

Configuration Manager (current branch) site sizing guidance and answers to common performance-related questions

March 2018

Introduction

The latest release of System Center Configuration Manager (current branch) leads the industry in scalability and performance. Documented online are the maximum supported scalability limits (<https://docs.microsoft.com/en-us/sccm/core/plan-design/configs/size-and-scale-numbers>) as well as the hardware guidelines (<https://docs.microsoft.com/sccm/core/plan-design/configs/recommended-hardware>) needed to run sites at the very largest environment sizes. This document gives supplemental performance guidance for Configuration Manager environments of all sizes.

Specific details and test results generated by the product team are supplied in this document, especially around the requirements for the largest contributor of performance bottlenecks in Configuration Manager environments, that of the disk subsystem. Suggested disk IOPS requirements are provided for various size environments as well as a documented repeatable method that you can use to reproduce these numbers in your own environments and on your own hardware. This allows for more accurate estimates of the hardware required to deploy Configuration Manager.

We also address frequently asked questions for common performance issues.

The information contained in this document represents the current view of Microsoft Corporation on the issue discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for information purposes only, MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

© Microsoft Corporation. All rights reserved.

Performance test methodology used by the product team

While Configuration Manager can be deployed in many unique ways, a few variables such as **Feature Intervals** (like inventory cycles) and the number of various **Objects** that are referenced or deployed by the system (like the number of users or total software deployments) are important to understand in any sizing discussions. When the product team validates product performance, these variables are applied as part of a **Load** which generates objects at a rate which is representative of what enterprise customers encounter on a daily basis during production deployments for a given size environment.

Note: Before we released the current branch version of the product, these numbers were tested at (and beyond) documented maximum values, or when unspecified maximums existed, based on knowledge gained from our experience in the Configuration Manager community. Now, with the current branch builds, the customer telemetry has helped return much better information about how customers use the product on an everyday basis. This data has enabled the product team to significantly improve and create more representative testing, ensuring validation of the scenarios, configurations, and settings that are used the most by the majority of our customers. All of the recommendations in this document are based on these averages; your experiences may vary based on the size and configuration of your environment. In general, common sense is required with Configuration Manager when it comes to objects and intervals. Just because you can collect every file on a system, or just because you can set the interval for a cycle to one minute, doesn't mean you should!

Below we've tried to highlight some of the key settings and recommendations we use when testing and modeling the processing needs that our largest enterprises demand. This helps to set some basic performance expectations on the system for the size of hardware we suggest later in the document.

Feature intervals/settings: In general, most testing is done with default intervals for the key cycles in the system (for example, hardware inventory at once per week with a larger than default .mof). Some re-occurring feature intervals (especially hardware and software inventory cycles) can have significant effect on an environment's performance characteristics. For environments where default intervals or very aggressive data collection is enabled, oversized hardware is recommended in direct proportion to the increase in activity. For example, if you have 25,000 desktop clients and want to collect hardware inventory two times faster than the default interval you should start by sizing your site's hardware as if you had 50,000 clients.

Objects: Testing is performed with "upper average" limits for what large enterprises tend to use with the system (for example, thousands of collections, and thousands of applications deployed to hundreds of thousands of users or systems). Tests are performed simultaneously with *all* objects in the system at this number. Many customers leverage several features but generally not *all* features of the product at these upper limits. By testing with *all* features of the product at these limits it helps to ensure the best possible system wide performance while allowing some amount of buffer for when one feature is used "above average" by a particular customer.

Loads: Testing takes place with various feature intervals and object counts as mentioned above, but testing is also done beyond the standard "average day" load. Simulations are performed that are meant to generate peak usage demands on the system. Examples of these loads include simulating Patch Tuesday rollouts to ensure update compliance data can be returned in a timely manner during these

peak activity days, or simulating site activity that would be generated during a widespread malware outbreak to ensure timely notification and response would be possible. This means after deploying machines of the recommended size you might think they are being underutilized on any given day but remember that some processing buffer is needed for these more extreme situations.

Configs: Testing is performed on a wide range of hardware (physical, Hyper-V, Azure) as well as across a wide mixture of supported operating systems and SQL Server versions. In general physical versus Hyper-V versus Azure all return comparable performance results when configured similarly to equivalent physical hardware. Newer server operating systems tend to perform equal or better to older supported server operating systems. The most variation we see in testing comes from the SQL Server versions in use. (More information on this is in the frequently asked questions section below.) When qualifying the product's scalability maximums, we always validate in the worst cases for our supported configurations. While all of our supported platforms meet the minimum requirements, usually the latest versions of supporting products (Windows, SQL, etc.) will produce *even better* performance.

Key items that can affect Configuration Manager performance

The product team tests with a wide variety of settings in order to exercise the product in many different ways and at many different site sizes. However, there are some particular settings and objects that can dramatically affect performance. Below we have outlined some of these to consider in your environment.

Note: While many aspects of the product do not have officially documented maximums or user interface limits to prevent excessive usage, going beyond the guidance below can have significant adverse effects on a site's performance. Exceeding these levels or ignoring this guidance should be done with caution, typically requires larger hardware, and may even render your environment unmaintainable until you reduce the frequency or count of various objects to lower levels.

Hardware inventory: We baseline performance for Configuration Manager with hardware inventory collection set to once per week with the default .mof size plus approximately 20% additional properties. It is not recommended to enable all properties and instead collect only properties you actually need. Pay special attention when collecting properties that will *always* change *every* inventory cycle (for example, available virtual memory) as they can cause excessive churn on every single inventory cycle from every client.

Software inventory: We baseline performance for Configuration Manager with software inventory collection set to once per week with "Product only" details. In particular, collecting a lot of files can place a significant strain on the inventory subsystem. Avoid specifying a filter which could end up collecting thousands of files (for example, *.exe or *.dll) across many clients.

Collections: We baseline performance for Configuration Manager with several thousand collections with a variety of scope, size, complexity and update settings. Configuration Manager site performance is not *directly* related to the sheer number of collections on a site but is instead a cross-product of the complexity of collection queries, the full update frequency of collections, incremental updates (and frequency of changes), dependencies among collections, as well as the number of clients in collections that deployments or other Configuration Manager items are scoped to impact.

Minimize, where possible, collections with very expensive or complicated dynamic rule queries. When these types of rules are required, carefully consider appropriate update intervals and update times to minimize the impact of collection re-evaluation on the system (for example, updating at midnight instead of 8:00 AM).

Collections enabled with incremental updates ensure quick and timely updates to collection membership. Even though incremental updates are very efficient there is still load put on the system when using them. Balance anticipated change frequency with the need for near real-time updates on membership. For example, if you expect heavy churn in collection members and you **do not** require near real-time membership then updating the collection using a scheduled full update on some interval is more efficient and produce less load on the overall system. When enabling incremental updates also reduce the interval of any scheduled full updates on the same collections. They are only a backup method of evaluation since incremental updates should keep your collection membership updated in near real-time. Product documentation (<https://docs.microsoft.com/en->

[us/sccm/core/clients/manage/collections/best-practices-for-collections#do-not-use-incremental-updates-for-a-large-number-of-collections](https://www.microsoft.com/en-us/sccm/core/clients/manage/collections/best-practices-for-collections#do-not-use-incremental-updates-for-a-large-number-of-collections)) does have a recommended maximum number of total collections marked for incremental updates, but as it points out, your experience can vary based on many factors.

Collections with only direct membership rules and with a limiting collection which is *not* performing incremental updates do not require any scheduled full updates. Update schedules for these types of collections can be disabled to prevent unnecessary load on the system. If the limiting collection is set to incremental updates, then collections with only direct membership rules may not reflect membership updates for up to 24 hours or until a scheduled refresh takes place.

While *not* generally considered a best practice, some organizations create hundreds or even thousands of collections as part of various business processes. When using automation to create collection it is *very* important to correctly enable incremental updates (if needed) and/or minimize and spread out any full update schedules to avoid hot spots of collection evaluation during a particular period of time.

Establish a regular collection grooming process to delete unused collections, especially when collections are created in an automated fashion and may no longer be required after some period of time.

Remember that policies are created for all objects in your collections when things (like deployments) are targeted to them. When membership changes (either due to scheduled refresh or incremental updates) it can create a lot of other work for the system as a whole to process. The latest current branch builds have special optimizations for policies targeted to the All Systems and All Users collections. When targeting your entire enterprise, we recommend using the built-in collections instead of using a clone of these built-in collections.

To investigate collection performance even deeper you can use the Collection Evaluation Viewer (CEViewer), found in the Configuration Manager Toolkit (<https://www.microsoft.com/en-us/download/details.aspx?id=50012>).

Discovery methods (various): We baseline performance with server-based discovery methods running once a week (with delta discovery as appropriate, enabled to keep the data fresh during the week) and discovering a number of objects proportional to the simulated enterprise size. The performance baseline for heartbeat discovery is also once a week.

Discovery data is global data. One of the most common performance problems we see with server-based discovery methods is that they are often misconfigured in a hierarchy causing duplicate discovery of the same resources from multiple primary sites. Carefully configure server discovery methods on multiple primary sites to optimize communication with the target service (for example, Active Directory domain controllers), but also to ensure the same scope of discovery is not duplicated at multiple primary sites.

Sizing guidance

Based on our testing using the methodology outlined above, **Table 1** (below) gives general *minimum* guidelines for necessary hardware requirements for a specific number of managed clients. The values in this table should allow most customers with that number of clients to process objects fast enough to administer a site of that size. However, computing power continues to decrease in price every year and some of the requirements below could almost be considered “small” in terms of modern available server hardware configurations. Hardware that exceeds these guidelines will proportionally increase site performance and is recommended for administrators where additional processing power may be required or special product usage patterns may exist.

Table 1

Desktop Clients	Site Type / Role	Cores ¹	Memory (GB)	SQL Memory Allocation	IOPS: Inboxes ²	IOPS: SQL ²	Storage Space Required (GB) ³
25k	Primary or CAS with database site role on the same server	6	24	65%	600	1700	350
25k	Primary or CAS	4	8		600		100
	Remote SQL	4	16	70%		1700	250
50k	Primary or CAS with database site role on the same server	8	32	70%	1200	2800	600
50k	Primary or CAS	4	8		1200		200
	Remote SQL	8	24	70%		2800	400
100k	Primary or CAS with database site role on the same server	12	64	70%	1200	5000	1100
100k	Primary or CAS	6	12		1200		300
	Remote SQL	12	48	80%		5000	800
150k	Primary or CAS with database site role on the same server	16	96	70%	1800	7400	1600

150k	Primary or CAS	8	16		1800		400
	Remote SQL	16	72	90%		7400	1200
700k	CAS with database site role on the same server	20+	128+	80%	1800+	9000+	5000+
700k	CAS	8+	16+		1800+		500+
	Remote SQL	16+	96+	90%		9000+	4500+
5k	Secondary Site	4	8		500	-	200
15k	Secondary Site	8	16		500	-	300

Table 1 Notes

1 - Cores: Configuration Manager performs a lot of simultaneous processing, so a certain minimum number of CPU cores is necessary for various site sizes. While cores get faster each year, ensuring a certain minimum number of cores working in parallel is important. In general, any server-level CPU produced after 2015 will meet the basic performance needs for the cores specified above. Configuration Manager will take advantage of additional cores beyond the recommendations, but generally, once the minimum suggested cores are available, prioritize investment in additional CPU resources by increasing the speed of existing cores and not by adding more, slower cores to your servers. For example, if given the choice between 16 faster cores or 24 slower cores, Configuration Manager will perform better on key processing tasks with the 16 fast cores (assuming enough other system resources, especially disk IOPS, are available).

The relationship between cores and memory is also important. In general, having less than three to four GB of RAM per core is not recommended as this reduces the total processing capability on your SQL servers. You need more RAM per core when SQL is co-located with the site server components.

Note: All testing was done with machine power plans set to allow maximum CPU power consumption and performance.

2 - IOPS: Inboxes / IOPS: SQL: These values refer to the IOPS needed on the logical drive(s) used by Configuration Manager and SQL. The **IOPS: Inboxes** column details the IOPS needed on the logical drive where the Configuration Manager inbox directories reside. The **IOPS: SQL** column details the total IOPS needed on the logical drive(s) used by various SQL files. These are differentiated because we recommend different formatting for these two drives. See below for more information and examples on suggested disk configurations and SQL file best practices, including details on splitting them across multiple volumes.

We generated the data for both of these IOPS columns with the industry standard tool, Diskspd. See **Appendix A** for instructions on duplicating these measurements. In general, once basic CPU and

memory requirements are met, the storage subsystem has the largest impact on site performance, and improvements here will give the most payback on investment.

3 - Storage Space Required: These are real-world values that may differ from other documented recommendations. We provide these numbers only as a general guideline; individual requirements could vary widely. Carefully plan for disk space needs before site installation. Some amount of this storage is assumed to remain as free disk space most of the time. This is used as buffer, if needed, in a recovery scenario, and for upgrade scenarios when free disk space is needed for setup package expansion. Your site may require additional storage for large amounts of data collection, longer periods of data retention, and for large amounts of software distribution content (which can also be stored on separate, lower throughput volumes).

Machine and disk configuration FAQs and examples

Q: How should I format the disks on my site server and SQL Server?

A: We recommend separating your Configuration Manager inboxes and SQL files on at least two different volumes if they reside on the same machine. This will allow for optimizing cluster allocation sizes on the volumes to match the characteristics of the I/O being performed.

For the volume hosting your sites server inboxes we recommend NTFS with 4K or 8K allocation units. ReFS writes 64k for even small files; because Configuration Manager has many small files ReFS can produce unneeded disk overhead.

For disks containing your various SQL database files we recommend either NTFS or ReFS formatting with 64K allocation units.

Q: How and where should I lay out my SQL database files?

A: Modern arrays of solid state drives (SSD) and Azure Premium Storage can provide very high IOPS on a single volume with very few disks. You typically add more drives to an array for additional storage not additional throughput. But if you require more IOPS than you can generate on a single volume (usually because you are using physical spindle-based disks) we recommend allocating 60% of the total recommended IOPS and disk space for the .MDF file, 20% for the .LDF file, and 20% for the log and data temp files. The .LDF and temp files can all reside on a single volume with 40% (20%+20%) of your allocated IOPS (the Configuration Manager product team performs most of its testing with this configuration).

By default, SQL will create one temp data file, but we highly recommend creating additional to avoid SQL locks and waiting for access to a single temp data file. Community opinions vary on the best number of temp data files from four to eight. In our testing we notice very little difference between four to eight, so all of our core testing and scalability validation is done with four *equally sized* temp data files. Your tempdb data files should be up to approximately 20-25% of the size of your full database.

Q: Any other recommendations for my disk setup?

A: When configurable, set RAID controller memory to approximately 70% allocation for write operations and 30% for read operations. In general, we recommend a RAID 10 array configuration for the site database. RAID 1 is also acceptable for small scale sites with low I/O requirements and/or if you use fast SSDs. With larger disk arrays configure spare disks to automatically replace failing disks.

Example: Physical machine with physical disks

For a collocated site server and SQL server supporting 100,000 clients you need 1200 IOPS for your site server inboxes and 5000 IOPS for the various SQL server files (see Table 1 above).

Your resulting disk configuration might look like this:

Drives ¹	RAID	Format	Volume Contains	Min IOPS Needed	Approx. IOPS Supplied ²
2x10k	1	-	Windows	-	-
6x15k	10	NTFS 8k	ConfigMgr inboxes	1700	1751
12x15k	10	64k ReFS	SQL .MDF	60%*5000 = 3000	3476
8x15k	10	64k ReFS	SQL .LDF, temp files	40%*5000 = 2000	2322

1 - Ignoring any additional disks for spares (recommended)

2 - From Appendix B

Q: I use Hyper-V on Windows Server; how should I configure the disks for my Configuration Manager virtual machines (VM) for best performance?

A: Hyper-V delivers similar performance as a physical server if hardware resources (CPU cores and pass-through storage) are 100% dedicated to the VM. Using fixed size .VHD or .VHDX disk files results in a minimal (1-5%) I/O performance impact. Using dynamically expanding .VHD or .VHDX disk files results in up to an approximate 25% I/O performance impact for the Configuration Manager workload. If you require dynamically expanding disks, you can compensate by adding an additional 25% IOPS performance to the array.

When running your Configuration Manager site server or SQL inside a VM, we recommend isolating the Hyper-V host OS drives from the VM OS and data drives.

See the Hyper-V product team's extensive documentation on optimizing VMs for more information: <https://docs.microsoft.com/windows-server/administration/performance-tuning/role/hyper-v-server/>

Example: Hyper-V VM-based site server

For a collocated site server and SQL server supporting 150,000 clients you need 1800 IOPS for your site server inboxes and 7400 IOPS for the various SQL Server files (see Table 1 above).

Your resulting disk configuration might look like this:

Drives ¹	RAID	Format ²	Contains	Min IOPS Needed	Approx. IOPS Supplied ³
2x10k	1	-	Hyper-V host OS	-	-
2x10k	1	-	(VM) site server OS	-	-
2xSSD SAS	1	NTFS 8k	(VM) ConfigMgr inboxes	1800	7539
4xSSD SAS	10	64k ReFS	(VM) Host SQL (all files)	7400	14346

1 - Ignoring any additional disks for spares (recommended)

2 - Fixed-size, pass-through .VHDX for the VM drive dedicated to the underlying volume

3 - From Appendix B

Q: Do you have suggestions for Configuration Manager environments in Microsoft Azure?

A: Start by reading our Configuration Manager Azure FAQ here: <https://docs.microsoft.com/en-us/sccm/core/understand/configuration-manager-on-azure>

Azure infrastructure as a service (IaaS) VM's that leverage Premium Storage-based disks can have very high IOPS. This could lead you to configure more disks for anticipated disk space needs rather than for additional IOPS.

Azure storage is inherently redundant so multiple disks are not required for availability, but disks can be striped in Disk Manager or Storage Spaces to provide additional space and performance.

Azure publishes specific information and recommendations in their online documentation to help you understand how to maximize your Premium Storage performance and run SQL servers in Azure IaaS VMs:

<https://docs.microsoft.com/en-us/azure/storage/storage-premium-storage-performance#optimizing-application-performance>

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sql/virtual-machines-windows-sql-performance#disks-guidance>

Example 1: Azure-based site server

For a collocated site server and SQL server supporting 50,000 clients you need 8 cores, 32 GB and 1200 IOPS for your site server inboxes and 2800 IOPS for the various SQL Server files (see Table 1 above).

Your resulting Azure machine might be a DS13v2 (8 cores, 56GB) with the following disk configuration:

Drives	Format	Contains	Min IOPS Needed	Approx. IOPS Supplied ¹
<standard>	-	Site server OS	-	-
1xP20 (512GB)	NTFS 8k	ConfigMgr inboxes	1200	2334
1xP30 (1024GB)	64k ReFS	SQL (all files ²)	2800	3112

1 - From Appendix B

2 - [Azure guidance](#) allows for placing the TempDB on the local, SSD-based D: drive, given it will not exceed available space and will allow for additional disk I/O distribution.

Example 2: Azure-based site server (for instant performance increase)

Azure disk throughput is limited by the size of the VM; the configuration in the Azure example above may limit future expansion or additional performance. By adding additional disks during

initial deployment of your Azure VM (with minimal upfront investment) this will allow you to upsize your Azure VM in the future for increased processing power. Planning ahead like this is a much simpler way to increase site performance as requirements change, instead of later needing to do a more complicated migration.

Let's change the disks we used the Azure example above to see how the IOPS change.

DS13v2

Drives ¹	Format	Contains	Min IOPS Needed	Approx. IOPS Supplied ²
<standard>	-	Site server OS	-	-
2xP20 (1024GB)	NTFS 8k	ConfigMgr inboxes	1200	3984
2xP30 (2048GB)	64k ReFS	SQL (all files ³)	2800	3984

1 - Disks are striped using Storage Spaces

2 - From Appendix B, performance limited by VM size

3 - [Azure guidance](#) allows for placing the TempDB on the local SSD based D: drive, given it will not exceed available space and will allow for additional disk I/O distribution

If more performance is needed in the future, you can upsize your VM to a DS14v2 which will double CPU and memory. The additional disk bandwidth allowed by that VM size will also instantly boost your available disk IOPS on your previously configured disks.

DS14v2

Drives ¹	Format	Contains	Min IOPS Needed	Approx. IOPS Supplied ²
<standard>	-	Site server OS	-	-
2xP20 (1024GB)	NTFS 8k	ConfigMgr inboxes	1200	4639
2xP30 (2048GB)	64k ReFS	SQL (all files ³)	2800	6182

1 - Disks are striped using Storage Spaces

2 - From Appendix B, performance limited by VM size

3 - [Azure guidance](#) allows for placing the TempDB on the local SSD based D: drive, given it will not exceed available space and will allow for additional disk I/O distribution

Other common Configuration Manager performance FAQs

SQL Server

Q: Is it better to run with SQL co-located with the site server, or run it on a remote server?

A: Both can perform adequately assuming the single server is appropriately sized, or network connectivity is sufficient between the two servers.

Remote SQL requires the upfront and operational cost of an additional server but is typical among the majority of large-scale customers. Benefits of this configuration include:

- Increased site availability options (for example, SQL Always On)
- Heavy reporting can run with less overhead to site processing
- Simpler disaster recovery (in some situations)
- Easier security management
- Role separation for SQL management (such as with a separate DBA team)

Co-located SQL requires a single server and is typical for most small-scale customers. Benefits of this configuration include:

- Cost saving (for example, fewer machines, licenses, maintenance, etc.)
- Fewer points of failure in the site
- Better control for planning downtime

Q: How much RAM should I allocate for SQL?

A: By default, SQL will use all available memory on your server, potentially starving the OS and other processes on the machine. It's important to explicitly allocate memory to SQL to avoid potential performance issues. On site servers co-located with your SQL server you need to ensure there is enough RAM remaining for SMSExec and other Configuration Manager processes, in addition to ensuring the OS has enough RAM for file caching and other operations. When running SQL on a remote server you can allocate the *majority* of the memory to SQL but not all. Review **Table 1** above for initial guidance. SQL Server memory allocation is generally recommended to be rounded to whole GB. Also, as RAM increases to "very large" amounts you can let SQL have a higher percentage. For example, when 256 GB or more of RAM is available, SQL can be configured for up to 95% as that still preserves plenty of memory for the OS. Monitoring the page file is a good way to ensure enough memory has been allocated for the OS and any Configuration Manager processes.

Q: Cores are cheap these days; should I just add a bunch of them to my SQL server?

A: You may run into memory contention issues when many (greater than 16) physical cores are available and there is not enough RAM on your SQL server. The Configuration Manager

workload tends to perform better when at least three to four GB of RAM per core is available for SQL. When adding cores to your SQL servers be sure to increase RAM in proportional amounts.

Q: Will SQL Always On impact my performance?

A: In general SQL Always On has negligible effect on performance of the system when sufficient networking is available between the SQL replica servers. In our experience you can have rapid database log .LDF file growth in a busy SQL Always On environment. However, log file space is automatically released after a successful database backup, so we recommend an additional step of adding a SQL job for the Configuration Manager database to perform a backup (for example, every 24 hours, and an .LDF backup every six hours). For more information about SQL Always On and Configuration Manager, including more about SQL backup strategies, see <https://docs.microsoft.com/en-us/sccm/core/servers/deploy/configure/sql-server-alwayson-for-a-highly-available-site-database>

Q: Should I enable SQL compression on my database?

A: SQL compression is not generally recommended for the Configuration Manager database. While there are no functional issues with enabling compression on a Configuration Manager database, test results don't show much size savings compared to the potential (sizeable) performance impact to the system.

Q: Should I enable SQL encryption on my database?

A: Any secrets stored in the Configuration Manager database are already stored in a secure manner but adding SQL encryption can add yet another layer of security. While there are not any functional issues with enabling encryption on your database, you may notice up to a 25% performance degradation depending upon the tables you choose to encrypt and the version of SQL you are using. Therefore, encrypt with caution, especially in large-scale environments. Also remember to update your backup and recovery plans to ensure you can successfully recover the encrypted data.

Q: What version of SQL should I run?

A: Supported versions of SQL can be found here: <https://docs.microsoft.com/sccm/core/plan-design/configs/support-for-sql-server-versions>. From a performance standpoint, all supported versions of SQL meet the required performance criteria. However, we have found that versions of SQL 2012, and versions of SQL 2016 or newer tend to outperform SQL 2014 in some aspects of the Configuration Manager workload. Additionally, running SQL 2014 at SQL 2012 compatibility level (110) improves performance in general. At installation time Configuration Manager databases running on SQL 2012 and SQL 2014 will be set to compatibility level 110, whereas versions of SQL 2016 or newer will be set to that SQL versions default compatibility level (eg: 130 for SQL 2016). If you upgrade SQL in-place compatibility levels will not be updated

until you install to the next major Configuration Manager current branch version. If you are experiencing unusual timeouts or slowness on certain SQL queries (in particular some customers have reported slowness in certain areas of the Admin Console especially when using RBAC), changing the SQL compatibility level on the Configuration Manager database to 110 (on SQL 2016 or later) may be beneficial. Running at SQL compatibility level 110 on SQL 2014 and newer versions of SQL is fully supported. For more information on using compatibility level 110 with Configuration Manager, see <https://support.microsoft.com/help/3196320/sql-query-times-out-or-console-slow-on-certain-configuration-manager-d>.

The Configuration Manager product team works closely with the SQL product team to test and optimize Configuration Manager functionality and performance on all supported versions of SQL. We occasionally find performance-related or other issues in some versions of SQL that might affect your Configuration Manager environment. As of January 2018, the Configuration Manager product team recommends *avoiding* the following SQL versions due to various known potential issues:

SQL 2012 SP3 CU1 to CU5
SQL 2014 SP1 CU6 to SP2 CU2
SQL 2016 RTM to CU3, SP1 CU3 to CU5

Q: Should I implement any additional SQL indexing tasks?

A: Yes, updating indexes as often as once a week and statistics as often as once a day improves SQL performance. There are scripts and additional information available from the Configuration Manager and SQL communities from 3rd parties to help optimize these tasks.

In large sites some SQL tables might be very big depending on your usage patterns and you may need to reduce or alter your maintenance approach for them one by one (for example, CI_CurrentComplianceStatusDetails, HinvChangeLog).

Q: Should I change MaxDOP settings on my database?

A: The Configuration Manager product team has tested with various MaxDOP settings and in general has found that leaving your setting at 0 (use all available processors) is optimal for *overall* processing performance in *most* circumstances.

Many Configuration Manager administrators follow the guidance in this SQL support article: <https://support.microsoft.com/en-us/help/2806535/recommendations-and-guidelines-for-the-max-degree-of-parallelism-confi>. On most modern large hardware this guidance tends to lead to a suggested maximum setting of 8. However, there are specific mentions in that article about running a lot of smaller queries compared to the number of processors, it may be beneficial to set it to a higher number. In our testing this is exactly what we have found in general with the Configuration Manager workload, so limiting yourself to 8 is *not* necessarily the best setting on larger sites when more cores are available. On SQL servers with greater than 8 cores we

recommend starting with a setting of 0 and only making changes if you are experiencing performance issues or excessive locking.

If you need to change MaxDOP to a specific value because you suspect you are encountering performance issues at 0, we recommend starting with a new value at least greater than or equal to the minimum number of cores recommended for that site's SQL server sizing. Going less than this value nearly always has negative performance implications based on testing. For example, a remote SQL server for a 100,000 client site needs at least 12 cores; if your SQL server has 16 cores in it, start your MaxDOP setting testing with a value of 12.

Antivirus

Q: Which folders on the site server (or other roles) should I exclude for antivirus software?

A: Take care when disabling antivirus protection on any system. In high volume and secure environments, we recommend disabling "active monitoring" for optimum performance.

Additional information on recommended antivirus exclusions can be found here:

<https://support.microsoft.com/en-us/help/327453/recommended-antivirus-exclusions-for-configuration-manager-2012-and-cu>

Secondary sites

Q: When should I use full SQL Server instead of SQL Express on my secondary sites?

A: SQL Express does not have any significant performance implications on secondary sites and for nearly all customers it is completely adequate. It is also very easy to deploy and manage and is thus the recommended configuration for nearly all customers at any size.

There is one situation where a full SQL Server installation might be needed. It is possible to exceed the 10 GB size limit of SQL Express if you are going to have a very large number of distribution points and packages/sources in your environment. If you anticipate having the <number of packages> times the <number of distribution points> to be greater than 4,000,000 (for example, 2,000 DP's with 2,000 pieces of content across the entire hierarchy) then you should consider using full SQL Server at your secondary sites to avoid running into potential database size constraints.

WSUS

Q: What can I do to make WSUS perform better when it's used with Configuration Manager?

A: Changing a few key IIS settings (WsusPool Queue Length and WsusPool Private Memory limit) can improve WSUS performance even on smaller installations. See this link for more information: <https://docs.microsoft.com/sccm/core/plan-design/configs/recommended-hardware>.

You should also make sure you have the latest updates installed for the Operating System running WSUS:

Windows Server 2012: Any non "Security only" cumulative update released October 2017 or after. (KB4041690)

Windows Server 2012 R2: Any non "Security only" cumulative update released August 2017 or after. (KB4039871)

Windows Server 2016: any non "Security only" cumulative update released August 2017 or after. (KB4039396)

Q: What type of maintenance should I be running on my WSUS Servers?

A: An excellent article has been posted on the official Configuration Manager blog here: <https://blogs.technet.microsoft.com/configurationmgr/2016/01/26/the-complete-guide-to-microsoft-wsus-and-configuration-manager-sup-maintenance>.

Monitoring

Q: I want to setup basic performance monitoring for my site; what should I be watching?

A: Traditional server performance monitoring works very effectively for general Configuration Manager. You can also leverage the various System Center Operations Manager management packs for Configuration Manager, SQL Server, and Windows Server to monitor basic health of your servers. In addition to and/or in conjunction with these you can also directly monitor the Windows Performance Monitor (PerfMon) counters provided by Configuration Manager. Monitoring the backlogs in the various inboxes is a good early warning sign of potential performance issues or backlogs in a site.

Appendix A: Measuring IOPS (disk writes)

By using the industry standard tool Diskspd we are providing standardized suggestions for the IOPS required for various sized Configuration Manager environments. While certainly not exhaustive they do provide a very simple and reproducible way to get an estimate of the throughput of your servers' disk subsystem versus the recommended IOPS provided in **Table 1** above. The test methodology and command lines below are what was used when calculating the suggested IOPS numbers provided in **Table 1** above.

In addition to giving you the steps below that allow you (or your hardware vendor) to reproduce these numbers, we also ran these tests against a variety of hardware in our labs and provide the results in **Appendix B**. The data in Appendix B can be used to provide a *rough* starting point when designing the storage subsystem for a new environment from scratch.

The Diskspd utility is available here: <https://gallery.technet.microsoft.com/DiskSpd-a-robust-storage-6cd2f223>

Testing methodology

Two runs of the tool are performed in sequence for the volume that you want to test. The first test performs 64k-size, random write operations for one minute. This test is intended to ensure the controller cache is loaded and disk space is allocated (in case the volume is dynamically expanding). Discard the results of the first test. The second test should *immediately* follow the first test without delay and will perform the same load for five minutes.

The specific command lines we use are shown below. In this example we are testing the G:\ volume.

Run Diskspd from an elevated command prompt, make sure you have at least 100 GB of free disk space, and disable any programs that might interfere or cause extra load on the disk (for example, active antivirus scanning of that directory, SQL, SMSExec, etc.).

```
DiskSpd.exe -r -w100 -t8 -o8 -b64K -c100G -d60 -h -L G:\test\testfile.dat
DiskSpd.exe -r -w100 -t8 -o8 -b64K -c100G -d300 -h -L G:\test\testfile.dat
del G:\test\testfile.dat
```

Review the output from the second test to find the total IOPS ("I/O per s" column) as **highlighted** below.

Total IO thread	bytes	I/Os	MB/s	I/O per s	AvgLat	LatStdDev
0	9651814400	147275	30.68	490.92	16.294	10.210
1	9676652544	147654	30.76	492.18	16.252	9.998
2	9638248448	147068	30.64	490.23	16.317	10.295
3	9686089728	147798	30.79	492.66	16.236	10.072
4	9590931456	146346	30.49	487.82	16.398	10.384
5	9677242368	147663	30.76	492.21	16.251	10.067
6	9637330944	147054	30.64	490.18	16.319	10.249
7	9692577792	147897	30.81	492.99	16.225	10.125
total:	77250887680	1178755	245.57	3929.18	16.286	10.176

Appendix B: Sample disk configurations

Use the data in this appendix to provide a *rough* starting point when designing the storage subsystem for a new environment from scratch.

After using the test steps in **Appendix A**, the table below shows the test results across various disk subsystems including spindle and SSD-based hard drives in various configurations in our test labs.

Hardware is always improving; expect newer generations of hardware and different hardware combinations (SSDs, SANs, etc.) to exceed the performance stated below. We share these results as a basic starting point to consider when designing a server or discussing with your hardware vendor.

Physical machines & Hyper-V

Disk Type	Disk count (not including +1 disk as a spare)	RAID	IOPS Measured
15k SAS	2	1	620
15k SAS	4	10	1206
15k SAS	6	10	1751
15k SAS	8	10	2322
15k SAS	10	10	2882
15k SAS	12	10	3476
15k SAS	16	10	4236
15k SAS	20	10	5148
15k SAS	30	10	7398
15k SAS	40	10	9913
SSD SATA	2	1	3300
SSD SATA	4	10	5542
SSD SATA	6	10	7201
SSD SAS	2	1	7539
SSD SAS	4	10	14346
SSD SAS	6	10	15607

All disks are attached to an enterprise class disk controller, formatted with 64k clusters, and configured in RAID 1 or RAID 10 as noted. In addition to the RAID array disk count at least one spare disk is used.

Hardware used in these tests

None of the information below should be construed as a recommendation of any specific hardware model or manufacturer, they are simply the devices used for this specific example.

Disk Type	Model	Raid controller	Cache Memory and configuration
15k RPM SAS HD	HP EH0300JDYTH	Smart Array P822	2GB, 20% Read / 80% Write
SSD SATA	ATA MK0200GCTYV	Smart Array P420i	1GB, 20% Read / 80% Write
SSD SAS	HP MO0800JEFPB	Smart Array P420i	1GB, 20% Read / 80% Write

Azure machine versus disk performance

Azure VM	Azure disk	Disk count	Available Space*	IOPS Measured	Limiting Factor**
DS2/DS11	P20	1	512 MB	965	Azure VM size
DS2/DS11	P20	2	1024 MB	996	Azure VM size
DS2/DS11	P30	1	1024 MB	996	Azure VM size
DS2/DS11	P30	2	2048 MB	996	Azure VM size
DS3/DS12/F4S	P20	1	512 MB	1994	Azure VM size
DS3/DS12/F4S	P20	2	1024 MB	1992	Azure VM size
DS3/DS12/F4S	P30	1	1024 MB	1993	Azure VM size
DS3/DS12/F4S	P30	2	2048 MB	1992	Azure VM size
DS4/DS13/F8S	P20	1	512 MB	2334	P20 disk
DS4/DS13/F8S	P20	2	1024 MB	3984	Azure VM size
DS4/DS13/F8S	P20	3	1536 MB	3984	Azure VM size
DS4/DS13/F8S	P30	1	1024 MB	3112	P30 disk
DS4/DS13/F8S	P30	2	2048 MB	3984	Azure VM size
DS4/DS13/F8S	P30	3	3072 MB	3996	Azure VM size
DS5/DS14/F16S	P20	1	512 MB	2335	P20 disk
DS5/DS14/F16S	P20	2	1024 MB	4639	P20 disk
DS5/DS14/F16S	P20	3	1536 MB	6913	P20 disk
DS5/DS14/F16S	P20	4	2048 MB	7966	Azure VM size
DS5/DS14/F16S	P30	1	1024 MB	3112	P30 disk
DS5/DS14/F16S	P30	2	2048 MB	6182	P30 disk
DS5/DS14/F16S	P30	3	3072 MB	7963	Azure VM size
DS5/DS14/F16S	P30	4	4096 MB	7968	Azure VM size
DS15	P30	1	1024 MB	3113	P30 disk
DS15	P30	2	2048 MB	6184	P30 disk
DS15	P30	3	3072 MB	9225	P30 disk
DS15	P30	4	4096 MB	10200	Azure VM size

*All disks are formatted NTFS 64k cluster size, and rows with more than one disk are configured as a striped volume via the Windows Disk Management utility.

**Azure disk performance is dependent on several factors including the size of the Azure VM and the number and type of disks used. Azure is also constantly adding new machine types and disk speeds that are different offerings from the above chart. For more information about Configuration Manager running on Azure, as well as additional information on understanding disk I/O on Azure, see this FAQ:

<https://docs.microsoft.com/en-us/sccm/core/understand/configuration-manager-on-azure>

Update History:

Jan 2018 – initial publication.

Mar 2018 – corrected 4,000 to be 4,000,000 in section on SQL Express and Secondary Sites, additional pointers for WSUS related information added, other minor clarifications and updates