

Estimate performance and capacity requirements for Access Services

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious.  No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2010 Microsoft Corporation. All rights reserved.

# Contents

Contents

[Contents 2](#_Toc259095698)

[Estimate performance and capacity requirements for Access Services 3](#_Toc259095699)

[Test characteristics 3](#_Toc259095700)

[Dataset 3](#_Toc259095701)

[Test workloads 3](#_Toc259095702)

[Recommended and max definitions 4](#_Toc259095703)

[Your results may vary 4](#_Toc259095704)

[Hardware settings and topology 4](#_Toc259095705)

[Test results 5](#_Toc259095706)

[Recommendations 11](#_Toc259095707)

[Performance-related Access Services settings 12](#_Toc259095708)

[Troubleshooting performance and scalability 13](#_Toc259095709)

[Performance monitoring 13](#_Toc259095710)

[See Also 15](#_Toc259095711)

|  |
| --- |
|  |
| Estimate performance and capacity requirements for Access Services  |
|  |

This performance and capacity planning document provides guidance on the footprint that usage of Access Services has on topologies running Microsoft® SharePoint® Server 2010.

In this article:

* [Test farm characteristic](#_Test_Farm_Characteristic)s
* [Test results](#_Test_results)
* [Recommendations](#section4)
* [Troubleshooting](#_Troubleshooting_performance_and)

## Test characteristics

### Dataset

Access Services capacity and performance is highly dependent on the makeup of the applications that are hosted on the service. The size of tables and the complexity of queries often have the most impact. Our testing used “representative” sizes and complexities, but every application and data set is different, and your capacity and performance will be dependent on the applications in use, and their specific complexity and data size.

To evaluate our capacity profile, we simulated Access Services applications on a farm dedicated to Access Services (no other SharePoint tests running). Within this farm, we had the following representative sites:

* 1,500 Access Services applications with a “Small” size profile – 100 items maximum per list.
* 1,500 Access Services applications with a “Medium” size profile – 2,000 items maximum per list.
* 1,500 Access Services applications with a “Large” size profile – 10,000 items maximum per list.

Each application is made up of multiple lists, with the other lists appropriately sized based on this largest list. Note that Access Services can handle more data than 10,000 items, but we chose this number for our “Large” profile expecting that larger applications will not be common.

The applications were evenly distributed between:

* Contacts – A simple contact management application, dominated by a single list
* Projects – A simple task and project tracking applications, dominated by two lists (projects and tasks associated with each project)
* Orders – A simple order entry system, similar to Northwind sample of Microsoft Access® but scaled down, with many interrelated lists (orders, order details, invoices, invoice details, purchase orders, purchase order details, and so on)

### Test workloads

To simulate application usage, workloads were created which would perform one or more of the following operations:

* Opening forms
* Paging through the forms
* Filtering and sorting data sheets
* Updating, deleting and inserting records.
* Publishing application
* Render Reports

Each workload includes think time between user actions, ranging from 5 to 20 seconds. This is different from other SharePoint capacity planning documents. Access Services is stateful – we maintain in memory cursors and record sets between user interactions – making it important to simulate a full user session and not just individual requests. For a single user workload, there are on average .2 requests per second.

To select which application and which size application to choose between, the following percentages were used:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Small | Medium | Large |
| Contacts | 16% | 10% | 9% |
| Projects | 18% | 12% | 10% |
| Orders | 11% | 8% | 6% |

### Recommended and max definitions

For each configuration we ran two tests to determine a “green zone” or Recommended throughput that can be sustained and a “red zone” or Max throughput that can be tolerated for a short period of time but should be avoided.

The green zone was defined as a point at which the test being run consumes at most half the bottle necking resource. In our case the bottlenecking resource was %CPU on either of the three tiers, front-end Web server, ADS and SQL Server. So we first identified the bottleneck for a particular configuration. If the bottleneck happened to be ADS CPU we made sure that the green zone test consumed CPU on the ADS computers in a range between 40 and 50%.

For the red zone we picked a point at which the maximum throughput was reached, this turned out to be a CPU range between 80 – 90%. When searching for bottleneck we looked at %CPU, memory usage (private bytes), disk queue length, network IO and other resources that could result in a bottleneck.

Both the green and red zone tests were run for 1 hour at a fixed user load.

### Your results may vary

It is important to note that the specific capacity and performance figures presented in this article will be different from the figures in real-world environments. The simulation above is only a crude estimate of what actual users may do. The figures presented are intended to provide a starting point for the design of an appropriately scaled environment. After you have completed your initial system design, test the configuration to determine whether your system will support the factors in your environment.

## Hardware settings and topology

###### Lab Hardware

To provide a high level of test-result detail, several farm configurations were used for testing. Farm configurations ranged from one to four Web Front Ends (front-end Web server) servers, one to four Application Servers (in the case of Access Services, ADS, or Access Data Services), and a single database server computer that is running Microsoft SQL Server® 2008. In addition, testing was performed with from 4 client computers. All server computers were 64-bit, and the client computers were 32-bit.

The following table lists the specific hardware that was used for our testing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Machine Role | CPU | Memory | Network | Disk |
| Front-end Web server | 2 proc/4 core 2.33 GHz | 8GB | 1 gig | 2 spindles RAID 5 |
| Application Server (ADS) | 2 proc/4 core 2.33 GHz | 8GB | 1 gig | 2 spindles RAID 5 |
| SQL Server | 4 proc/4 core 2.6GHz | 32GB | 1 gig | DAS attached RAID 0 for each LUN |

###### Topology

From our experience, CPU on the Application Tier, where Access Data Services (ADS) is running, is an important limiting factor for throughput. So we varied our topology by adding additional ADS computers until it was no longer the bottleneck, and then added a front-end Web server to attain even more throughput.

* 1 front-end Web server computer to 1 ADS computer
* 1 front-end Web server computer to 2 ADS computers
* 1 front-end Web server computer to 3 ADS computers
* 1 front-end Web server computer to 4 ADS computers
* 2 front-end Web server computers to 1 ADS computer
* 2 front-end Web server computers to 2 ADS computers
* 2 front-end Web server computers to 4 ADS computers

Our SQL Server is a relatively strong computer and at no time did it become the bottleneck (although it started to approach CPU saturation on our 2x4 test), so we did not vary this in our topologies. Depending on the queries that are a part of a real world application mix, it is expected that the SQL Server tier could become the bottleneck.

Reporting Services was run in connected mode for all of our tests, running in the ADS tier.

### Test results

The following tables show the test results of Access Services in SharePoint Server 2010. For each group of tests, only certain specific variables are changed to show the progressive impact on farm performance.

Note that all the tests reported on in this article were conducted with think or wait time (see above for more details). This differs from the capacity planning results for other parts of SharePoint.

For information about bottlenecks of Access Services in SharePoint Server 2010, see the [Common bottlenecks and their causes](#bottlenecks) section later in this article.

##### Overall scale

The table below summarizes the impact of adding additional front-end Web server and dedicated ADS computers to the farm. These throughput numbers are specifically for the ADS computers, and do not reflect the impact on the overall farm.

|  |  |  |
| --- | --- | --- |
| Topology | Baseline solution Max (RPS) | Baseline Recommended (RPS) |
| 1x1 | 25 | 15 |
| 1x2 | 54 | 29 |
| 1x3 | 82 | 45 |
| 1x4 | 88 | 48 |
| 2x1 | 25 | 15 |
| 2x2 | 55 | 29 |
| 2x4 | 116 | 58 |

##### Recommended results

The following shows our results for recommended sustainable throughput.

As you can see above adding the 4th ADS shifts the bottleneck to the front-end Web server, and that adding a 2nd front-end Web server resolves the resource constraint on the front-end Web server tier. This would imply, that 1x1, 1x2, and 1x3 are reasonable configurations, but when the 4th ADS is added that a front-end Web server should also be added. Since we are scaling in a linear fashion (straight line between from 1x1 to 1x4), it can be assumed that the addition of a 7th ADS would also imply the addition of a 3rd front-end Web server, and so on, to satisfy the needs of the farm.

Remember that these results are based on a simulated work load only, and that an actual deployment should be monitored to find the point at which additional front-end Web servers are needed to support additional ADS computers. Also note that our front-end Web servers are dedicated to Access Services, and in reality the front-end Web servers are likely shared with other SharePoint workloads.

Response time at this throughput level is very fast, at less than ¼ second on average per request.

These results show that SQL Server was not a bottleneck, as adding a second front-end Web server resolved the resource shortage, and the SQL Server CPU was always less than 50%. However, be aware that the instance of SQL Server is shared with other SharePoint services and SharePoint itself, and so the cumulative effect may drive CPU or Disk I/O queue lengths to the point that they do become a bottleneck.

##### Maximum

In these results, throughput was pushed beyond was could be sustained.

In this first graph, we see that again a second front-end Web server was needed to maximum the usefulness of the 4th ADS computer. Again, your results may vary, as this is highly dependent on the applications and their usage patterns.

In this case, the response time is higher, as the overall system is under stress. However, these levels are still approximately one second, and acceptable to most users.

It may seem odd that with 4 ADS computers, 2 front-end Web servers have a higher response time than 1 front-end Web server. This is because the overall throughput of the system is higher with 2 front-end Web servers.

SQL Server is again not a limiting factor here, as adding the 2nd front-end Web server put us back on a linear scaling line. However, we are reaching nearly 90% CPU utilization on the instance of SQL Server, so there is very little headroom remaining. If we were to add a 5th ADS, SQL Server likely would have become the bottleneck.

##### Detailed results

#### Recommended

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Overall | 1x1 | 1x2 | 1x3 | 1x4 | 2x1 | 2x2 | 2x4 |
| Req/Sec | 14.96 | 28.76 | 45.22 | 48.01 | 14.85 | 28.77 | 58.02 |
| Tests/Sec | 2.00 | 3.81 | 6.11 | 6.42 | 1.99 | 3.81 | 7.80 |
| Average Latency | 235.80 | 241.21 | 247.21 | 244.87 | 240.70 | 242.26 | 250.94 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Average front-end Web server Tier | 1x1 | 1x2 | 1x3 | 1x4 | 2x1 | 2x2 | 2x4 |
| %CPU | 13.82 | 24.40 | 41.02 | 43.62 | 6.31 | 12.48 | 26.18 |
| Max w3wp private Bytes | 9.46E+08 | 2.31E+08 | 1.49E+09 | 1.55E+09 | 8.43E+08 | 9.84E+08 | 1.19E+09 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Average ADS Tier | 1x1 | 1x2 | 1x3 | 1x4 | 2x1 | 2x2 | 2x4 |
| %CPU | 46.30 | 42.83 | 43.74 | 34.51 | 46.56 | 43.45 | 42.13 |
| %CPU w3wp  | 33.61 | 31.15 | 30.71 | 24.29 | 33.48 | 31.64 | 29.72 |
| %CPU RS | 8.62 | 7.94 | 9.17 | 6.84 | 9.03 | 8.02 | 8.71 |
| Max total Private Bytes | 4.80E+09 | 4.89E+09 | 4.91E+09 | 4.62E+09 | 5.32E+09 | 4.82E+09 | 5.07E+09 |
| Max w3wp private Bytes | 2.10E+09 | 1.97E+09 | 2.04E+09 | 1.86E+09 | 2.00E+09 | 2.00E+09 | 2.07E+09 |
| Max RS Private Bytes | 1.78E+09 | 2.00E+09 | 1.97E+09 | 1.86E+09 | 2.30E+09 | 1.89E+09 | 2.02E+09 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SQL Server Tier (single computer) | 1x1 | 1x2 | 1x3 | 1x4 | 2x1 | 2x2 | 2x4 |
| %CPU | 12.07 | 18.64 | 32.53 | 36.05 | 9.89 | 21.42 | 47.46 |
| Avg Private Bytes | 2.96E+10 | 3.22E+10 | 3.25E+10 | 3.25E+10 | 2.89E+10 | 3.22E+10 | 3.25E+10 |
| Max Private Bytes | 3.26E+10 | 3.25E+10 | 3.25E+10 | 3.25E+10 | 3.25E+10 | 3.25E+10 | 3.25E+10 |
| Avg Disk Queue Length Total | 0.74 | 1.18 | 1.64 | 1.77 | 0.67 | 1.24 | 2.18 |

#### Maximum

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Overall | 1x1 | 1x2 | 1x3 | 1x4 | 2x1 | 2x2 | 2x4 |
| Req/Sec | 14.96 | 28.76 | 45.22 | 48.01 | 14.85 | 28.77 | 58.02 |
| Tests/Sec | 2.00 | 3.81 | 6.11 | 6.42 | 1.99 | 3.81 | 7.80 |
| Average Latency | 235.80 | 241.21 | 247.21 | 244.87 | 240.70 | 242.26 | 250.94 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Average front-end Web server Tier | 1x1 | 1x2 | 1x3 | 1x4 | 2x1 | 2x2 | 2x4 |
| %CPU | 13.82 | 24.40 | 41.02 | 43.62 | 6.31 | 12.48 | 26.18 |
| Max w3wp private Bytes | 9.46E+08 | 2.31E+08 | 1.49E+09 | 1.55E+09 | 8.43E+08 | 9.84E+08 | 1.19E+09 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Average ADS Tier | 1x1 | 1x2 | 1x3 | 1x4 | 2x1 | 2x2 | 2x4 |
| %CPU | 46.30 | 42.83 | 43.74 | 34.51 | 46.56 | 43.45 | 42.13 |
| %CPU w3wp  | 33.61 | 31.15 | 30.71 | 24.29 | 33.48 | 31.64 | 29.72 |
| %CPU RS | 8.62 | 7.94 | 9.17 | 6.84 | 9.03 | 8.02 | 8.71 |
| Max total Private Bytes | 4.80E+09 | 4.89E+09 | 4.91E+09 | 4.62E+09 | 5.32E+09 | 4.82E+09 | 5.07E+09 |
| Max w3wp private Bytes | 2.10E+09 | 1.97E+09 | 2.04E+09 | 1.86E+09 | 2.00E+09 | 2.00E+09 | 2.07E+09 |
| Max RS Private Bytes | 1.78E+09 | 2.00E+09 | 1.97E+09 | 1.86E+09 | 2.30E+09 | 1.89E+09 | 2.02E+09 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SQL Server Tier (single computer) | 1x1 | 1x2 | 1x3 | 1x4 | 2x1 | 2x2 | 2x4 |
| %CPU | 12.07 | 18.64 | 32.53 | 36.05 | 9.89 | 21.42 | 47.46 |
| Avg Private Bytes | 2.96E+10 | 3.22E+10 | 3.25E+10 | 3.25E+10 | 2.89E+10 | 3.22E+10 | 3.25E+10 |
| Max Private Bytes | 3.26E+10 | 3.25E+10 | 3.25E+10 | 3.25E+10 | 3.25E+10 | 3.25E+10 | 3.25E+10 |
| Avg Disk Queue Length Total | 0.74 | 1.18 | 1.64 | 1.77 | 0.67 | 1.24 | 2.18 |

## Recommendations

This section provides general performance and capacity recommendations.

Note that Access Services capacity and performance is highly dependent on the makeup of the applications that are hosted on the service. The size of tables and the complexity of queries often have the most impact. Our testing used “representative” sizes and complexities, but every application and data set is different, and your capacity and performance will be dependent on the applications in use, and their specific complexity and data size.

#### Hardware recommendations

Access Services uses standard hardware for both Web Front Ends and Application Servers – no special requirements are necessary. General SharePoint 2010 guidelines on CPU number, speed, and memory are applicable for computers in the ADS tier.

Scaled-up and scaled-out topologies

To increase the capacity and performance of one of the starting-point topologies, you can do one of two things. You can either scale up by increasing the capacity of your existing server computers or scale out by adding additional servers to the topology. This section describes the general performance characteristics of several scaled-out topologies.

The sample topologies represent the following common ways to scale out a topology for an Access Services scenario:

* To provide for more user load, check the CPU for the existing Access Services application servers. Add additional CPUs and/or cores to these servers if possible, and more Access Services server computers as needed. This can be done to the point that the front-end Web server’s become the bottleneck, and then add additional front-end Web server computers as needed.
* In our tests, memory on the front-end Web server and ADS tiers was not a bottleneck. Depending on the size of the result sets, it is possible that memory could become an issue, but we do not expect that to be the norm. Track the private bytes for the ADS w3wp process, as described below.
* In our tests, SQL Server was not a bottleneck. However, our tests were run in isolation from other SharePoint Services. SQL Server CPU and Disk I/O should be monitored and additional servers or spindles added as needed.

### Performance-related Access Services settings

One of the ways to control the performance characteristics of Access Services is to limit the size and complexity of queries that can be performed. Access Services provides a set of configurable throttles for controlling queries. Each of the following can be set through SharePoint Central Administration > Application Management: Manage Service Applications > Access Services.

In general, the amount of data that needs to be retrieved from SharePoint in order to perform a query will have a significant impact on performance. This can be controlled in a number of ways. First, the inputs to a query can be limited:

* Maximum Sources per Query
* Maximum Records per Table

Second, the resulting size of a query can be limited:

* Maximum Columns per Query
* Maximum Rows per Query
* Allow Outer Joins

In addition to the size of the query (data size in and out), the processing complexity on the data can be controlled, to reduce the CPU load on the ADS tier:

* Maximum Calculated Columns per Query
* Maximum Order by Clauses per Query

Obviously the above settings will impact the applications that can be run on the server. For example, if an application is written with 40 output columns from a query, and the settings is below this level, then the application will throw a runtime error. A balance between user need and acceptable performance must be struck, and is highly dependent on the type of Access applications that are expected to run be run on the farm.

One additional, more drastic measure, can be taken. SharePoint supports a set of query operations natively, which Access Services augments to cover a broader set of application scenarios. For Access Services to enhance SharePoint’s queries, there is the potential that a large amount of data may need to be retrieved from the SharePoint content database. Instead, Access Services can be set to stick with only query operations which can be natively supported by SharePoint, thus avoiding the data fetch required for more complex operations:

* Allow Non-Remotable Queries

#### Common bottlenecks and their causes

During performance testing, several different common *bottlenecks* were revealed. A bottleneck is a condition in which the capacity of a particular constituent of a farm is reached. This causes a plateau or decrease in farm throughput.

The following table lists some common bottlenecks and describes their causes and possible resolutions.

## Troubleshooting performance and scalability

|  |  |  |
| --- | --- | --- |
| Bottleneck | Cause | Resolution |
| ADS CPU | Access Services is dependent on a large amount of processing in the application tier. If a 1x1, 1x2, or 1x3 configuration is used, likely the first bottleneck encountered with be the CPU on the ADS servers. | Increase the number of CPUs and/or cores in the existing ADS computers, or add additional ADS computers. |
| Web server CPU utilization | When a Web server is overloaded with user requests, average CPU utilization will approach 100 percent. This prevents the Web server from responding to requests quickly and can cause timeouts and error messages on client computers. | This issue can be resolved in one of two ways. You can add additional Web servers to the farm to distribute user load, or you can scale up the Web server or servers by adding higher-speed processors.  |
| Database server disk I/O | When the number of I/O requests to a hard disk exceeds the disk’s I/O capacity, the requests will be queued. As a result, the time to complete each request increases. | Distributing data files across multiple physical drives allows for parallel I/O. The blog [SharePoint Disk Allocation and Disk I/O](http://go.microsoft.com/fwlink/?LinkId=129557) (http://go.microsoft.com/fwlink/?LinkId=129557) contains much useful information about resolving disk I/O issues. |
| Reporting Services CPU Utilization | The Reporting Services process is using a large share of the CPU resources. | Dedicate a computer to reporting services, taking load off of the ADS tier (connected mode) or the front-end Web server tier (local mode). |

### Performance monitoring

To help you determine when you have to scale up or scale out your system, use performance counters to monitor the health of your system. Use the information in the following tables to determine which performance counters to monitor, and to which process the performance counters should be applied.

#### Web front end

The following table shows performance counters and processes to monitor for Web servers in your farm.

|  |  |  |
| --- | --- | --- |
| Performance counter | Apply to object | Notes |
| % Processor time | Processor(\_Total) | Shows the percentage of elapsed time that this thread used the processor to execute instructions. |
| Private Bytes | Process(w3wp) | This value should not approach the Max Private Bytes set for w3wp processes; if it does, further investigation is needed into what component is consuming the memory. |

#### Access Data Services

The following table shows performance counters and processes to monitor for Application Servers, or Access Data Services (ADS) in this case, within your farm.

|  |  |  |
| --- | --- | --- |
| Performance counter | Apply to object | Notes |
| % Processor Time | Processor(\_Total) | Shows the percentage of elapsed time that this thread used the processor to execute instructions. |
| % Processor Time | Process(w3wp) | The ADS runs within its own w2wp process, and it will be obvious which w2wp process this is as it will be getting the bulk of the CPU time. |
| Avg. Disk Queue Length | PhysicalDisk(\_Total) | Watch for excessive disk writing due to logging. |
| % Processor Time | Process(ReportingServicesService) | Reports are handled by SQL Server Reporting Services. If an excessive number of reports are being run, or reports are very complex, then the CPU and Private Bytes for this process will be high. |
| Private Bytes | Process(w3wp) | Access Services caches the results of queries in memory, until the user’s session expires (the timeout for which is configurable). If a large amount of data is being processed through the ADS, then memory consumption for the ADS’ w3wp will rise. |
| Private Bytes | Process(ReportingSrevicesService) | Reports are handled by SQL Server Reporting Services. If an excessive number of reports are being run, or reports are very complex, then the CPU and Private Bytes for this process will be high. |

#### SQL Server

The following table shows performance counters and processes to monitor for database servers in your farm.

|  |  |  |
| --- | --- | --- |
| Performance counter | Apply to object | Notes |
| % Processor time | Processor(\_Total) | Shows the percentage of elapsed time that this thread used the processor to execute instructions. |
| % Processor Time | Process(sqlservr) | Average values greater than 80 percent indicate that processor capacity on the database server is insufficient. |
| Private Bytes | Process(sqlservr) | Shows the average amount of memory being consumed by SQL Server |
| Avg. Disk Queue Length | PhysicalDisk(\_Total) | Shows the average disk queue length – database requests waiting to be committed to disk. This is often a good indicator that the instance of SQL Server is becoming overloaded, and that possibly additional disk spindles would help to distribute the load. |