Optimizing Service Instance Configuration for Efficient System Resource Use

Last generated: December 17, 2025

Table of Contents

- **Test Study**

Introduction

ArcGIS services expose capabilities and make maps and their associated data accessible over the web. A key factor in optimal map service configuration is choosing the appropriate number of service instances, each of which operates as a separate ArcSOC process dedicated to running a map service. An ArcSOC (ArcGIS Server Object Container) is an ArcGIS Server process that runs a GIS service, where each ArcSOC handles one request at a time.

While hosted and shared service types are automatically managed by ArcGIS, map service instance configuration of dedicated services is an often neglected aspect of system optimization. Services and system configuration need to be tested and revisited regularly as services are added or usage patterns change. Otherwise, you might find unacceptable system performance and poor end user experience and efficiency over time.

This test study explores two concepts using dedicated instances with equal minimum/maximum values:

1. How an under-resourced database impacts broader system usability
2. How the ratio of ArcSOCs to vCPUs (the number of ArcSOCs configured per vCPU on the ArcGIS Server instance) impacts the system and end user experience

Keep in mind that a dedicated instance that is not running requires some time to start when it receives a request. Our system tests are focused on user experience, so we don’t want to incur the delay of waiting for an ArcSOC to startup. Therefore, for our purposes, all service instances are configured by setting maximum equal to minimum.

Note:

This test study is not intended to recommend a specific ratio of CPU to service instances for every system. Rather, it shows that organizations must perform testing to determine the optimal configuration for their system in a way that balances performance with infrastructure costs. Learn more about how to Monitor system performance.

This study tested real-world workflows against a Network Information Management System hosted in Amazon Web Services (AWS) cloud infrastructure using AWS EC2 instances.

Tested workflows

To ensure the test study provides meaningful results, the workflows need to represent real user experiences, and the actual steps that users will take in interacting with the system. The workflows used in this test study represent some of the foundational activities required to maintain and access an as-built electric network.

The contents of the workflows were defined by experienced staff, along with Esri customer feedback to identify the specific steps, sequencing and type of activities involved in each workflow. The following key workflows were run manually against the system under load to capture user experience and overall performance:

1. Create a new service with existing feature– provide service from existing transformer
2. Create new service from new feature – provide service with new pole and transformer
3. Update asset – move an asset or update attributes
4. Load management – redirect load from one circuit to another
5. Phase management – move a service to a different phase
6. Electric tracing – upstream protective trace and downstream customer trace
7. View assets – search and view assets and attributes
8. Summarize assets – identify dirty feeders, counts of new features

You can read more about these workflows in the related system test studies.

Software

The system capabilities are delivered through the following software, deployed, and tested as part of this test study, at the listed versions with all available patches applied:

- ArcGIS Pro 3.3 (latest version here)
- ArcGIS Enterprise 11.3 (latest version here)
- ArcGIS License Manager 2024.0 (License Manager is being deprecated, and is not needed for versions of ArcGIS Enterprise 11.4 and later.)
- ArcGIS Monitor 2023 (latest version here)
- ArcGIS Online

- PostgreSQL v14.6, with the STGeometry spatial data type

ArcSOC optimization

Users expect quick response times when they interact with ArcGIS services. A key contributing factor is the ArcSOC configuration (or service instances), which can be optimized in a few ways:

- Instance type – either a dedicated or shared instance pool
- Service time outs – maximum time a client can use a service or wait for a service
- Number of instances – minimum and maximum number of instances per machine
- Ratio of service instances to vCPU cores

Note:

Optimizing your service instance configuration is not a one-time job. Usage patterns of your services can change over time, so balancing resources is an ongoing process. As an example, see test observations from Evaluating impact of adding mobile capabilities to a foundational network information management system.

In general, adequate ArcSOC processes are required to handle the load your services receive, and adequate server resources are required to support a given number of service instances. Because each busy ArcSOC requires an available vCPU, allocating too many ArcSOCs per vCPU can cause unacceptable wait times when vCPUs are busy. Further, too many ArcSOCs can also lead to excessive memory utilization. This is because each ArcSOC consumes memory related to your data and workflows. Conversely, allocating more server resources than necessary leads to unutilized capacity and unnecessary expenses.

Putting that all together, this means the ratio of running ArcSOCs to vCPU should be high enough that there are as many service instances as needed to support end users’ workflows, without exceeding acceptable resource utilization thresholds. A good general practice is that normal operation shouldn’t incur ArcSOC spin up wait time. For business-critical services that require predicable performance, consider setting the instance minimum and maximum to the same value.

The optimal ratio of ArcSOCs : vCPU will depend on your specific system and the work it performs. As a result, you can determine your system’s optimal ratio only through proper testing and observation practices. This test study looks at approaches to balance service instances with compute resources to help you get the best performance with your available resources.

- See ArcSOC availability and optimization in the Esri Community.

Test results on database resource impacts

In our first set of tests, we looked at the impact of database resources on the system's ability to handle load, even with ample GIS server resources. This is to show how impactful the database is across the entire system. In other words, if your users are experiencing long wait times, the issue may not necessarily solely reside within the ArcGIS Server tier.

Test methods and results

We conducted two tests to compare how database resources impact the system performance, even with enough server resources:

- first using a small database virtual machine instance with 8 vCPU
- then with a larger database virtual machine instance with 16 vCPU

We held all other aspects of the system constant, with a 1:1 ArcSOC configuration. In other words, one ArcSOC configured per vCPU on the ArcGIS Server instance, where the number of ArcSOCs and vCPU are equal. We captured and monitored performance metrics like ArcSOC use and availability, service wait times, system resource utilization, and error rates to evaluate each configuration. Tests were performed at 8 times (8x) the design load of the original system test study, and the ArcGIS Enterprise server resources were cut in half (compared to our previous test studies) to ensure there was enough load to impact the system.

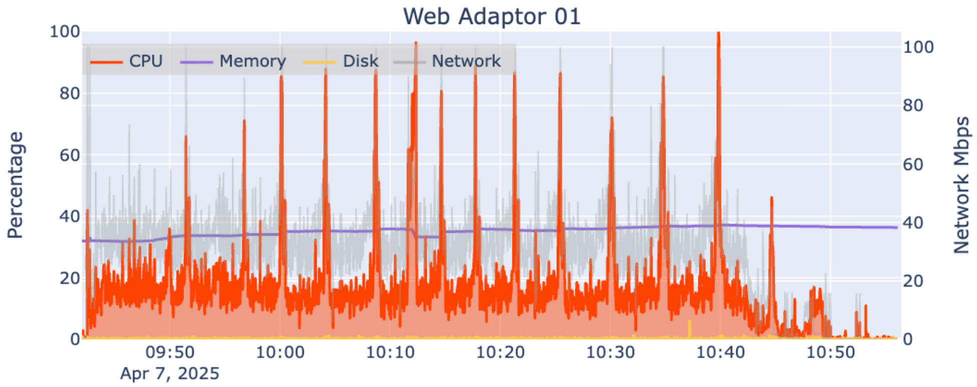
Because ArcGIS is a multi-tier system, tests were conducted across client, service, and data storage tiers, as well as the underlying infrastructure itself. In this test study, JMeter was used to simulate the user workflows and measure system performance under different loads.

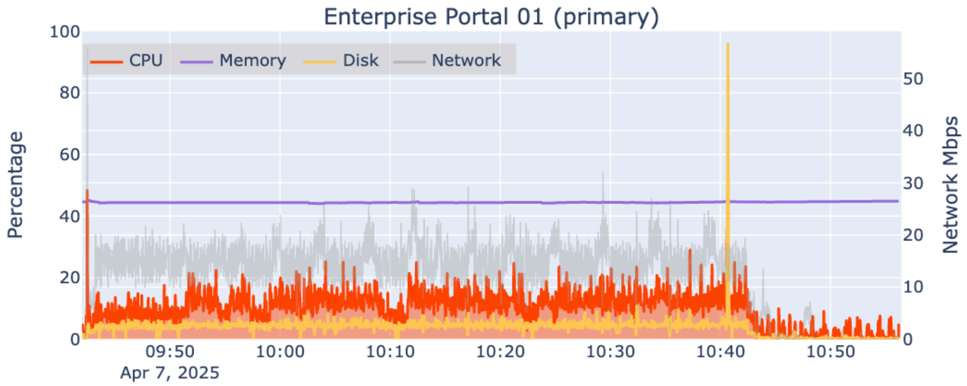
Load test 1 : small database instance – 8 vCPU

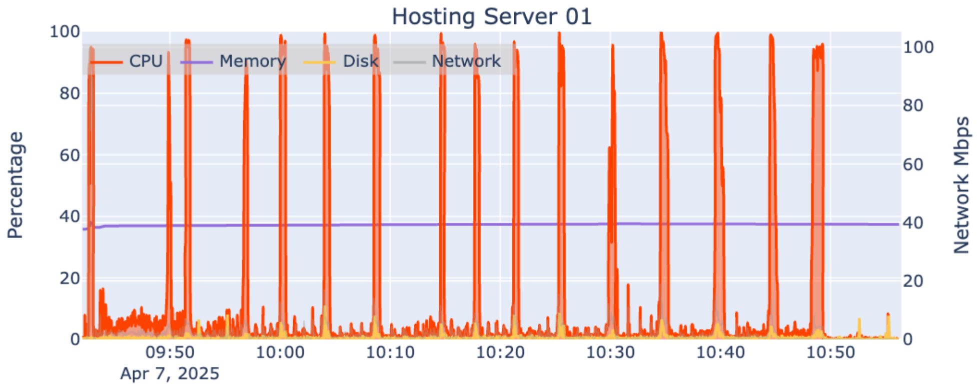
This run was performed with 8 vCPU available on the PostgreSQL instance to observe system impact on a scaled down database server. The ArcGIS GIS Server (UN Server) was also provisioned with 8 vCPUs. There are six instance resource charts below showing resource utilization across the system and one chart showing concurrent requests. In each of the resource charts, the orange lines represent % CPU utilization, the gold lines represent % disk usage, and the purple line is % memory utilization.

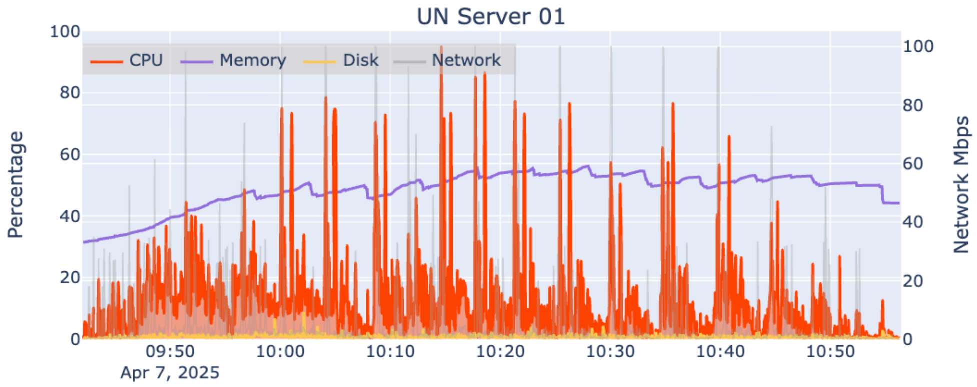
In the diagram below, you can see that the PostgreSQl DB has the CPU running at 100% and the hosting server CPU has frequent spikes to 100%. This is from the high number of view requests associated with the workflows at the 8x design load.

The bottom chart shows concurrent requests, as measured from JMeter logs. Notice the red line, which represents concurrent view requests, trends up to 538 as the test runs. This indicates that requests are not closing. In later charts, you will see this line moving steadily up and down, indicating that the system is responding and requests are closing quickly enough to handle the load.


Web Adaptor 01
CPU
Memory
Disk
Network
Percentage
0
20
40
60
80
100
Network Mbps
0
20
40
60
80
100
09:50
Apr 7, 2025
10:00
10:10
10:20
10:30
10:40
10:50


Enterprise Portal 01 (primary)
CPU
Memory
Disk
Network
Percentage
0
20
40
60
80
100
Network Mbps
0
10
20
30
40
50
09:50
Apr 7, 2025
10:00
10:10
10:20
10:30
10:40
10:50

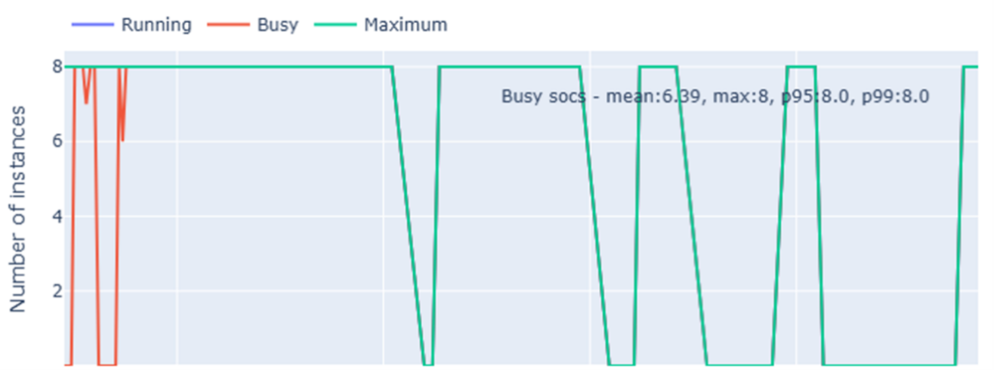

Hosting Server 01
CPU
Memory
Disk
Network
Percentage
0
20
40
60
80
100
Network Mbps
0
20
40
60
80
100
09:50
Apr 7, 2025
10:00
10:10
10:20
10:30
10:40
10:50


UN Server 01
CPU
Memory
Disk
Network
Percentage
0
20
40
60
80
100
Network Mbps
0
20
40
60
80
100
09:50
Apr 7, 2025
10:00
10:10
10:20
10:30
10:40
10:50

Relational Data Store 01 (primary)
100

This configuration did not support the load because the database server was under resourced, as seen by the amount of orange (CPU utilization) in the PostgreSQL database chart, the spikes in the hosting server CPU, and the number of concurrent requests.

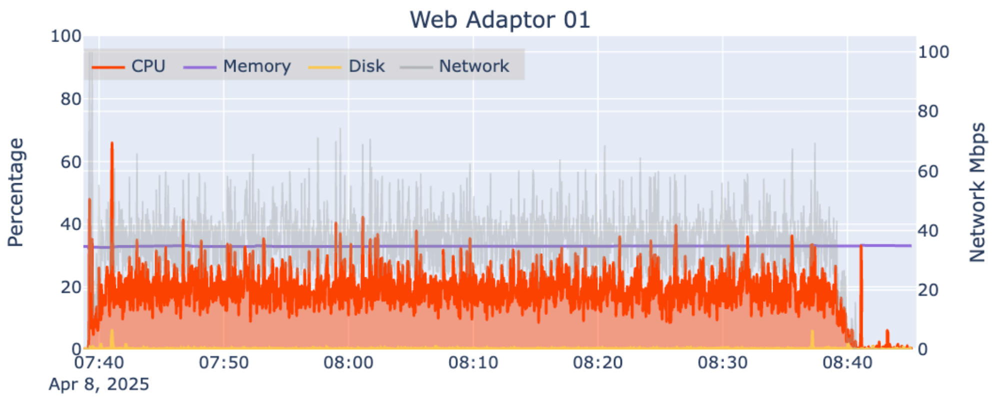
To further reinforce this claim, the chart below represents ArcSOC utilization on the hosting server as captured by the Soccer (ArcSOC Monitoring) utility, which is running on a remote machine. The red line shows busy ArcSOCs at 100% (8), likely attributed to the overloaded database. ArcSOCs are held (busy) while they wait for the database to respond. In fact, the ArcSOCs were so busy that Soccer could not track their state, as illustrated by the incomplete red line and the sudden drops in maximum (green line).

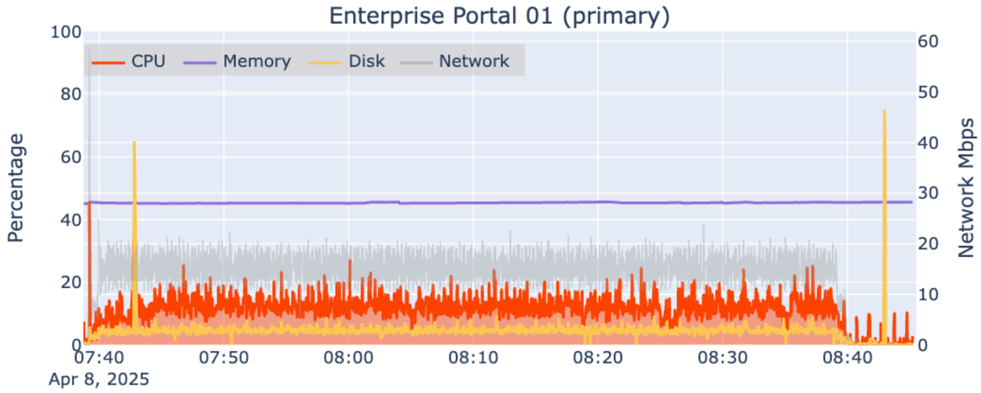


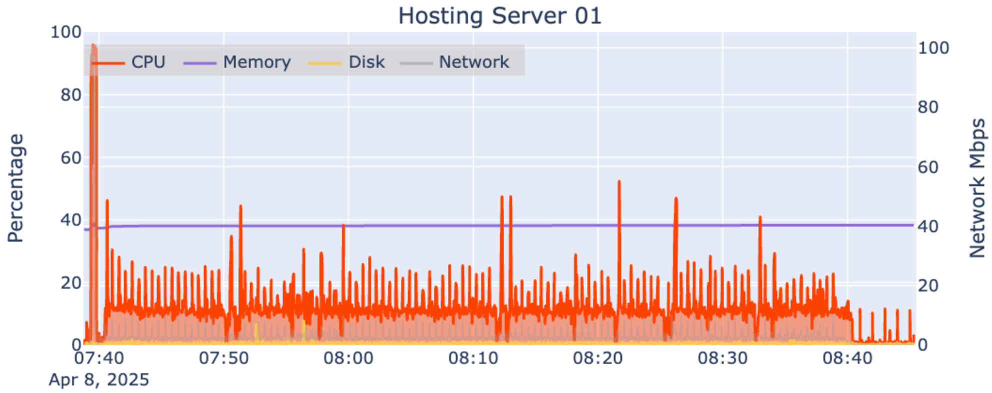
Load test 2 : large database instance – 16 vCPU

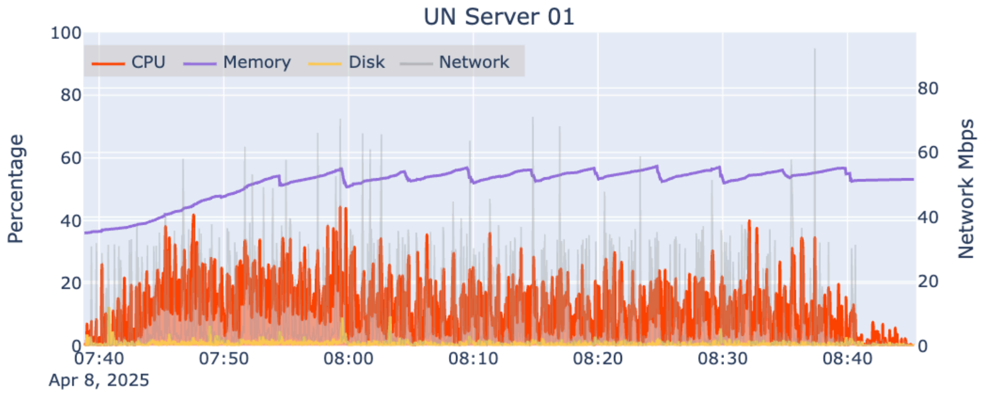
In the second test, we doubled the PostgreSQL database instance to 16 vCPU to observe possible differences from the first test. The ArcGIS GIS Servers remained provisioned with 8 vCPUs each. As in the previous diagram, percent utilization for CPU is orange, disk is gold, and memory is purple. Notice that, minus a few spikes, all servers are generally running below 60% CPU, disk, and memory.

The concurrent request chart shows concurrent view requests averaging 36, with a few spikes. The open requests are not trending up as in the previous chart, indicating this system is handling the load.


Web Adaptor 01

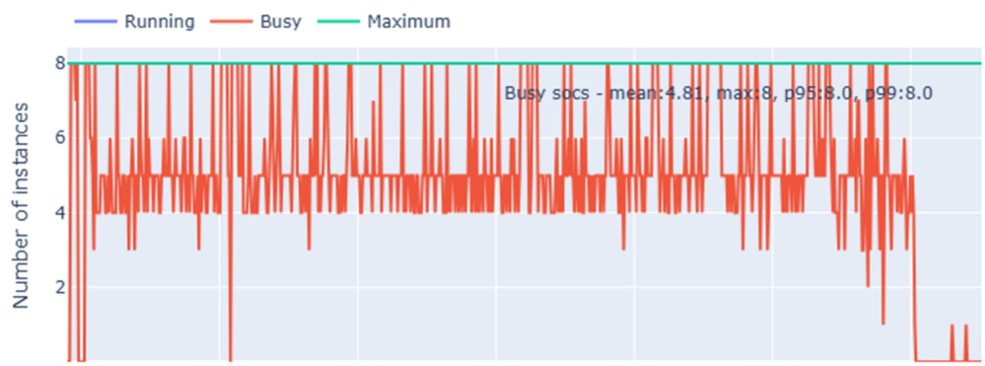

Enterprise Portal 01 (primary)


Hosting Server 01


UN Server 01

Relational Data Store 01 (primary)

The ArcSOC chart below shows that ArcSOCs on the hosting server are busy, but the overall system response is good. Even though 99% (p99) of the usage is 8 socs or less, the average is 4.81. Later we'll look at user experience to see if the system enables people to work efficiently.

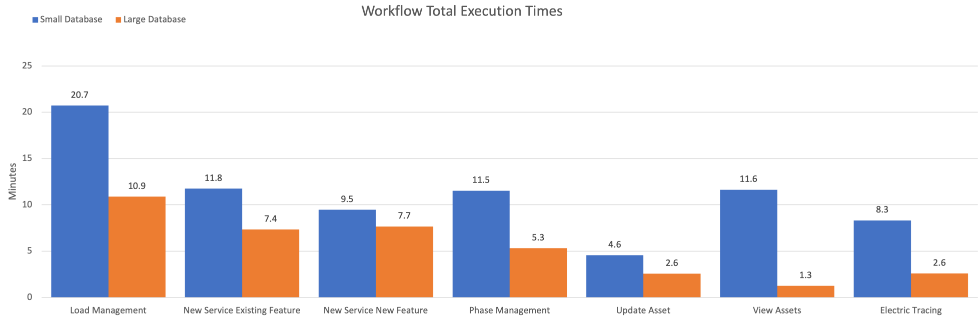


User experience

In addition to overall system utilization and performance, the increased resources available on the database instance significantly improved the end user's ability to complete their work efficiently. This test study evaluated end user efficiency by observing workflow execution times - how long it takes a user to complete workflow's steps, as well as workflow step execution times - how long it took to complete a key step within a workflow.

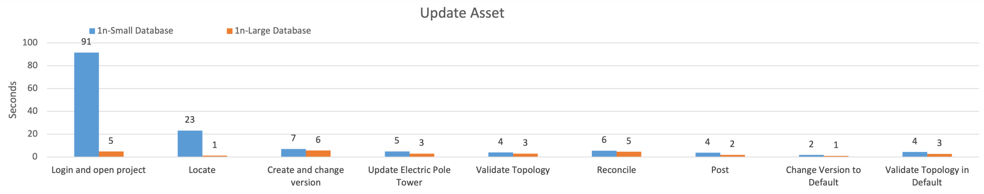
Note:

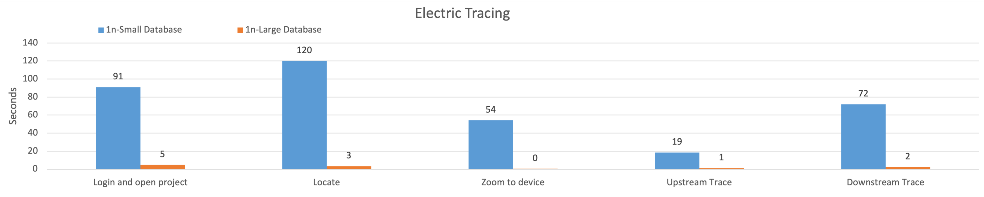
The user experience is the ultimate measurement in these test studies. We have seen throughout testing that even when the system seems to be performing within normal parameters, aspects like network latency, GPU implementation, map instance misconfiguration, etc. can negatively impact end users. Focus on the end users to improve your return on investment.



In the chart above you can see how increased system resource saturation in the small database test run results in longer workflow execution times, which result in an overall worse end-user experience. The overall execution time for all workflows increased significantly with an improperly resourced database as compared to a properly resourced one, even when ArcGIS Server was well-resourced. In particular, the View Assets workflow saw a dramatic decrease in workflow execution time with a properly sized enterprise geodatabase.

In addition to workflow execution time, we can look more deeply at how database resources impact the duration of specific editing workflow steps. However, it does show the vast difference in duration to open a project and locate an asset in the Update Asset workflow. Similarly, Electric Tracing showed a significant increase across all workflow steps. This pattern continued for all workflows captured.





Test results on ArcSOC configuration impacts

In addition to an assessment of database resources, the second set of tests was focused on identifying the optimal ArcSOC configuration to support an 8x design load on our system and its workflows, assuming sufficient database resources.

Test methods and results

Three tests were performed with different ratios of ArcSOCs : vCPU:

- 2:1, or two ArcSOCs per vCPU on the ArcGIS Server instances
- 3:1, or three ArcSOCs per vCPU on the ArcGIS Server instances
- 4:1, or four ArcSOCs per vCPU on the ArcGIS Server instances

We also performed this test with a 1:1 ArcSOC per vCPU ratio, which you can see in the large database instance size test as described in the Impact assessment of database resources.

We ran several load tests, systematically varying the ratio of ArcSOC instances to vCPUs to observe and measure the performance and user experience impacts. All other aspects of the system were held constant to achieve meaningful results.

Performance metrics like ArcSOC use and availability, service wait times, system resource utilization, and error rates were monitored to evaluate each configuration.

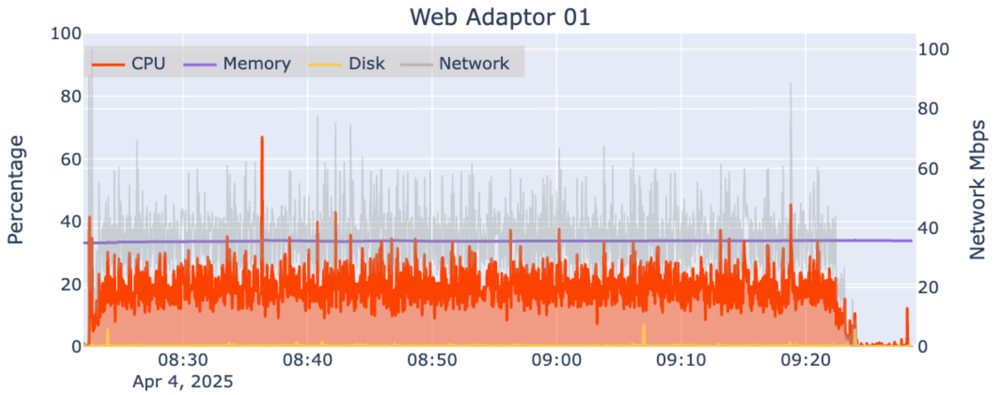
Tests were performed at 8 times (8x) the design load of the original system test study and the ArcGIS Enterprise server resources were cut in half to ensure there was enough load to impact the system. JMeter was used to simulate the user workflows and measure system performance under different loads.

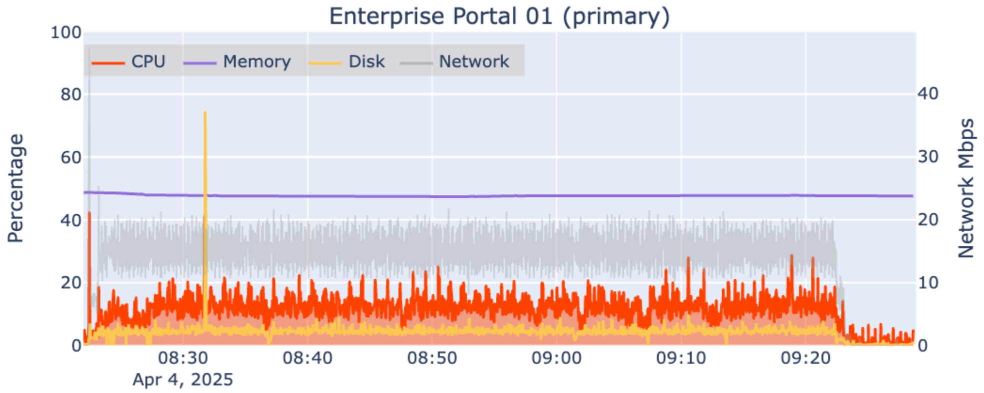
Because ArcGIS is a multi-tier system, tests were conducted across client, service, and data storage tiers, as well as the underlying infrastructure itself. In this test study, JMeter was used to simulate the user workflows and measure system performance under different loads.

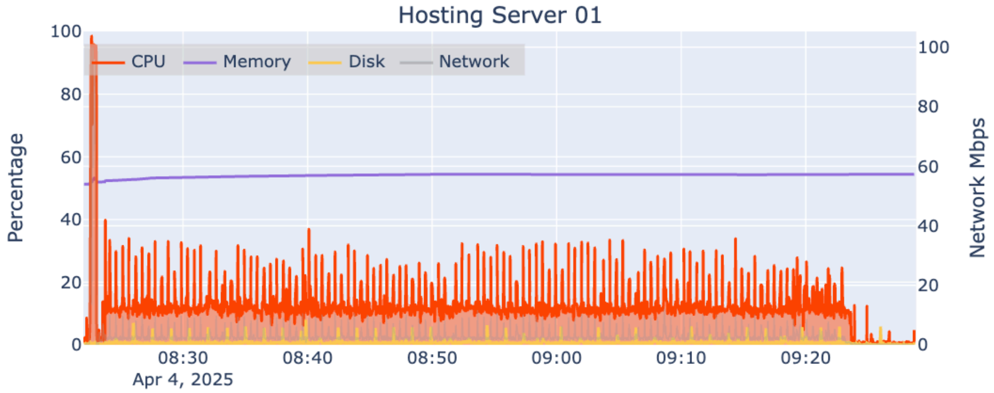
2:1 ArcSOCs : vCPU ratio

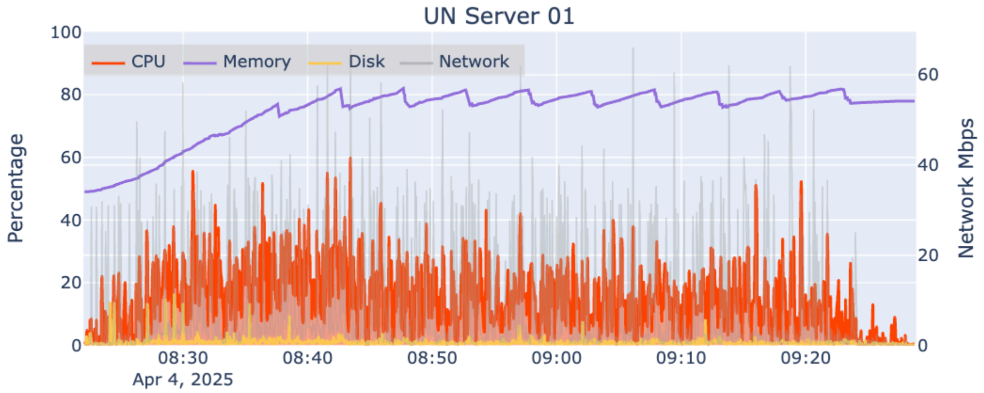
In this run, we configured two ArcSOCs per vCPU on the ArcGIS Server instances. In this case, 16 running ArcSOCs to 8 vCPUs. As in the previous charts, percent utilization of CPU is orange, disk is gold, and memory is purple.

In the chart below, CPU for all machines is generally less than 60%. However, you can see that memory utilization on the UN GIS Server peaks at over 80%. This is due to the additional running ArcSOCs as compared to a 1:1 ratio. Services on the UN GIS Server enable versioned database editing. While the system appears to be running smoothly, memory will need to be monitored closely to avoid problems. The concurrent request chart shows concurrent view requests (red) steadily opening and closing, while averaging 35.


Web Adaptor 01
CPU
Memory
Disk
Network
Percentage
100
80
60
40
20
0
Network Mbps
100
80
60
40
20
0
08:30
Apr 4, 2025
08:40
08:50
09:00
09:10
09:20

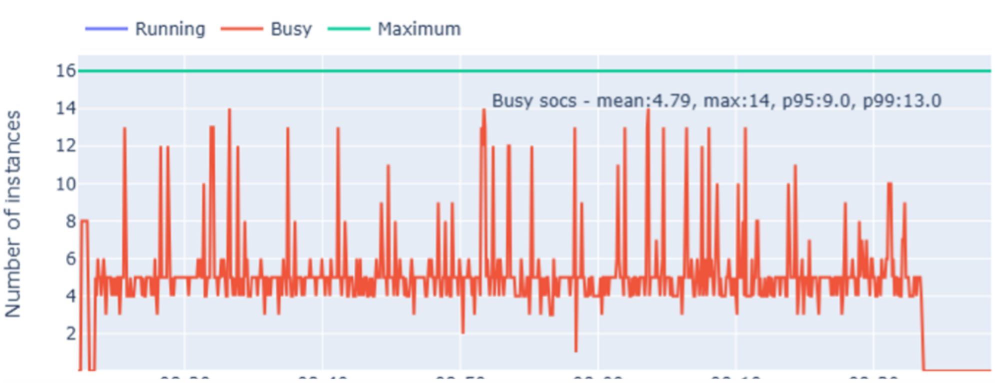

Enterprise Portal 01 (primary)
CPU
Memory
Disk
Network
Percentage
100
80
60
40
20
0
Network Mbps
40
30
20
10
0
08:30
Apr 4, 2025
08:40
08:50
09:00
09:10
09:20


Hosting Server 01
CPU
Memory
Disk
Network
Percentage
100
80
60
40
20
0
Network Mbps
100
80
60
40
20
0
08:30
Apr 4, 2025
08:40
08:50
09:00
09:10
09:20


UN Server 01
CPU
Memory
Disk
Network
Percentage
100
80
60
40
20
0
Network Mbps
60
40
20
0
08:30
Apr 4, 2025
08:40
08:50
09:00
09:10
09:20

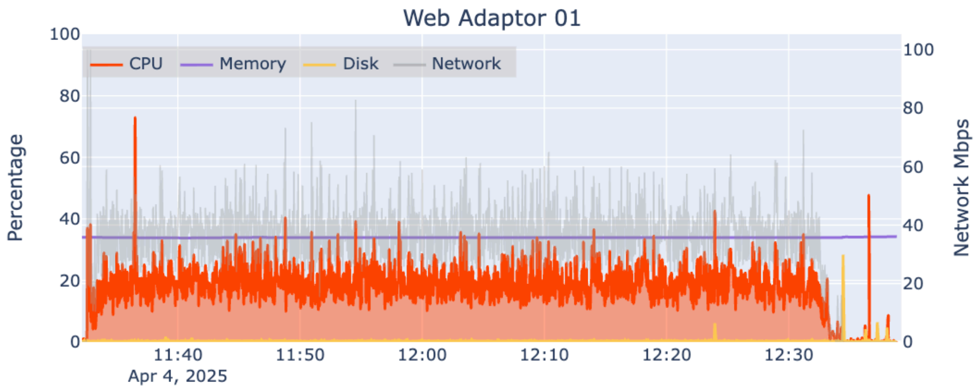
Relational Data Store 01 (primary)
100

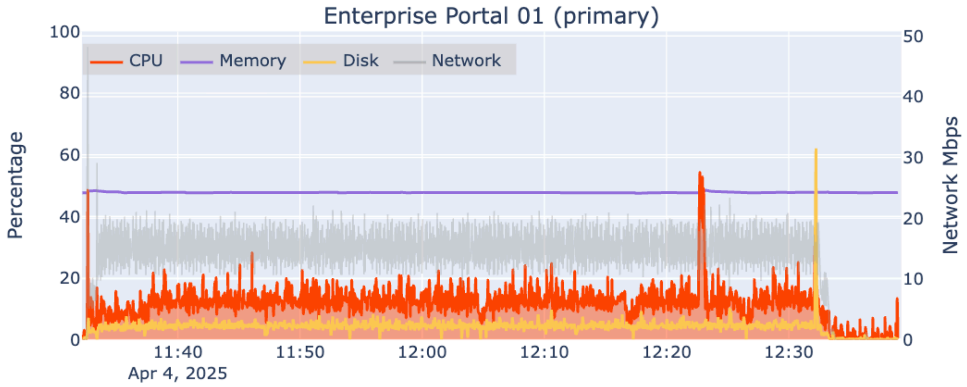
The chart below shows ArcSOC usage for the Hosting Server, where 16 ArcSOCs are running (the blue line is covered by the green line), the maximum in use (busy) is 14. The UN GIS Server (not shown) had a maximum of 7 busy ArcSOCs, so it did not take advantage of the additional service instances, where 16 ArcSOCs were running, but were mostly idle. Because this server showed memory utilization in excess of 80%, reducing the service instances to min/max 8 on the UN GIS Server might relieve some of the memory pressure and make this system optimal for these workflows and loads. Any change in workflows or loads could impact the balance.

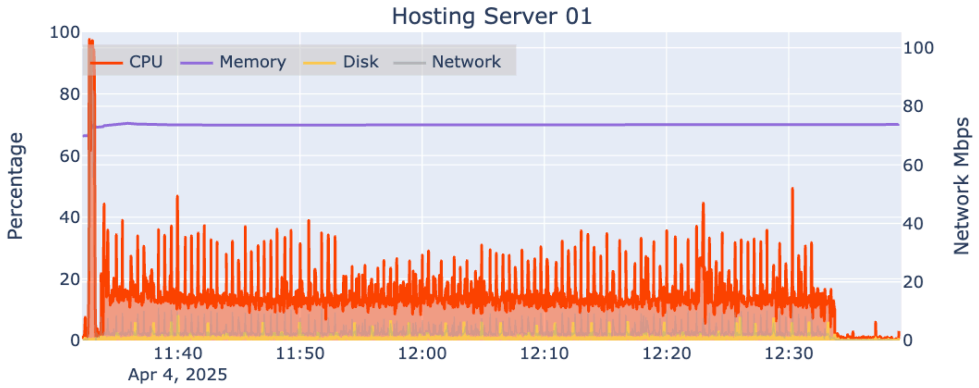


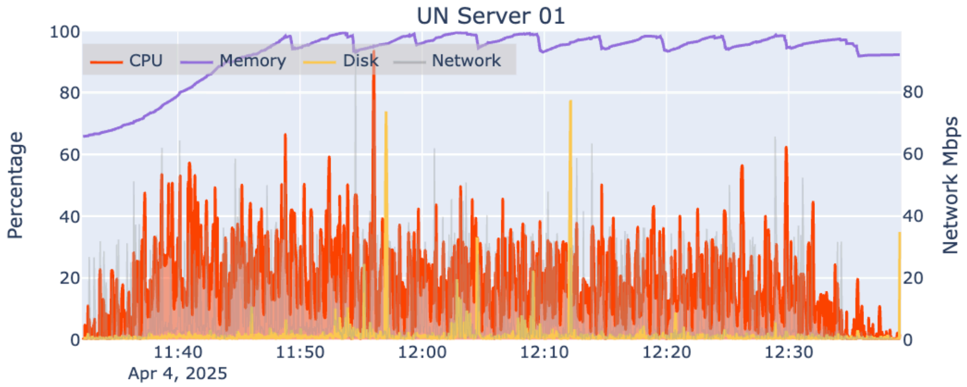
3:1 ArcSOCs to vCPU ratio

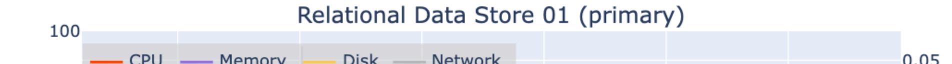
In this run, we configured three ArcSOCs per vCPU. In this case, 24 running ArcSOCs to 8 vCPU. Once again, CPU utilization (orange) is generally below 60% across all machines. Unfortunately, memory utilization (purple) on the UN GIS Server is maxing out, with the dips to 95% occurring as part of the cleanup process. Concurrent view requests (red) show they are steadily opening and closing, with an average of 36. This system appears to be handling the load, but it’s not sustainable due to a memory shortage.


Web Adaptor 01
CPU
Memory
Disk
Network
Percentage
Network Mbps
0
20
40
60
80
100
11:40
Apr 4, 2025
11:50
12:00
12:10
12:20
12:30

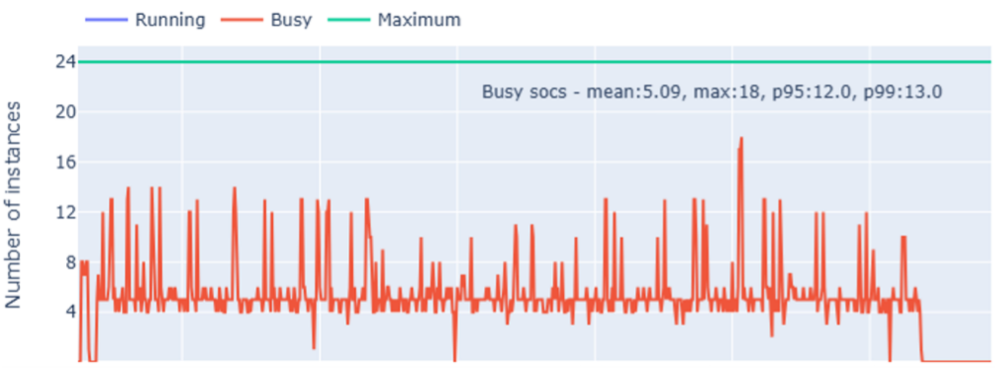

Enterprise Portal 01 (primary)
CPU
Memory
Disk
Network
Percentage
Network Mbps
0
20
40
60
80
100
0
10
20
30
40
50
11:40
Apr 4, 2025
11:50
12:00
12:10
12:20
12:30


Hosting Server 01
CPU
Memory
Disk
Network
Percentage
Network Mbps
0
20
40
60
80
100
11:40
Apr 4, 2025
11:50
12:00
12:10
12:20
12:30


UN Server 01
CPU
Memory
Disk
Network
Percentage
Network Mbps
0
20
40
60
80
100
0
20
40
60
80
11:40
Apr 4, 2025
11:50
12:00
12:10
12:20
12:30


Relational Data Store 01 (primary)
100
CPU
Memory
Disk
Network
0.05

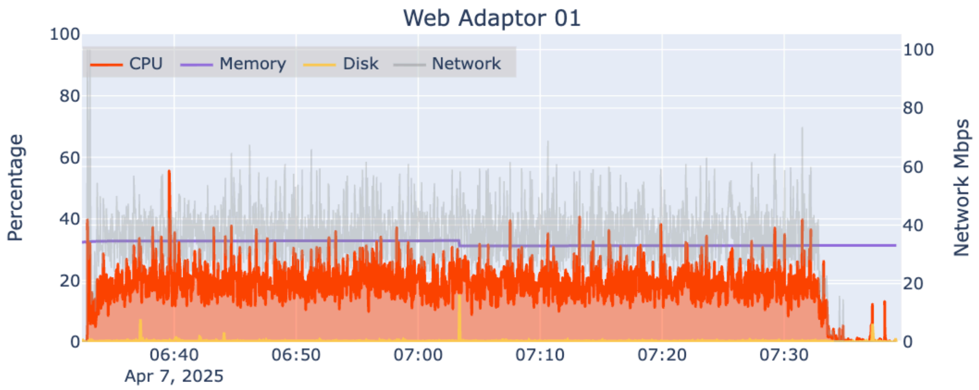
Further, by looking at the ArcSOC chart below, you can see that with a 3:1 ratio the UN GIS Server has 24 ArcSOCs running (the blue line is covered by the green line). However, the maximum busy is only 18, which consume memory even when not in use. This is an example of a poor configuration. Our workload doesn't require all of the ArcSOCs that are available. The six that are not needed (24 running minus 18 that are busy) consume resources (memory) unnecessarily. Increasing memory on the UN GIS Server may improve this situation, but it could also move the problem to the CPU or to the database. Testing and observation are required to make apt configuration and design choices to support the system.



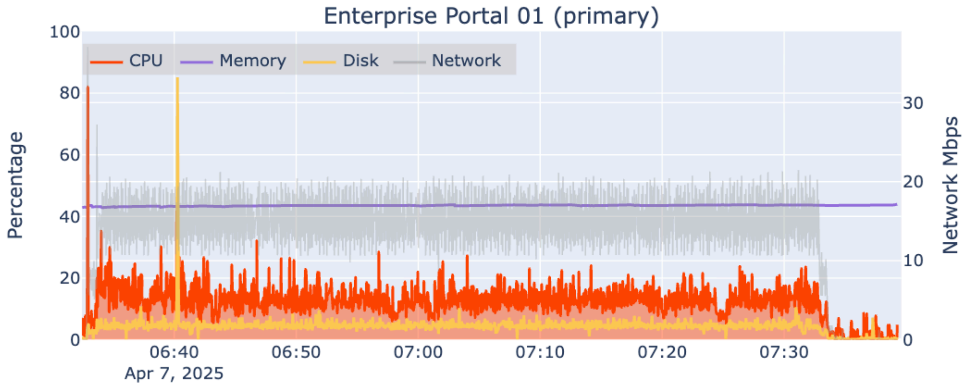
4:1 ArcSOCs to vCPU ratio

In this run, there were four ArcSOCs configured per vCPU. In this case, 32 running ArcSOCs to 8 vCPU on the UN GIS Server. CPU utilization (orange) is growing, with two peaks above 80% on the UN GIS Server. However, the biggest issue is near 100% memory usage (blue) on that instance, where even the cleanup process is struggling to keep pace.

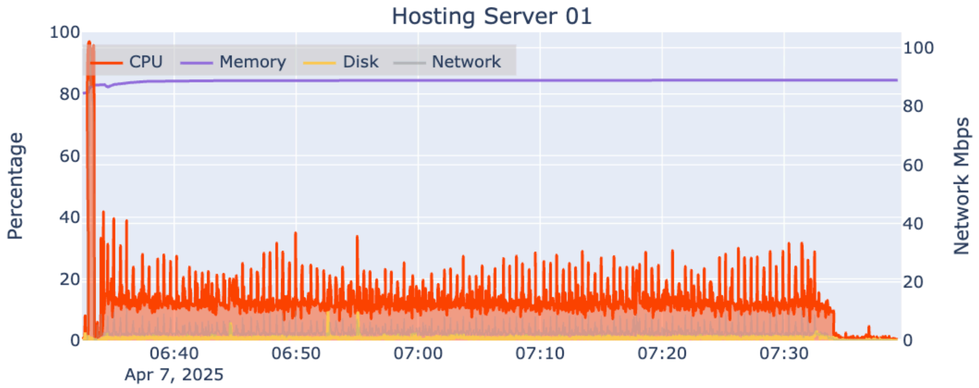
Concurrent view requests (red) at the bottom show they are still opening and closing steadily, with the same average of 36, like with the 3:1 ratio. This shows that the additional ArcSOCs are not providing any benefit to the system's performance or end user experience. Rather, they are merely consuming GIS server resources.



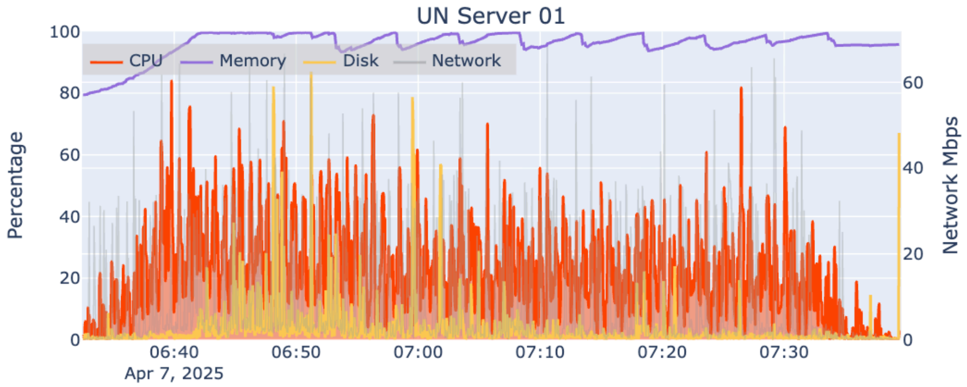
Web Adaptor 01



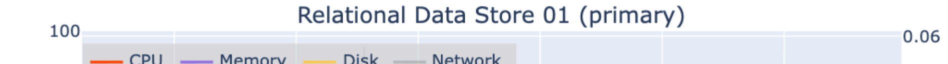
Enterprise Portal 01 (primary)



Hosting Server 01

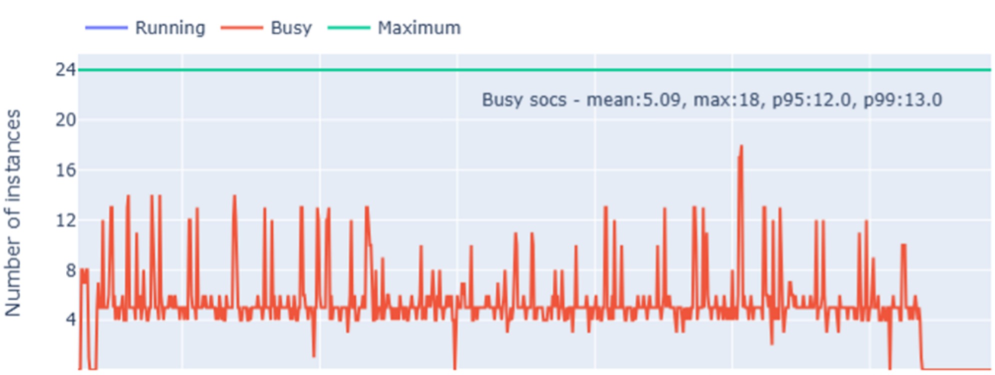


UN Server 01



Relational Data Store 01 (primary)

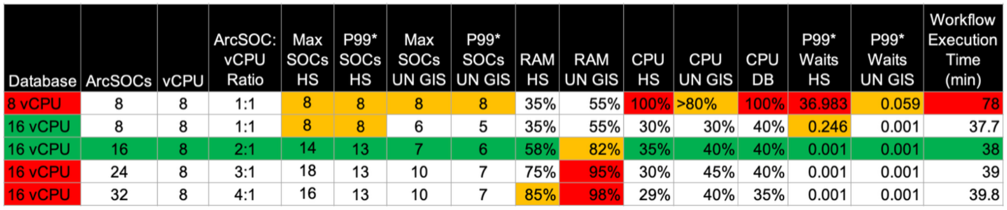
This is further validated by the ArcSOC chart below that shows a maximum of only 16 busy ArcSOCs. We can see clearly here how adding additional service instances only consumed unnecessary server resources, without providing any performance or user experience gains. Increasing UN GIS Server CPU and memory may improve results, but this could move the problem to the database server CPU. Testing and observing is key.



Conclusions and key takeaways

Throughout the five test runs, it was clear that the under-resourced database instance in the first test run noticeably degraded system performance and user experience. Once the database instance was properly sized for our workloads, the ArcSOC : vCPU ratios had a smaller impact on workflow execution times. Looking at the table below, we can see the 1:1 ArcSOC : vCPU ratio imposed some additional wait times (0.246s) on viewing workflows, but didn't significantly impact editing workflows (see P99 Waits HS). This is likely due to busy ArcSOCs on the hosting server.

The 2:1 ratio produced nearly identical workflow execution times with no significant waits, but did have high memory utilization on the UN GIS Server (82%). The 2:1 ratio is too high for these versioned editing workflows, where max ArcSOCs on the UN GIS Server only reach 7. Therefore, by increasing ArcSOCs on the UN GIS Server, we are only wasting server resources. However, the hosting server, supporting view-only workflows, easily supported the 2:1 ratio. The UN GIS Server needs more memory to support ratios at 2:1 and beyond. At 4:1, the hosting server also needs more memory.



Database	ArcSOCs	vCPU	ArcSOC: vCPU Ratio	Max SOCs HS	P99* SOCs HS	Max SOCs UN GIS	P99* SOCs UN GIS	RAM HS	RAM UN GIS	CPU HS	CPU UN GIS	CPU DB	P99* Waits HS	P99* Waits UN GIS	Workflow Execution Time (min)
8 vCPU	8	8	1:1	8	8	8	8	35%	55%	100%	>80%	100%	36.983	0.059	78
16 vCPU	8	8	1:1	8	8	6	5	35%	55%	30%	30%	40%	0.246	0.001	37.7
16 vCPU	16	8	2:1	14	13	7	6	58%	82%	35%	40%	40%	0.001	0.001	38
16 vCPU	24	8	3:1	18	13	10	7	75%	95%	30%	45%	40%	0.001	0.001	39
16 vCPU	32	8	4:1	16	13	10	7	85%	98%	29%	40%	35%	0.001	0.001	39.8

Resource utilization

We concluded that adding more service instances did not result in better user experience for editors, although it may improve responsiveness in our view workflows by reducing wait times. We saw that even ArcSOCs that were not in use were still consuming server resources. The table above shows workflow times slightly increasing as the ArcSOC : vCPU ratio is increased.

This implies that the additional available ArcSOCs were not necessary to support user requests and were unnecessarily consuming system resources (primarily memory). The table above confirms that editing workflows did not take advantage of the additional ArcSOCs, but the view workflows with much higher operations per hour did. Therefore, for our system, a 2:1 ratio of ArcSOCs to vCPU was most optimal for view only services on the hosting server and 1:1 is most optimal on the UN GIS Server.

Key takeaways

- An under-resourced database instance negatively impacted the whole system:
    - For our system, we determined the larger geodatabase instance size (16 vCPU) was critical
    - ArcSOCs, GIS Server CPU, and Web Adaptors were busy, making performance issues appear to be system wide
    - Workflow execution times took roughly twice as long to complete with an undersized database
    - An under-resourced database impacted performance significantly more than poorly configured map instances
    - Simply increasing database resources greatly improved system behavior and performance
- With a properly resourced database, increasing the ratio of ArcSOCs (map service instances) to vCPU did not improve the end user experience or workflow execution times
- Adding unnecessary service instances can negatively impact the system by consuming unnecessary additional resources
- Increasing the number of running map instances will impact GIS Server memory utilization, even when they are not busy
- Workload separation remains important - feature services exposing versioned data will use more GIS Server memory than view-only services
    - The available resources (CPU, RAM, and Disk I/O) on the database instance significantly improved the entire system's ability to handle load
- At a minimum, monitor database CPU, ArcSOC usage, GIS Server resources, and user experience to identify the optimal configuration for your system, especially after making any changes