload includes optimization instances that must finish within one hour, the workload should be categorized as high performance computing. The Azure guide recommends Azure Batch which will be the main focus in this thesis.

It is important to remember that there are other useful computing services that might work as well. Another thing to have in mind is that it is possible to use multiple compute services that cooperate to optimize different workloads.

5.2 Demo application

A demo application was built to communicate with the Azure Batch service. The core of the demo application was implemented to imitate a Batch client used in production by a company, but with an additional module containing several tests and experiments. The application was built in C# version 7.3 with .NET framework 4.7.2. The version of C# and .Net framework was chosen because most of BM System’s current systems are built with it.

5.3 Test and Experiments

Tests were made to achieve data on performance and time on services used by the Batch service.

5.3.1 Performance and time of VM’s

Microsoft Azure has various numbers of different virtual machines to choose from. When choosing which virtual machine that would be the best choice for BM System to use, there are several aspects to take into account in the decision:

- Performance
- CPU/RAM ratio
- Price

Each VM is placed in a category depending on the use case for it. The most commonly used categories are "general purpose", "memory optimized" and "compute optimized". A main difference between these categories is the CPU and RAM ratio.

Azure divides VMs into series depending on the VMs specification. An example of a VM-series is the Av2-series. Within each series are multiple VM sizes which
Table 1: Overview of all the different CPUs used by VMs.

<table>
<thead>
<tr>
<th>CPU</th>
<th>VM-series</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Xeon Platinum 8370C</td>
<td>Av2, Dv3, Ev3, FSv2</td>
</tr>
<tr>
<td>Intel Xeon Platinum 8272CL</td>
<td>Av2, Dv3, Ev3, FSv2</td>
</tr>
<tr>
<td>Intel Xeon 8171M 2.1 GHz</td>
<td>Av2, Dv3, Ev3</td>
</tr>
<tr>
<td>Intel Xeon E5-2673 v3 2.4 GHz</td>
<td>Av2, Dv3, Ev3</td>
</tr>
<tr>
<td>Intel Xeon E5-2673 v4 2.3 GHz</td>
<td>Av2, Dv3</td>
</tr>
<tr>
<td>Intel Xeon Platinum 8168 2.7 GHz</td>
<td>FSv2</td>
</tr>
</tbody>
</table>

Table 2: Overview of the VM sizes used in the performance tests, and their basic specifications.

<table>
<thead>
<tr>
<th>Category</th>
<th>VM-Series</th>
<th>VM-Size</th>
<th>vCPUs</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Purpose</td>
<td>Av2-series</td>
<td>Standard_A4_v2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>General Purpose</td>
<td>Dv3-series</td>
<td>Standard_D4_v3</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Memory Optimized</td>
<td>Ev3-series</td>
<td>Standard_E4_v3</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>Compute Optimized</td>
<td>FSv2-series</td>
<td>Standard_F4s_v2</td>
<td>4</td>
<td>8</td>
</tr>
</tbody>
</table>

The VMs have no single standard CPU, instead there is a pool of different CPUs that each VM size has the chance to get. Table 1 shows which CPU’s each VM size can be assigned to.

To measure the differences in performance between the different categories, two tests were made. Four different VM sizes were used in the tests. Two are in the "general purpose" category, one is in the "memory optimized" category and one in the "compute optimized" category (see Table 2).

The first test measured the time for a VM to start. That is, the time it takes from the start of an allocation of a VM until the VM is runnable and ready for the user to use. The test included steps to create a storage container as well as uploading necessary files to the storage. The test used a meagre 80 KB csv file, and therefore the time to create and upload the input is negligible compared to the total runtime. To get the time it took for the VM to start, the test was using the Batch API to poll the status of the VM. The time between each poll was 50 ms.

The second test was a CPU performance test. This test measured the time for the node to calculate all permutations of every word located in a csv file with 10,400 words. The max length of the words was set to 9 characters. Because of the limited length of each word as well as only using 10,400 words, the RAM available in all VMs will be more than enough for this workload and will not slow down the calculation. Due to its stochastic nature, calculating all permutations
should not effectively utilize the cache memory in the CPU either. Each test was executed 15 times on each virtual machine and the max value as well as the median value was retrieved. The second test also measured the ability to execute code in parallel. One part of the test was using one thread and another was using 4 threads.

5.3.2 Chaining virtual machines

Nodes contained in a pool can only be of one VM size. Usually a workload only require one machine to do the job, but is it possible to have a workflow with several different VM’s that calculates different parts of a workload? Say that the workload consists of one heavy preprocessing part which is parallelizable and requires extensive amount of RAM and one heavy part which is not parallelizable. If one VM would do both of these calculations in serial, there is no way to optimize the hardware for both these different workloads. Either will the VM have no parallelization for the first workload or several vCPU may get idle and RAM unused when the second workload is running. One workaround for this problem is to chain multiple nodes in different pools one after each other. To solve the chaining of nodes, a test was made to simulate the actual workflow. The test used the Azure storage to store the output from the first VM, which then the second VM was using as its input.

5.4 Pricing

Almost all services provided by Microsoft Azure cost money and usually with a pay-as-you-go payment method. Some services that cost are for example virtual machines, storage, logging and network. With the pay-as-you-go concept and various services, there can be difficult to get a good grasp of what the accumulated and final cost might be. Microsoft Azure has a price calculation service that can be used to help estimate the final cost of a month. The difficult part is to estimate how much of a service that will be used. That can for example be how many hours a VM will be used during the month or how many operations (i.e. read operation) that will be performed on a storage account.

To get a good grasp of the pricing two approaches was used. The first was to explore the pricing options for all the services that was likely for BM System to use. This was done using the Azure pricing calculator. It is impossible to get an absolute number of the final cost, with the pay-as-you-go approach that Azure uses. It is at least possible to get a fair estimation. There is also possible to compare the costs between different resources within a service (i.e. storage types or VM’s). The second approach was to use the services and empirically examine the cost. With this approach it is possible to find additional fees that might be overlooked.
5.5 Logging

Microsoft Azure has several features that handles logging. The logging operates at two different levels.

- **Subscription level** - Azure saves logs when for example a resource is modified or a virtual machine is started [15]. These kinds of logs are called Activity logs.

- **Batch account level** - Logs made at the Batch account level include resources within the Batch account and are often more detailed than at the subscription level [16]. There are two types of logs collected at the Batch account level:
  - Metrics
  - Resource logs

 Metrics are logs that every minute automatically saves state of resources used by the Batch account. These logs contains data of how many tasks that are running right now or how many idle nodes that exist. Resource logs is updated when certain events complete. Such events include starting of a new node or task gets completed. Both metrics and resource logs can be stored on the storage account associated with the Batch account.

 If the consumer wants more control of when to log data, a feature called Application Insight can be used. Application Insight works by the developer manually entering the source code where you want to log certain data. The logs collected by the Application Insight can be looked at within the Azure portal.

 The logging features mentioned earlier were evaluated using Azure portal on the logs gathered from the performance tests.

6 Result

6.1 Evaluation of virtual machines

The median of all runs was calculated instead of the average, to minimize the risk of outliers affecting the result. Some of the optimizations BM System provides needs to be done within one hour. With this promise it is super important that the VMs are rather stable in their performances. Since this is important, the maximum time in the tests were compared.

The startup test showed that there is a difference in time it takes for different VM sizes to start. The slowest in both the median and the max time was the Standard_A4_v2 and the fastest in both measurements was the Standard_F4s_v2. The Standard_A4_v2 was severely slower to start than the Standard_F4s_v2. The max time of Standard_A4_v2 was 2.02 of the time it took for the Standard_F4s_v2. The Standard_D4_v3, the Standard_E4_v3 and the Standard_F4s_v2 had all fairly similar times.
The performance tests, which calculated all permutations of a list of words had results that correlates with the startup test. In the test with 1 thread, the Standard\textsubscript{A4,v2} took the most time and the Standard\textsubscript{F4s,v2} the least. The Standard\textsubscript{A4,v2} was extremely slow compared to the rest and had execution time 3.27 times longer than the Standard\textsubscript{F4s,v2}. (See figure 5). In the test with 4 threads the result was a bit more even across all VM’s, but the Standard\textsubscript{A4,v2} was still a lot slower than the rest (see figure 6).

Figure 4: Showing the time it took for a VM to start up.
Figure 5: Shows the time for all VMs to execute the permutation test by only using 1 thread.

Figure 6: The time it took for all VMs to execute the permutation test with 4 threads running in parallel.
With the results of Standard_A4_v2 taking so long compared to the rest of the three VMs, and the Azure documentation provided no clear reason why that would be, a test to see why this might happen was made. The test used was the performance test done earlier, but now with additional logging information. The additional logging was the CPU time, that is the aggregated time all CPU cores allocated was executing. The logging also included the current CPU speed, measured in GHz, as well as the name of the actual vCPU that was running on the VM. This test was like the previous test run 15 times on 1 thread as well as 4 threads. To measure the CPU time, a 'process' class located in the namespace System.Diagnostics was used.

The result showed that all of the test’s 120 runs used the same CPU model. The CPU model every node had was the Intel Xeon Platinum 8272CL. The CPU speed was also constant during and across all runs, with a speed of 2.6 GHz. The CPU times were also linear. When the faster Standard_F4s_v2 was done with its calculation, the CPU time was the same as with the slower Standard_A4_v2, which at that time only had around 25% of all calculations complete.

### 6.2 Auto Scaling

Azure Batch has a feature called "auto scale" which automatically allocates virtual machines or nodes based on the needs. Batch auto scale has two properties set by the user. The first one defines how often auto scaling should be applied. This value cannot be lower than every five minutes. The second is a formula that when evaluated returns the number of nodes that should be allocated. The formula is provided as a string with statements declared in it. These statements can include user defined variables and Batch service defined variables. With the Batch service variables, the formula can get values like the amount of task in queue at that moment or the average amount of CPU usage in percent for the last 3 minutes. For example can a formula adjust the node count to be the minimum number of tasks waiting to run plus the number of already running tasks plus 1 and a maximum amount of VM’s. See listing 1

```csharp
pool.AutoScaleFormula = "maxNumberOfVMs = 25;" +
    "$tasksSample = $ActiveTasks.GetSample(180 * TimeInterval_Second)" +
    "$RunningTasks.GetSample(180 * TimeInterval_Second);" +
    "$TargetDedicatedNodes = min(maxNumberOfVms, $tasksSample + 1);"
```

Listing 1: Example of an auto scale formula

A test was made with the formula in listing 1 as well as a evaluate interval of 5 minutes. The test was made to verify that it actually evaluates every 5 minutes and not with a bigger interval.

1 System.Diagnostics is a namespace built into .NET Framework.
The test was successful since the evaluation of the formula was made exactly every 5 minutes. It was also possible to change the formula during a running pool.

### 6.3 Chaining nodes in serial

When constructing the test for chaining nodes, one problem occurred due to security reasons, the pools cannot communicate and send data to each other. To overcome this problem it is possible to use the storage as an intermediator where data is stored for the next node to process. Since the nodes cannot communicate, the Batch client has to coordinate everything. The workflow now includes the Batch client to poll or get an event trigger when the first node has finished and the intermediate data is stored on the storage. Then the Batch client starts the next node which gets its data from the storage. See figure 7. The test was successful as the solution tested in this thesis to chain nodes was shown to be successful.

![Figure 7: Illustration of how a workflow could look like when chaining two different VM's.](image-url)
6.4 Pricing

6.4.1 Costs of different virtual machines

The cost of a VM is essential when it comes to the Batch service. The Azure pricing calculator was used to compare the price of the four VMs. The cost per vCPU that you allocate for each VM is pretty even. The most expensive VM is the VM within the Ev3-series. It costs 1,12 kr per vCPU and hour. The cheapest VM is the Av2-series with an hourly cost of 0,86 kr per hour. The most cost-effective VM is not the Av2 however. When combining the median time of the "Time until runnable test" as well as the "CPU performance test" and multiply with the hourly cost of the respective VM, the FSv2 VM is the most cost-effective. See Figure 9. The tests show that the FSv2-series of VMs are both the best performing VM-series as well as the most cost-effective VM-series. It is important that to keep in mind that the tests omit an eventual requirement of high amount of RAM. An workload which require high amount of RAM might benefit to use the Dv3-serie or even the Ev3-serie if the required RAM is very high.

![Price per vCPU / hour](image)

Figure 8: The cost for using a VM for one hour. The price is per vCPU used.
Figure 9: The cost of executing the performance test. The cost is calculated by multiplying the time it took for the VM to complete the test with the price per hour.

Figure 10: The price of using a VM running with Windows compared to Linux. The price is per vCPU.
6.4.2 Cost of storage

The storage account, used together with the Batch service, is using a pay-as-you-go approach as well. The storage service has four different access tiers which determine the cost when using the storage. The four access tiers are Premium, Hot, Cool and Archive. The differences between these are the cost of the storage of data as well as the operations used on the data. The Archive tier is only used when data is stored for many years and not regularly interacted with. The cost per GB stored is very low but there is no way of accessing the data in real time. Instead the blob of data has to change tier to the Hot or Cold tier before it is possible to access the data. Therefore Archive is not appropriate to use together with the Batch service. Premium, Hot and Cold can all be used with Batch and they have all the same performance when it comes to latency and throughput. The cost of storing data is highest with the Premium tier and the Cold is cheapest. The cost of writing or reading data to the storage is inverted with Cold tier being the most expensive and Premium being the cheapest. See figure 11. The price for writing and reading data from the storage is defined per 10,000 operations. One operation is limited to 4 MB which means a file of 1 GB are 250 operations. Operations can also be function calls using the Batch- or REST API to communicate with the storage.

Figure 11: Compares price of different operations. Price per GB stored is storing during one month. Price of writing and reading data is per 10,000 operations. All prices are using the LRS redundancy setting.
6.4.3 Overlooked costs

In addition to the Azure pricing calculator, the prices from all tests and services used during the performances tests were examined to find out which services that costs money. During the tests, there were especially three services which accounted for the majority of the cost.

- Virtual Machines
- Storage
- Load Balancing

Load balancing is something that may get overlooked when doing a budget for Azure services. The load balancer is a mandatory part of Batch. This is because the VM’s that Batch creates are required to be part in a network, due to the connection that’s available to the Batch service or the Batch Client. If a pool of nodes doesn’t get specified to a particular network, a default one will be used. In either way, a fee of the networks load balancer will be taken. The role of a load balancer within Azure Batch is to handle all requests that being delivered from as well as to the VMs within each pool. The price of the load balancer is 0,25 kr per hour as well as 0,05 kr per GB of data transferred. After the performance test, the cost of the load balancer was around 12.5% of the total cost.

7 Discussion

7.1 Selecting a virtual machine

The results achieved from the startup test and performance test show huge differences. The huge differences between the A4_v2 versus the other VMs are not shown in the specifications provided by Microsoft Azure. On the specification website for A4_v2, there exists a sentence that might explain the result:

"The size is throttled to offer consistent processor performance for the running instance, regardless of the hardware it is deployed on."

It is very difficult to see what this really means, but the tests show that there might be a throttle in performance to just 30% of the original hardware performance. After this test the VM-series Av2 is not something that suits the needs for BM System.

The variance between the maximum times and the median time is quite small (around 30 seconds). Are these tests enough to confidently show that the VM’s performance are stable? I would say not really. The total number of tests between all VM’s is 120. One needs to have in mind that the actual CPU of the VM is automatically selected from a set of pre-defined CPU’s. These tests don’t know which CPU was used. Naively calculated the number of test on each CPU is 120 divided by the number of different CPU’s which is 120/6 = 20.
The result when comparing D4_v3, E4_v3 and F4s_v2 is pretty similar. The F4s_v2 provides a little bit better performance in all tests and might be the primary choice. One reason why F4s_v2 is the fastest VM in the test may be that the CPU has a boost functionality which increases the clock speed for a moment. The reason why F4s_v2 is not the most expensive of the VMs tested might be the relatively low amount of RAM that comes with each vCPU.

The test that aimed to answer why there are such different time difference between VMs didn’t give any direct useful information. It was interesting that all 120 runs used the same kind of CPU. It isn’t possible to know if the runs in the performance tests had the same CPU model, but it is likely that they did. Since the maximum execution time is a good metric to compare, I would say that additional tests have to be made with CPUs that confirmed aren’t the Intel Xeon Platinum 8272CL.

7.2 Storage account

As the pricing of writing and reading data is defined by the number of operations and not directly expressed in GB transferred, it may be difficult to know how much it will cost in the end. The Cool tier should be used together with the Batch service if the data that you upload or generate will be stored for a longer period of time. If that is not the case the Premium or Hot tiers should be used. Something to have in mind when choosing storage tiers for the Batch service is that it is not only temporary or intermediate data uploaded before or generated during the execution of a workload, but also the application that will be run on all nodes as well as eventual logging generated by Batch. Hot might be the a good middle ground between Premium and Cool.

7.3 Chaining of virtual machines

Although the chaining test was a success, there are still some issues. The first issue is the time that is required to store the data on the storage and then retrieve it again. All these transactions do cost some extra money as well. With a premium tier on the storage account, the money will be relative small though. The problem of additional time taken is even bigger when including the startup time of nodes. The startup test showed that the max time for a Standard_F4s_v2 took 2 minutes and 7 seconds which could be a deal breaker if the total time limit is one hour. One solution to this problem is to always have a node ready beforehand to start executing, but idle nodes cost as much as running nodes, which will increase the total cost. One more advanced solution could be to have the node notify the Batch client when it is almost done, so it can create the node to be ready when the right time comes. I don’t think there is a built in support for these kind of calls, which means that an own implementation is required. One option could be to use a web hook to notify the Batch client.
7.4 Evaluation of logging features

The tests for Metrics and Resource logs were a bit underwhelming. The data received from these logs was hard to read without coding your own parser. You also had to manually download the JSON files.

Inside Azure portal, Batch has a feature where it is possible to look at the data in a graphical interface. The quality of the results was very poor, with a lot of bugs that for example stated that all nodes created by Batch are low priority nodes (i.e. nodes that are cheaper, but comes with a risk of being preempted) and regular nodes at the same time.

8 Conclusion

The study showed that Microsoft Azure together with the Azure Batch service may be a solution for BM System. The VMs belonging to the FSv2-series got the best result in both the performance tests as well as the cost efficiency test and could be the VM to choose. Because of huge difference in price by using a VM containing Windows as the operating system compared to Linux, I would recommend Linux if the applications executed on it are compatible. If that’s not case, it might even be worth invest into a refactor of the application.

For the storage account, a Block blob storage with either Premium or Hot tier would be the best choice depending on how much data that will be stored long term.

The study also showed that the automatic scaler of Azure Batch reliably evaluates the number of nodes every 5 minutes. It is also possible to chain VMs one after another to use a specific VM for each kinds of workload.

9 Future Work

There are several of subjects that could be evaluated further:

- Security - There are several security measures that should be investigated further and then implemented if used with Azure Batch. In the demo application, a shared access key was used to access the Batch account, and further more, this key was stored in a JSON file together with the source code. This should absolutely not be done when Azure Batch is used in production. One concept to further investigate is the use of Azure Key Vault, which is a way to securely store the access keys. Another concept that can be used is the use of Azure Active Directory, which is a way to explicitly assign access to the services for each client.
• Azure spot nodes - Azure offers virtual machines at a huge discount with the catch that they can be preempted and only used if there are VM's not used at that moment. For workloads with hourly deadlines these kind of spot nodes are not really an alternative, but for other workloads with not the time limit these VM's might be an option.

• High-performance computing (HPC) and GPU - Azure offers really fast VM's optimized for really huge workloads as well as VM's with access to GPU's which could be explored further if there is a need for it. Note that the price for these VM's is really high tho.

• Application Insights - To get more control of what and when to generate logs, application insights can be used. It works by manually in the source code add what to log. The data is then transferred to the service Azure Monitor to be examined.

• Events and alerts through logs - There are features in Azure that either calls an event or an alert when a certain log value is received. It could be very handy to get notification when something bad happens.

• Because VMs located in different pools cannot directly communicate, an Azure network might be used to make chaining easier and more effective to implement than the solution evaluated in this thesis.

References


