

Test Study: Implementing a Parcel Information Management System in Azure on SQL Server

Last generated: March 06, 2026



Table of Contents

- **Test Study**
 - Introduction and overview 2
 - Tested workflows 4
 - Physical architecture 10
 - Design choices and considerations 15
 - Test methods and results 21
 - Using this test study 29

Parcel Management System (SQL Server)

We conducted these system tests in January 2026 to evaluate a Reference Architecture for a [Land Information Management System](#) with:

- Workflows representative of a medium sized organization managing parcels with the ArcGIS Parcel Fabric
- An enterprise geodatabase configured with SQL Server
- Microsoft Azure cloud infrastructure

Note:

Parcel management is often a core component of a Land Information Management System (LIMS); however, the workflows tested here represent only a subset of the broader business processes a full LIMS can support. For this reason, we refer to the system tested as a Parcel Management System, which is a type of Land Information Management System.

We designed the system presented in this test study at a physical level. It incorporates several design choices that are described in detail in the [design choices and considerations section](#). We designed the system to support an organization supporting a volume of typical parcel management workflows representative of a parcel management organization, like a local, state, or federal government, utility, or agricultural company.

We deployed and tested the system for both performance and end-user experience. The load was determined in multiples of operations per hour, or how many times each workflow is executed in an hour. You can read details about this approach in the [workflow pacing section](#).

You can also learn more [about test studies](#) at a high level to gain important context.

Note:

You should become familiar with the [reference architecture](#) before this test study, which has important information and resources that are not duplicated here.

Overview of capabilities

A foundational Parcel Management System delivers the following capabilities, as defined in the data editing and management system pattern, including:

- Service-based editing of relational data, including geometry and attributes

- Data collection workflows using forms and/or maps
- Complex parcel fabric management using the ArcGIS Parcel Fabric as an advanced data model
- Viewing and querying parcels through web applications

Software list

The system capabilities are delivered through the following software with all available patches applied:

- [ArcGIS Pro 3.5](#) (latest version [here](#))
- [ArcGIS Enterprise 11.5](#) (latest version [here](#))
- [ArcGIS Monitor 2025](#) (latest version [here](#))
- [ArcGIS Online](#)
- SQL Server 2022 - 16.0.1000.6

The design was deployed on virtual machines in Microsoft Azure with a Microsoft Windows operating system.

Note:

The software versions listed above were the latest available when this system test study was deployed and tested. Esri strongly recommends using the latest available software versions when possible.

Data characteristics

For these tests, we used a 5 GB parcel fabric dataset with a geographic extent of roughly 2,000 square miles. It contains about 1.5 million tax parcels, 30,000 subdivision boundaries, and 50,000 lots. This dataset is governed by a set of attribute rules that automatically populate parcel-related attributes such as tax year, dimensions, names, and identifiers, as well as enforcing data consistency.

Tested workflows

To ensure that workflows are useful for system design purposes, they need to represent real user experiences, and the actual steps that users will take in interacting with the system. These workflows represent some of the foundational activities required to maintain and access parcel and cadastral data. The contents of the workflows were defined by working with experienced staff and Esri customers to identify the specific steps, sequencing and type of activities involved in each workflow.

Developing workflows as detailed, stand-alone sets of steps allows them to be repeated, iterated on, or adjusted to improve performance or user experience. Once each workflow is clearly defined, it is used consistently throughout the remainder of testing to establish the performance baseline and conduct load tests.

We worked with real customers and Esri experts to define these workflows to make sure they portrayed parcel information management tasks realistically. However, your organization's workflows may have different or additional steps, like running a [Least Squares Analysis](#).

If your organization does not have clearly documented, standard workflows with individual steps defined, consider using these workflows as a starting point that can be adjusted based on your needs. To learn more about documenting your systems' workflows, see the blogpost describing how to [Unlock your ArcGIS system's ROI through workflow-centered design](#).

This implementation of the Land Information Management System's reference architecture was load tested for the following eight (8) workflows:

- [Merge parcels](#)
- [Split by metes and bounds](#)
- [Adjust boundary](#)
- [Import subdivision](#)
- [Shrink to seeds](#)
- [Move parcels](#)
- [Summarize parcels](#)
- [View parcels](#)

These workflows support both editor and general [user personas](#).

Editor workflows

Editors maintain the geometry and attributes of the land parcels. Editors may range in expertise and overall comfort with technology, so having the steps documented can have a positive impact on their work. These workflows represent the work done to maintain parcel information.

Merge parcels

Represents the set of steps an editor persona performs to combine two or more parcels into one.

1. Open Project
2. Locate tax name
3. Zoom to selection
4. Create and change version
5. Create record
6. Merge parcels
7. Toggle historic
8. Save
9. Reconcile
10. Post
11. Change version to default

Split parcel

Represents the set of steps an editor persona performs to split a parcel using metes-and-bounds descriptions.

1. Open Project
2. Locate Tax Name
3. Zoom to selection
4. Create and change version

Editor workflows

5. Create Record
6. Traverse edit
7. Create seeds
8. Build extent
9. Align Parcel
10. Clip Parcel
11. Update attributes
12. Validate topology
13. Error inspector
14. Save edits
15. Reconcile
16. Post
17. Change version to default

Adjust boundary

Represents the set of steps an editor persona performs to adjust adjacent parcel boundaries.

1. Open Project
2. Locate tax name
3. Zoom to selection
4. Create and change version
5. Create record
6. Copy lines to record
7. Copy parallel
8. Delete old line
9. Validate topology
10. Save
11. Reconcile
12. Post

13. Change version to default

Import subdivision

Represents the set of steps an editor persona performs to import subdivision layout into the existing fabric.

1. Open Project
2. Create and change version
3. Locate
4. Zoom to parcel
5. Create record
6. Import subs
7. Copy lines to
8. Build active
9. Align parcels
10. Clip
11. Sequential numbering
12. Highlight
13. Validate topology
14. Save
15. Reconcile
16. Post
17. Change version to default

Shrink to seeds

Represents the set of steps an editor persona performs to reduce complex polygons to their core "seed" points, making it easier to modify the lines that define parcels.

1. Open Project
2. Locate

Editor workflows

3. Zoom to parcel
4. Create and change version
5. Select parcel
6. Activate record
7. Shrink to seeds
8. Select line
9. Delete line segment
10. Create parcel line
11. Reconstruct from seed
12. Validate topology
13. Save
14. Reconcile
15. Post
16. Change version to default

Move parcels

Represents the set of steps an editor persona performs to shift several parcels to better align to a control network or aerial imagery.

1. Open Project
2. Manage record
3. Zoom to record
4. Create and change version
5. Select polygons
6. Move parcels
7. Build extent
8. Save
9. Reconcile
10. Post

11. Change version to default

General user workflows

General users are typically considered “viewers” within the system and primarily discover and use content created by others within the organization. These workflows represent those that users perform to access and view parcel information to get answers to their questions and make decisions in their work.

Summarize parcels

These steps represent a use case where an ArcGIS Dashboard is open on a screen and refreshes every 30 seconds.

1. Sign in to ArcGIS Enterprise
2. Open dashboard - Keep dashboard running for test duration - Layers are set to refresh every 30 seconds

View parcels

Represents the set of steps a general user persona performs to query parcel information to find lots that match certain criteria.

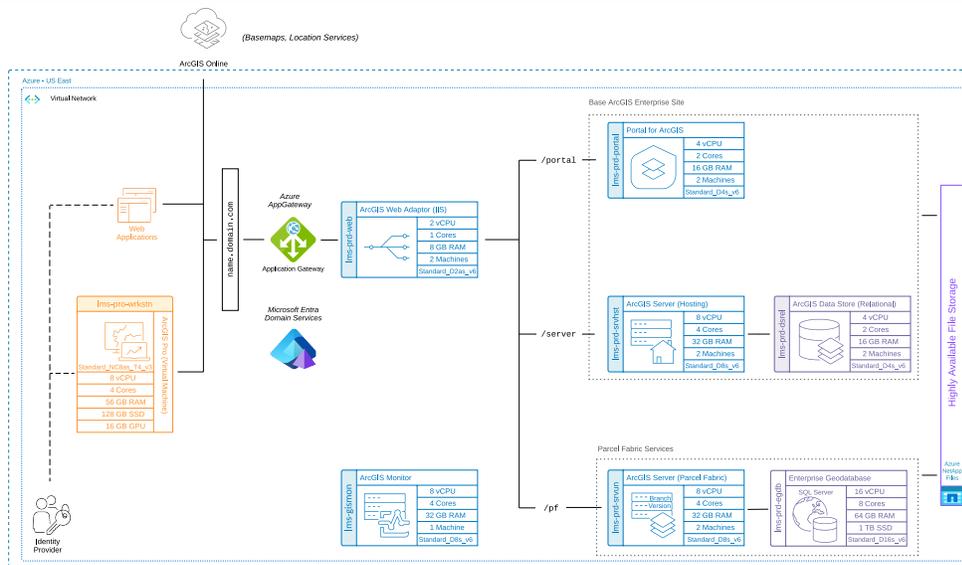
1. Sign in to ArcGIS Enterprise
2. Open web application
3. Enter tax name in search
4. Zoom to tax parcel

Physical architecture

This architecture was evaluated in January 2026 with:

- A medium-sized parcel management organization in mind
- Support for parcel management workflows, with a target design load as described in the [test methods](#) section
- An enterprise geodatabase configured with SQL Server
- Microsoft Azure cloud infrastructure

The system was designed for and tested against the specified workflows, with adjustments to the machine types and sizes as needed based on test results.



Download a [Microsoft Visio file](#) of this architecture. Learn more about [diagramming resources](#) for ArcGIS systems.

Note:

For more information on the software components and key interactions in this architecture, review the Land Information Management System [reference architecture](#).

Machine types and sizes

The following represent the machine sizes chosen and validated for the scope and purpose of this test study. However, it is highly recommended to follow a complete design process to account for your business and technical requirements.

Esri offers [system architecture design](#) services should you need help determining all of the different factors relating to your organization's physical design - such as networking, storage, system environments, and sizing. Minimum system requirements for each component are listed in the respective software [documentation available online](#).

Desktop (ArcGIS Pro & web browser)

- 3 machines used in [testing](#)
- Standard_NC8as_T4_v3
- 8 vCPU
- 56 GB RAM
- 16 GB GPU
- 128 GB Disk

Portal for ArcGIS

- 2 machines
- Standard_D4s_v6
- 4 vCPU
- 16 GB RAM
- 128 GB Disk

Note:

For this test system, 2 cores was sufficient. However, Esri [highly recommends a minimum of 4 cores for production systems](#).

ArcGIS GIS Server

- 2 machines
- Standard_D8s_v6

Machine types and sizes

- 8 vCPU
- 32 GB RAM
- 128 GB Disk

ArcGIS Server (hosting server)

- 2 machines
- Standard_D8s_v6
- 8 vCPU
- 32 GB RAM
- 128 GB Disk

ArcGIS Data Store (relational)

- 2 machines
- Standard_D4s_v6
- 4 vCPU
- 16 GB RAM
- 128 GB Disk

ArcGIS Web Adaptor

- 2 machines
- Standard_D2as_v6
- 2 vCPU
- 8 GB RAM
- 128 GB Disk

ArcGIS Monitor

- 1 machine
- Standard_D8s_v6
- 8 vCPU

Additional infrastructure considerations

- 32 GB RAM
- 128 GB Disk

File storage

- 1 instance
- NetApp Files
- 100 MiB/s throughput
- 1 TB Disk

Database

- 1 machine
- Standard_D16s_v6
- 16 vCPU
- 64 GB RAM
- 1 TB Disk

Directory Services

- Microsoft Entra Domain Services

Additional infrastructure considerations

The following are additional areas of consideration when designing a Network Information Management System and an explanation of some infrastructure choices made for this test study.

To learn more about designing ArcGIS systems with Azure components and services, see [Azure technologies](#).

Load balancing and reverse proxying

At least one third-party load balancer is required in a highly available ArcGIS Enterprise deployment - they handle client traffic to your portal and server sites and internal traffic between the software components. Although the ArcGIS Web Adaptor is considered a load balancer, it is insufficient on its

own to serve as a load balancer in a [high availability](#) configuration. Therefore, in this test study, Azure Application Gateway was used.

Database considerations

For the scope and purposes of this test study, a choice was made to deploy SQL Server on a virtual machine. However, depending on your needs, you might consider leveraging a database platform service like [Azure SQL Managed Instance](#) in a Microsoft Azure environment.

Shared storage

To successfully implement a highly available ArcGIS Enterprise deployment, the configuration stores must be stored in a highly available, shared location. This ensures this data remains accessible even if one server fails, providing uninterrupted service to end-users. Additionally, shared storage simplifies data management in a multi-machine deployment and improves scalability by centralizing data storage and allowing for expansion as needed. In this test study, Azure NetApp Files was used.

System components not included in the diagram

Note that while antivirus software and Azure networking components are not illustrated in this diagram, they were present in the test study.

Design choices and considerations

The following considerations are organized around the [architecture pillars](#) of the [ArcGIS Well-Architected Framework](#). The appropriate application of best practices and architectural approaches in each of these technical areas contributes significantly to the successful design and implementation of well-architected systems.

You can also refer to these [physical design considerations](#) for additional recommendations.

Performance and scalability

Workload separation

We chose to design for [workload separation](#) to help achieve an optimal distribution of compute resources across the system. In the test study, editing requests generally took longer to process than standard map requests, so we isolated editing workloads with dedicated compute resources in the form of a separate ArcGIS GIS Server site.

Additionally, isolating the system components themselves onto different machines helps to ensure they don't compete for system resources and allows for the opportunity to tailor machine types and sizes to the system requirements of each component.

GPU-enabled desktop machines

Selecting the proper GPU (Graphics Processing Unit) is essential for ensuring the performance of ArcGIS Pro in a virtualized environment. Tests revealed that adding a dedicated GPU to ArcGIS Pro virtual machines significantly improved end-user productivity and produced a net reduction in cost when operational expenses, such as labor costs, are considered.

- For example test results, see [Evaluating GPU and CPU impact on desktop editing workflows](#).
- Learn more about GPU hardware selection and [ArcGIS Pro virtualization](#) in the ArcGIS Architecture Center.

Watching for vCPU: CPU in the cloud

It's important to understand the ratio of virtual CPU (vCPU) to physical CPU when making design decisions so system components may be assigned appropriate resources. There is a 2:1 ratio of vCPU

: CPU for most of the instances in the diagram, apart from our Desktop instances, which have a 1:1 ratio.

Consider that different virtualization options may have different ratios. In addition to potential performance impacts, this may also have [Esri licensing](#) implications. For examples of public cloud ratios, see the following resources.

- [Supported CPU options for Amazon EC2 instance types](#)
- [Change the size of an Azure virtual machine](#)
- [Set the number of threads per core for Google Cloud Compute Engine](#)

GIS services configuration

Proper configuration of GIS services is critical to system performance and user experience satisfaction, and the misconfiguration of GIS service instances can introduce problems or reliability challenges in a system. For example, if the number of instances for a map or feature service are set too low, it can result in long client wait times and timeout errors. However, configuring too many instances consumes excessive machine resources, limiting the number of services that can be deployed on a fixed hardware configuration.

When the maximum instance setting is higher than the minimum, the system can automatically add new instances in response to demand. However, this can also cause perceived performance problems, since incoming requests must wait for the instance to start. For any system, it is important to understand service usage so that instance numbers and server resources can be adjusted to provide optimum performance.

- Learn more about [configuring service instance settings](#).

For our testing purposes, the ratio of service instances to vCPU (virtual CPUs) was set to 1:1 for each service, with the minimum and maximum instances set equally. Therefore, since our GIS and hosting server sites each had two instances with 8 vCPU, we had 16 instances per server site. The instance usage was monitored to determine how the system was handling the load. If at any point all instances on a GIS server become busy, we would expect high wait times for that service.

Note:

This was the optimal service instance configuration for our test system, but your organization's configuration may differ. Monitoring and telemetry capture are necessary to make informed choices for configuring your own service instance settings. See [The art and science of ArcSOC optimization](#) for guidance.

In this test study, the parcel fabric editing services were configured as follows:

- Minimum number of instances per service: 16
- Maximum number of instances per service: 16
- The total number of available instances was 16 because there were two ArcGIS GIS Server machines in the site

The hosting servers (view-only workflows) were configured as follows:

- Minimum number of instances per service: 16
- Maximum number of instances per service: 16
- The total number of available instances was 16 because there were two ArcGIS GIS Server machines in the site.

The specified service timeouts were configured as follows:

- Maximum time a client can use a service: 1800 seconds
- Maximum time a client will wait to get a service: 800 seconds
- Maximum time an idle instance can be kept running: 1800 seconds

Note:

Our timeout configuration was adjusted iteratively to address timeouts encountered during the testing process. Since these settings may vary based on specific requirements, it is recommended to conduct your own testing to identify the most optimal configuration.

Reliability

Backups

Backups are critical for Parcel Management Systems, like most data editing and management focused systems. While the tested design was not a production system, we did capture machine snapshots and database backups for each test run and before making any changes to the system. Virtual

machine snapshots were taken before and after any change in the environment, such as resizing a machine, installing a patch, or updating Windows. Snapshots were then cataloged to enable either:

- Roll back of a specific machine to a specific point in time
- Roll back of the entire environment to a specific point in time

Note that snapshots may not be enough to enable you to recover your environment. See [Backups and disaster recovery](#) for an overview of the backup process in ArcGIS Enterprise.

Refer to the [Land information management system reference architecture](#) for more information.

High availability

The choice to design this system with a [high availability](#) configuration of ArcGIS Enterprise components was made based on business and technical system requirement, along with other organizational goals such as achieving uninterrupted operations and minimizing downtime. This configuration is illustrated in the design with redundant system components and a cloud-native, highly available file store for file storage. This test study did not configure a highly available database for testing purposes, though relational database vendors have a variety of methods to approach high availability including cloud-native services.

Note:

Keep in mind that high availability configurations can significantly increase infrastructure and operational costs of the system and requires specialized skills to be successful. Learn more about [design choices and considerations](#) with regard to high availability for a Parcel Management System.

Observability

To perform successful system validation and deliver meaningful results, [system monitoring and telemetry capture](#) were key aspects to the test study.

ArcGIS Monitor and enterprise IT monitoring tools like Windows Performance Monitor were used to monitor the system's performance and capture telemetry on its behavior under certain conditions.

Logs were collected across different system components, including:

- IIS Web Server
- ArcGIS software components
- Windows Events
- ArcGIS Pro

Machine-level metrics such as CPU usage, RAM consumption, disk activity, and network activity were captured across all machines in the environment. Review the [test results](#) for more information.

Additionally, screen recordings were captured of conducted workflows to observe and assess end-user experience and productivity.

Automation

Because the scope of the test study was primarily focused on load testing, most types of automation that would be recommended for a production system, such as scripting administrative tasks, were not employed. However, in your environment, administrative scripts can have significant value to workflows and operations. Any automation scripting should be tested in a lower environment before deploying to production.

In this test study, the primary application of automation was for the purpose of simulating requests during load tests. Multiple workflows were run with virtual users at scale with the ability to apply to different load sizes, as illustrated in the [test results](#).

We used Python scripts to perform analysis on and identify patterns in service wait times, services instance utilization, response times, and failed requests to inform needed system changes. Python, PowerShell and SQL scripts were also used to restore the database to an original state after completing a load test.

Security

Security is an essential consideration for any enterprise IT system, including authentication and authorization, filtering, encryption, auditing, and hardening. ArcGIS software is architected to work effectively within secure networks, including those that are fully disconnected from the internet. It is critical to consider security requirements early in the design process for any production system.

While security was not a focus for this test study, we did include the use of an identity provider to provide proper user authentication and authorization, as seen in the physical architecture diagram. Subnet segmentation is another foundational security practice applied in this test study, rooted in the principles of least privilege and network isolation.

Related resources:

- [ArcGIS authentication models and providers](#)
- [Authorization and access models](#)

- [Secure network design](#)

Integration

While integration was not within the scope of the test study, a Parcel Management System often requires integration with other enterprise systems like Computer Aided Mass Appraisal (CAMA) systems. Learn more about [integration considerations](#) with ArcGIS.

Test methods and results

Testing was conducted to validate that the design would perform as expected and support the workflows, users, and intended load. System tests provide the opportunity to discover and correct problems during system deployment in lower environments, ideally before they appear in production. For this test study, the focus of the testing approach was to validate the system would support the workflows, to understand how load would impact the system and its components, and end-user experience.

Each component was monitored as the workflows were conducted against different load scenarios. Upon test completion, results were assembled and analyzed to identify both bottlenecks and over-resourced components in the system. This information was used to identify system components that needed to be scaled up, down, or out before further testing was repeated.

Manual user experience testing was conducted by capturing screen recordings of the workflow testers to ensure users of the system could complete their workflows productively.

For more information, see how to [design an effective test strategy](#).

Workflow pacing

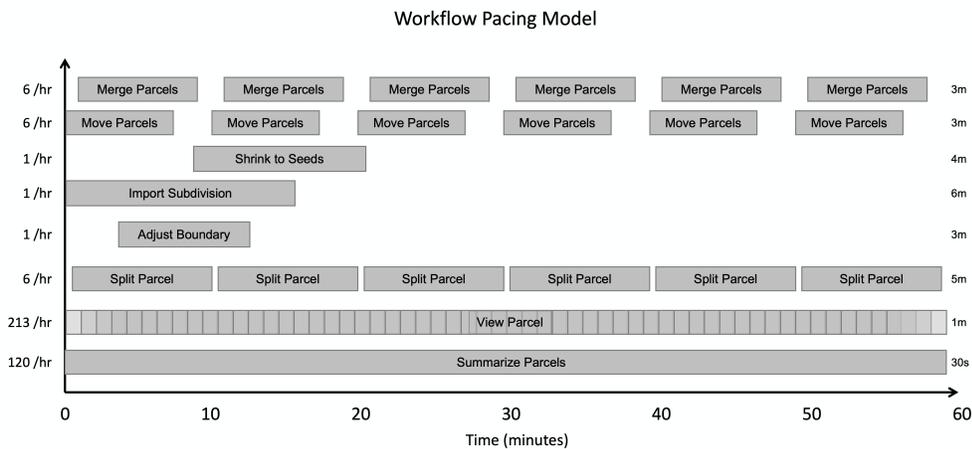
This test study applied a pacing model to the tested workflows. The pacing model shows how the test intends to simulate the pace of work at a parcel management organization, where workflows are performed as some number of operations per hour across a team of staff resources. This approach was based on Esri customer input and aimed to match the medium-sized parcel management customer scenario that the data was based on.

During the one-hour test period, workflows were distributed and staggered to avoid simultaneous initiation, while still allowing for overlap - mimicking how tasks unfold in real-world environments. In the pacing model below, you can see how we specified each workflow's pacing and number of operations per hour, which define the system's "design load".

Note:

For example, you can see that in the pacing model our design load accounts for 120 “Summarize Parcels” workflows to be performed per hour. From working with customers, we determined this to be representative of how many times a medium-sized organization would collectively perform this workflow in an hour. However, this number of workflows could be completed by any number of actual users, as some organizations might have a smaller number of staff who each perform the workflow multiple times per hour, while others could have a larger group of users who each perform the workflow less frequently. However, the total number of operations per hour that the system supports remains the same regardless of the distribution among the users.

The load was then increased by multiplying the workflows to a point where the system was no longer able to provide acceptable responses or support successful workflows, or in this case, to a point where it was large enough to validate the system would work for the intended type of organization. Note that the workflow pacing model applied in this test study might not match typical daily use at your organization.



Performance testing tools

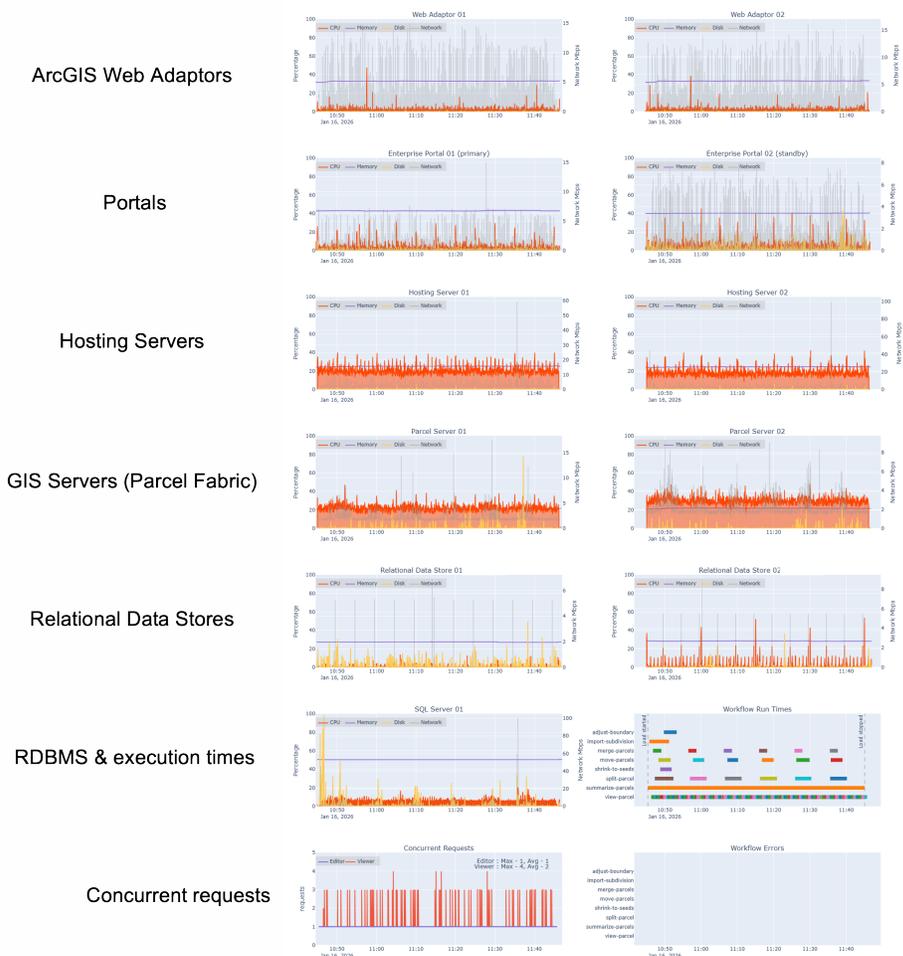
Because ArcGIS is a multi-tier system, performance tests were conducted across client, service, and data storage tiers, as well as the underlying infrastructure itself. In this test study, [JMeter](#) was used to simulate the user workflows and measure system performance under different loads. ArcGIS Pro requests were recorded and then replayed to simulate load in addition to manual workflows that were performed to assess end-user experience. Windows Performance Monitor and ArcGIS Monitor were also used to monitor resource utilization across different components.

For more information, see [tools for performance testing](#).

Test results

This architecture was validated with automated load tests and manual users in three scenarios, and you can see the results from each below. At a high level, the test results show that as implemented, the system is adequately resourced to support loads from the design load through 8x the design load. Tests also reinforced the importance of proper application and system configuration for performance.

Test scenario: Design load



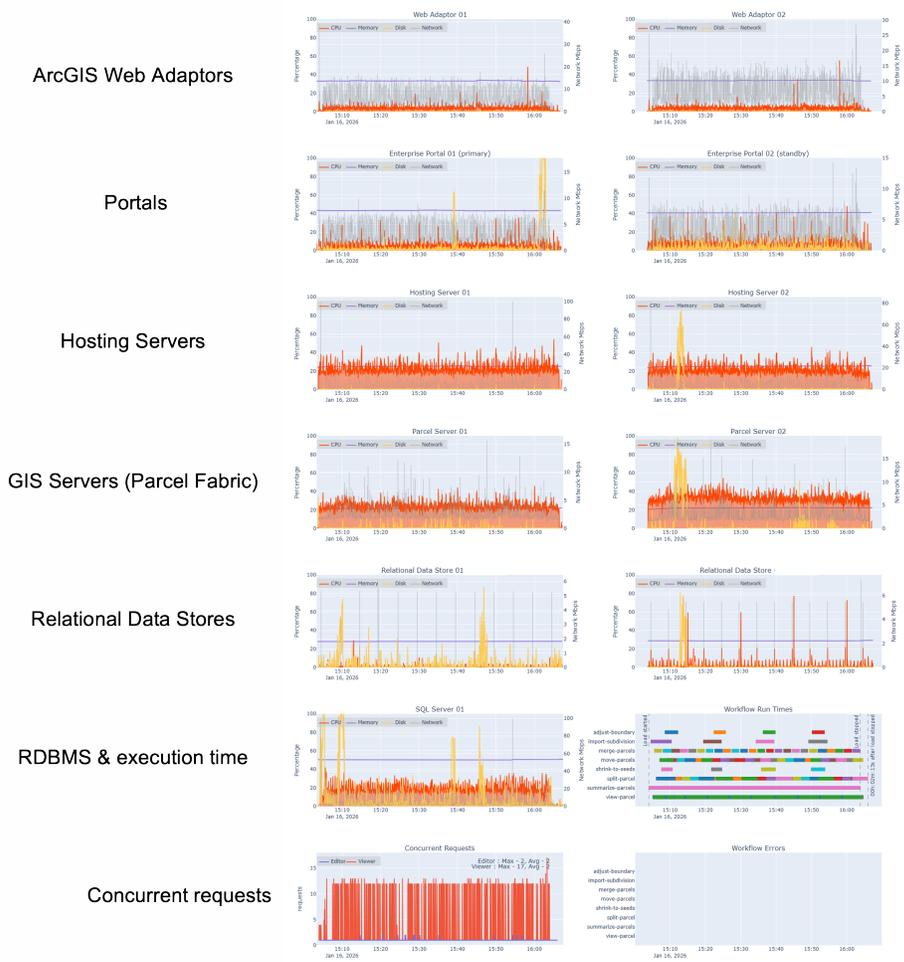
Observations:

- The system supported the load
- Hosting servers (viewing workflows) averaged below 30% CPU utilization (orange lines)

Test results

- The parcel servers (editing workflows) averaged below 40% CPU utilization
- SQL Server shows very low CPU utilization, typically staying below 15%
- The spike in disk utilization on the SQL Server instance can be attributed to a background Windows process (gold lines)
- Concurrent requests show the system supporting roughly 3 concurrent viewer requests (red) and 1 concurrent editor request (blue) at any given time across the test period

Test scenario: 4x design load



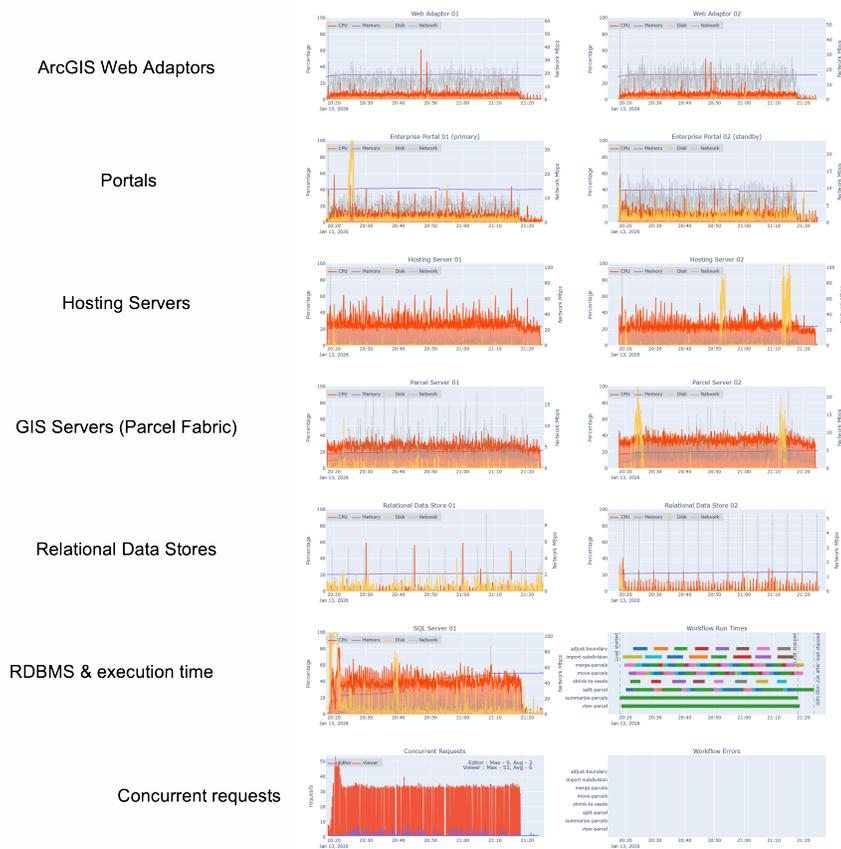
Observations:

- The system supported the load, with marginal resource utilization increases across components
- Hosting servers generally averaged below 40% CPU utilization

Test results

- Parcel servers generally averaged below 50% CPU utilization
- SQL Server generally stayed below 40% CPU utilization
- Periodic spikes in disk utilization can be attributed to workflows starting or specific workflow steps. Specifically, the spike between 15:10-15:20 is associated with the Summarize Parcels workflow, where multiple dashboards open at the same time
- Concurrent requests show the system supporting an average of 10 - 15 concurrent viewer requests and editor requests throughout the test period, with larger peaks of concurrent view requests that close very quickly.

Test scenario: 8x design load



Observations:

- The system supports the load, with expected resource utilization increases across components
- Hosting servers generally stay below 50% CPU utilization
- Parcel servers generally stay below 50% CPU utilization

Test results

- SQL Server shows significant increases in resource utilization, consistently reaching about 60% CPU utilization.
- Concurrent requests show the system consistently supporting peaks of 35 concurrent viewer requests and an average of two concurrent editor requests across the test period.
- The spike of read requests at 20:20 is the Summarize Parcels workflow starting.

Service instance configuration

In addition to virtual machine resource utilization, we also monitored ArcSOC utilization for each test run, to help understand if our services were properly tuned. For all runs up to 8x design load, busy ArcSOCs were well below the maximum (16), indicating we had configured more map instances than we needed for those loads. If this was a production environment with loads below our 8x design load, we could choose to reduce the size of the hosting server and GIS server machines to save money. This assumes we would monitor ArcSOC utilization alongside server CPU and memory to know when scaling is needed to meet demand. Further, we would need to make sure we are not overloading those machines, because every ArcSOC uses some memory and every busy ArcSOC uses a virtual CPU.

We can see in the diagram below that all 16 ArcSOCs are busy at certain times on the hosting server site at 8x design load. When all ArcSOCs are busy, we would expect to see service wait times increase (which we observed). However, the parcel server (right) shows lower ArcSOC utilization, only reaching a maximum of 9 in use out of the 16 configured.

The initial spike on the hosting server (left) was caused by dashboards starting up at the start of testing. We've corrected the workflow pacing for future test runs, spreading dashboard startups across several minutes to better reflect real-world scenarios.

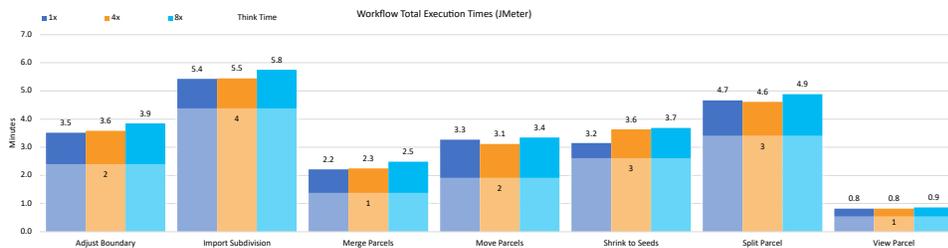


- To learn more, see this blog that details [The art and science of ArcSOC optimization](#).

User experience - manual workflow times

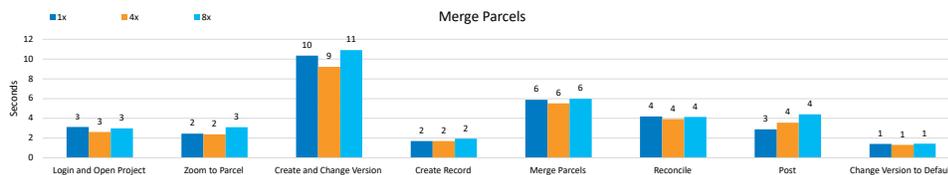
In addition to automated workflows, we also observed user experience by capturing screen recordings of the workflow testers and we extracted workflow durations (how long it took users to perform all steps in a workflow) from those recordings. This practice is to ensure the system’s users could complete their workflows productively.

As you can see in the chart below, the conducted workflow times are largely consistent with only minor variances across all test scenarios. This tells us that the system can support the increased load without negatively impacting perceived system responsiveness of end-users.



User experience - manual workflow step times

In addition to the workflows themselves, we also captured workflow times of key steps across all workflows. This represents the average time it took to complete a given step of each workflow while the system was under load. You can see in the chart below an example for the merge parcels workflow, where the time taken to complete each step was very consistent across all load scenarios. This pattern, with minor variability, is consistent across all workflows.



Conclusions and key takeaways

These tests were conducted in a test environment, not in a production system. Your system will likely differ in workflows, configuration, or design. For example, in Azure, the Web Adaptor is typically not used (assuming SAML) and the AppGateway distributes load directly to the servers. However, you can learn from these testing approaches and results for your own purposes:

- Designing for observability across the system provides invaluable information to properly tune performance against infrastructure costs, in addition to supporting other critical activities, like troubleshooting.
- Monitor your systems' server resources, ArcSOC utilization, and workflow completion times - both during testing (in a stage/test environment) and in production.
- Look for areas of misalignment in resources and resource utilization. For example:
 - At 8x our design load, the hosting server site seems properly sized for this volume of requests. However, the GIS Servers (parcel server) still have a lot of unused resources.
 - There are several opportunities to scale down our infrastructure to save on costs while maintaining the same performance and user experience.
 - There are also potential opportunities to reconfigure our ArcSOCs to be distributed more optimally to support our workflows.

Using this test study

This test study has been prepared and presented as part of the ArcGIS Architecture Center to help organizations understand how a specific reference architecture can be designed and deployed on specific physical infrastructure, and how it may perform when tested under load. To make best use of this test study, the following approach is suggested:

- Understand the [reference architecture](#) first. It contains important information and resources not duplicated in the test study.
- Assess the workflows and data used in the test study against those your system supports. The system may support additional or different workloads, but adjustments to the system, the workflows, and/or the load introduces risk.
- Use the test study as your guide for implementing your Parcel Management System. Where your workflows and/or requirements differ, we offer a design process you can follow.
- Determine possible improvement areas for your current implementation by considering design choices made in this test study.
- Evaluate the testing approach used against your own and look for improvement areas.
- Learn more about the architecture practices illustrated in the reference architecture in the ArcGIS Architecture Center.

Share your feedback!

Do you have ideas for how we can improve test studies in the future? [Please share your thoughts with us!](#)

Related resources

These resources reference key concepts and practices that may not have been explicitly stated within the test study, but can be significant factors in designing, implementing, and operating a successful Parcel Management System.

- [Build and manage a GIS program](#)
- [IT Governance](#)
- [Upgrades and patching](#)

Related resources

- [Choosing architecture components](#)
- [Physical design considerations](#)
- [Seven ways to use ArcGIS test studies](#)
- [ArcGIS Technology and Innovation resource catalog](#)