ARGENT
ENCYCLOPEDIA

SharePoint Monitoring Best Practices

Contents

Overview

Microsoft Office SharePoint Server (MOSS) 2007 and Windows SharePoint Services (WSS) 3.0 have opened the doorway to large scale web systems for companies worldwide like never before.

As such, businesses are now leveraging the SharePoint development framework to house numerous business critical custom web applications, in addition to the countless out of the box functions provided by both MOSS and WSS SharePoint versions.

Quite simply, the ease of mass implementation has exponentially increased the demand to monitor appropriately these critical business applications required to ensure seamless functionality and optimal performance. Whether stand alone or integrated platforms, this paper intends to illustrate some of the best practices for monitoring your SharePoint environment(s).

Common Hurdles For SharePoint Monitoring

There are several aspects of monitoring SharePoint system metrics that tend to cause questions. From a high-level overview, these questions are as follows:

- Where is the data?
- What data should be collected?
- What are good performance values for the farm function of focus?
- How can the collected data inform the business?
- When should the business be concerned about the data values returned?

Where Is The Data?

One of the most common hurdles that administrators face is that they often are not developers. As MOSS and WSS 3.0 don't come out of the box with terribly granular benchmark and reporting tools, administrators are left to hook into either the off-the-shelf web services (for light reporting) or object model (for deep metric gathering). These two methods provide varied gateways to the backend database structure and performance information. Attaching to your data, however, is just the beginning.

What Data Should Be Collected?

Based on the SharePoint function of focus, administrators determine whether the business would best benefit from quantitative (how much?) or qualitative (what type?) information. Once hooked into the appropriate data store, identification of the monitored data pertinent to business initiatives is key. Classifying the returned data values feeds from and iteratively feeds into performance benchmark analysis.

What Are Good Performance Values For The Farm Function Of Focus?

All too often, SharePoint administrators find their businesses' focus toward off-the-shelf benchmark values. The classic realization however, is that these nearly never incorporate data types even remotely similar to those the business may be using. Therefore, the determination of a "good performance value" should reflect the following:

- The resident content
- The implementation of their related farm functions
- Each SharePoint farm design structure created to meet identified requirements.

In addition, the business often cannot establish their benchmarks until a system (often QA environment) is implemented to meet said requirements. It should also be understood by the administrators and the business they support that these benchmarks are, and will be in flux.

Informed by system integration points and overall farm function performance, these will continue to change over time.

Obviously, each farm function can and should be tuned to meet optimal performance. This, however, often cannot be completed effectively until the baseline is established by the administrators and iteratively agreed upon by the systems' stakeholders.

How Can The Collected Data Inform The Business?

Because the likelihood that business stakeholders will have phenomenal psychic powers is slim, SharePoint administrators need the ability to provide gathered metrics in an acceptable fashion. Again, falling back to web services and object model development for data collection, several firms provide packaged toolsets available in several formats including but not limited to:

- SQL Reporting Services packages
- STSADM extension packs
- Powershell reporting toolkits
- Web driven dashboard systems
- GUI based metric and reporting tools

All of these are viable solutions for reporting monitored performance metrics. Some have interactive reporting interfaces, some have slick, easy-to-read charts and graphs, while others get right down to the numbers. An administrators' best bet for adoption is typically a combination of these. The tools sets leveraged will vary by business, size and scope of their SharePoint implementation(s), and their respective budgets. In short, administrators should be able to gather quickly metrics in a fast fashion and business stakeholders should be able to understand quickly the bottom line.

When Should The Business Be Concerned About The Returned Data Values?

While administrators monitor performance metrics constantly, these should be evaluated by the business on a regular basis or as required by performance threshold. The operative word here is ‘threshold.’ Realistically, very few businesses today have the capacity to review regularly any system performance metrics even for tier-one mission critical applications. As such, each SharePoint implementation should have a level of governance establishing things like review cycle (to be discussed in a later chapter) and the spill over.

As a system is reviewed, optimized, and performs well, businesses often become complacent (or worse, reactive), only reviewing metrics after the system is performing poorly. Establishing thresholds and an escalation pattern per monitored farm function aids the business in catching poor functional performance before becoming a threat to the overall system performance.

These thresholds (and administrators’ corresponding response(s)) should also be accommodated for in each implementation’s respective governance structure.

Common Repercussions For Incorrect Monitoring

Every time the discussion of SharePoint monitoring is broached, it is inevitably followed by an exploration of the conversation participants' various 'learning opportunities.' Typically, system administrators tend to have very similar results regardless of the SharePoint function they consistently monitor. The significant commonalities identified for nearly all reactive approaches to SharePoint farm functions follow a repeatable, very recognizable, and highly avoidable pattern as outlined here:

1. Farm function becomes sluggish on system response or task completion
2. Function performance degradation becomes noticeable to business users, either directly or through dependencies
3. Function fails completely within sanctioned time window if at all
4. Administrators check logs (event, 12-hive, etc.), hardware and DB performance, network latency, etc. until finally determining the respective solution for the given implementation.

Illustrated here is the lateness of the administrators' knowledge and reactivity of their efforts. The very first bullet point above should have been caught and addressed as it occurred.

Content Databases And Their Content

SharePoint performance problems are very rarely related to hardware or infrastructure issues. In fact, the common ground is typically due to one or more of the following:

- The type of content.
- Where the content lives.
- How the content is rendered.

The Type Of Content

SharePoint administrators know that content databases are simply containers to house any number of data types (page content, static files, streaming files, workflows, etc.). The default option is to leverage a single content database per site collection. This works well for small implementations, as it is clean and easy to manage. As your farm and its respective applications grow, the demand becomes ever more pressing to separate data type by content database.

Where The Content Lives

The separation of data types allows each content database to live potentially on a SQL server optimized for its respective data type. Administrators will often overlook this requirement and field significant performance hits because of it. To illustrate, consider the following example.

o A line of business application utilizes Forms Server Integration and Excel Calculation Services. These functions are working well together without impact to other farm functions (i.e. Search Crawling/Indexing).

o A streaming media training video is added to the forward facing application landing page but is just placed in a separate document library.

o The content database optimized for InfoPath and Excel functionality is being used to host significantly more I/O traffic than before. This has now impacted the Forms and Excel calculation timeliness and in effect, the Search Crawl/Indexing function.

Placing the streaming media content, or its relative document library into another content database (on another server, if possible), frees up the original database I/O for the more critical application functions. As one can imagine, this issue becomes exponentially more problematic as the SharePoint farm, and its functions, grow.

How The Content Is Rendered

Another consideration with this example is the rendering of different data types. Were this application's resident Excel files to depend on client side calculation, they could potentially decrease the I/O or processing footprint on the database. Likewise, were the training video to be a downloadable file rather than run as streaming content, it would also reduce this footprint (though not quite so elegant). As with countless other variables to consider, the method used to render content becomes pertinent to the decision of how (and where) the data should be stored.

Monitoring The Matter

These are just a few considerations for the application structure throughout its lifecycle. Remember, the SharePoint framework is designed to ease changes in this structure as each implementation grows. List item one in the example above also illustrates the necessity for qualitative monitoring. Gathering metrics by data type illustrates another. This administrator could identify/isolate a data type, its location, or render method prior to affecting other farm functions. By taking a proactive stance, the administrator also significantly decreases the potential for overall system performance degradation. Monitoring several content/system properties such as performance metrics and data type information are a few ways to do so. Monitoring alone is useless, however. The business needs to have a method to meter the unit data collected. This is where monitoring thresholds come into play.

Thresholds

Monitoring thresholds are the mechanisms for establishing and maintaining SharePoint farm content stability and performance. In essence, they are the measuring stick administrators use to answer our combined question, “How many of what type?”

Outlined by best practice strategies, establishing a staged threshold monitoring model empowers its corresponding phased response approach at each of the following five basic stages:

1) Acceptable Operation
2) Approaching Limits
3) At Limits
4) Exceeding Limits
5) Major Overload

With these threshold stages, administrators will have business-defined processes outlining their associated mitigation steps. Each process is focused toward keeping monitored values (“How many of what type?”) within the acceptable limits at each stage. In addition, each stage will also be associated with mitigation tactics to increase optimization of farm functions through respective performance metrics.

Essentially, performance and results validity values are measured against these thresholds on a regular (or real-time) basis. Were values to fall outside the ‘Approaching Limits’ threshold stage, an administrator would then take the corresponding mitigation steps to prevent the system from reaching the ‘At Limits’ stage and if possible optimize the content or its container for better performance.

The level of proactive monitoring and mitigation rigor required for each threshold stage must be defined along with respective strategies in the governance structure.

Governance Structure As It Pertains To SharePoint Monitoring

As monitoring thresholds undoubtedly differ per farm environment (and possibly per web application) the governance structure becomes the mechanism for iterative review of monitoring practices and procedures associated with each monitored facility. It takes our monitoring question, "How many of what kind?" and reshapes it to a statement. The ideology here is to leverage the governance structure to analyze the state of your thresholds with established controlled response. In essence, this ensures the statement accommodates action (preferably, proactive rather than reactive). Now our question has changed to a statement more like, "Before we accrue this many, of this kind, we need to . . ."

The conversion to an action statement simply creates a baseline of activity for stakeholders, illustrating intent. However, the governance structure is what facilitates empowerment of the SharePoint administrator to accomplish the action defined by the stakeholders in these actions statements. As such, administrators will find it necessary to have at least one plan of action per monitoring threshold, and affirm these with the business stakeholders through the applications' associated governance structure(s).

Monitor The Right Areas Of SharePoint For Your Business

Properly gauging the brunt of the primary SharePoint based business application is a decent start to establishing a monitoring strategy, but often doesn't cover the whole picture. As stated previously, the SharePoint framework is designed to be easily expandable and as such has an exponential growth pattern. Gauging application instances in a farm, however large, is likely never fully inclusive of their current or future peers nor dependant farm functions.

As such, a more inclusive monitoring strategy must also accommodate for farm function. Yes, content and immediate application functionality are applications. However, farm functions, such as searching/indexing, Forms Services, Excel Calculation Services, etc. may have far more significant impact to each application, while far less visibility to the application consumers. For this reason alone, taking a farm function focused approach enables the monitoring strategy to be far more scalable addressing each application as yet another function. In doing so, administrators are best able to gauge business impact by function and determine the proper monitoring and mitigation tactics associated with each.

Simply put, every farm is different. There are no hard and fast rules outlining what and what not to monitor. Because of this, farm functions are the heart of every implementation's monitoring strategy. Building on the core monitoring foundation, every business will evaluate the volumetric values (quantitative/qualitative) and corresponding actions per application, and accommodate for both as previously discussed. Administrators should start with core farm functions and grow their monitoring strategy with their farm, including each addition as yet another function of the farm as a whole.

Inside The Core

Administrators typically find there are several key farm functions that, if deployed, every implementation should monitor as a starting baseline. Looking deeper into these core functions, one should note that most are found in the Shared Service Provider model (SSP). As such, several of these may not impact WSS implementations. This is a general list, which is not intended to be all-inclusive. On the contrary, it is intended to establish a monitoring pattern for administrators. By knowing these functions, their farm locations (both configuration and execution), and their peer functions, administrators have the edge on monitoring and maintaining farm performance. Each of these functions has associated farm performance impacts to be monitored, which are discussed as follows:

- My Sites (and self service site creation)
- Excel Calculation services
- InfoPath Forms Server Services
- Enterprise Search Crawling
- Enterprise Search Indexing
- User Profile Import and Audience Compilation

My Sites (And Self Service Site Creation)

Both self service and My site collections have the tendency for exponential growth. Thankfully, SharePoint is able to accommodate for storage and capacity concerns via site quota templates and separation of content databases. Now, rather than monitoring size of the sites or volume of content within, administrators can focus monitoring efforts on the rapid growth and quantity of both self served and My sites. Regardless, content is content and still uses system resources. However, performance optimization of content becomes simpler by monitoring and maintaining the quantity of these site types.

Excel Calculation Services

Excel calculation services (and by right access integration points) do exactly what they are titled to do. A farm resource is leveraged for formulaic calculation at the farm layer, as opposed to the client layer. More so in small implementations, the calculations are often deployed to a single farm member server. This can be remarkably taxing for any farm member server if not configured and monitored properly, regardless of deployment scale. Monitoring focus areas are time to calculation completion, associated hardware utilization during complex calculations, and quantity of trusted locations.

InfoPath Forms Server Services

InfoPath has so many different deployment strategies and implementation tactics that associated monitoring strategy is difficult to standardize. Common areas of contention are centralized data connections, client sideform editing, and communication of excess nested form sections. Note that these monitor points require development of a web service or custom code to the object model (or integrated off-the-shelf packages).

Enterprise Search Crawling

Enterprise search monitoring is primarily focused on crawl time to completion and crawl record count. Crawl record count is typically directly proportional to crawl database size and can be monitored from that aspect as well. Monitoring threshold stages should focus on increased crawl rule optimization based on previously captured performance metrics while paying close attention to variance at each iterative pass.

Common optimization strategies leveraging these monitoring tactics heavily employ search scope separation by data/content type and alternative or distributed schedule for each. They are often pursued in the following order:

- Refactoring the crawl rule to slim (or tune) results
- Performance tuning the pertinent content databases for the resident content
    - o e.g. streaming media files vs. office documents
- Combining crawl rules to capture and index a larger, more relevant results set
- Amending crawl schedule to incorporate additional farm functions

Regardless of the response path an administrator takes to maintain the threshold limit iteratively determines the limit(s) for each successive stage of the monitoring process. On the other hand, time to completion data is resident in the SSP and Central Administration content databases and, again, must be consumed via web service or object model tools. This poses potentially problematic for non-developer savvy administrators and will likely call for packaged monitoring products.

Enterprise Search Indexing

Search indexing is typically a fairly lightweight service. However, farm deployment strategy becomes an area of contention as the creation and consumption of the index file by farm member server(s) can cause severe traffic related performance impact.

Impact is evident in time to return rendered results at the consumer interface and as impacted will cause sluggish return for large results sets and compound searches (even simple combinations). Powershell, while typically extremely beneficial, can be either your best friend or your worst nightmare when trying to capture consumer experience benchmark data.

As such, monitoring result set render time is a function of many packaged software suites and script packs as well. Thus, either makes a valuable addition to every administrator's toolkit.

User Profile Import And Audience Compilation

User Import functionality is essentially a search crawl function focused at an authentication provider of choice. Highly customizable for LDAP queries, the compound rule sets are often a point of contention but this is typically a very lightweight service (hence LDAP design). Similarly, audience compilation is essentially a combination of the search scope definition, indexing, and security trimming. Therefore, employing audiences can create greater impact to both consumer experience and farm performance due to the following areas:

- Rule complexity
- Audience quantity
- Member quantity
- Audience compilation time

Administrators should focus monitoring efforts respective to profile import toward time to completion and compound query complexity. Efforts respective to audiences should primarily focus on audience membership quantity and proper implementation (or scoping) of audience to site functionality.

Outside The Core

With all this covered, administrators often ask the question, "What's left?"

Well, everything else of course. To reiterate, the SharePoint framework is remarkably flexible and accommodates for countless development methodologies leveraging innumerable integration factors. Offthe- shelf core functions, while complex in their own right, don't hold a candle to the yet ventured complexity of any applications still to be developed.

Nearly every facet of this complexity adds monitoring points for all deployed applications, canned or custom. Typically, the integration points are of interest. Again, this list is intended for administrators to pattern from, giving them the upper hand. Monitoring of a few common examples of integration factors will be discussed as follows:

- Business Data Catalogue
- Web services and Data connection libraries
- Object model
- Workflow

Business Data Catalogue

Leveraged for the direct access to upstream application resident databases, Business Data Catalogue (BDC) functionality provides SharePoint applications with query capability, thus fueling event receivers. This is a spectacular tool when leveraged appropriately and an outright danger to application performance when deployed incorrectly. BDC applications pulling too large of a result set cause event receivers to halt on process. If the event receiver happens to trigger a timer job, this can cause catastrophic performance problems with queued farm timer jobs as well.

Another consideration is BDC authentication. If the BDC application is leveraging encryption, impersonation, or calls to dependant binaries (rare but possible) requiring security, these can also cause slight performance impacts but typically only with consumer experience. Therefore, monitoring strategy should incorporate both query functions such as string length and results set count, as well as authentication or decryption latency upstream system connections.

Web Services And Data Connection Libraries

Essentially ODBC connections for server side processing, Data connection files are used by site resident data types (Excel files, InfoPath forms, workflows, etc.) to connect to both up and downstream data sources. Probably the most common usage is web service queries. Utilized for speed of throughput, the possibility for leveraging many Data Connections to do mass communication increases greatly. Monitoring should simply focus on the number of connections called simultaneously and transfer size.

Object Model

Object model '"code behind" for custom applications is hands down the largest potential farm performance impact to nearly all SharePoint implementations. Errant or poor code and non-SharePoint deployment practices are evident in so many component layers, it is difficult for most administrators to know where to start troubleshooting when affected.

As such, each and every custom SharePoint application's monitoring strategy should accommodate for design objects respective to object model dependencies. These often include the following:

- Binary deployment
- .config file changes
- Custom workflows, event receivers, actions, etc.

  Note: Non "feature" or "solution" based deployment of these design objects is also potentially problematic to farm replication and performance.

Monitoring heavy logging, infrastructure performance, and high database I/O during testing phases, for any custom application normally highlights pertinent issues but monitoring tactics typically mandate an ongoing heartbeat for these areas.

Workflow

Available to SharePoint in two fashions, workflows are deployed as SharePoint Designer workflows or Visual Studio workflows (Windows Workflow Foundation). Dependant application functionality determines the development and deployment strategies respectively. Fortunately, administrators' monitoring strategy is typically the same for both. Workflow monitoring should encompass event receiver response time, timer job time to completion, and increased logging.

In summation, continuing to expand monitoring strategy beyond core functions to accommodate additional application complexities is most commonly driven by either implementation scale or disaster recovery. Applications and respective host farms, which expand to national or global demographics, demand impeccably consistent performance monitoring and maintenance.

Scaled To Variance

Scaling to regional, national, or global markets means applications must render to multiple demographics and most likely with multiple key driven content values. Enter . . . variance. While a relatively straightforward farm function configurable to suit, and an extremely elegant end user experience, this function can be heavily taxing if not carefully monitored. For example, if this function were configured for three markets with three different languages it would, in essence, contain three different content values per asset, per market.

Every page that needs rendered to English, Kanji, and French will effectively have three page blob objects. Thus, the SharePoint content databases hosting these variances often suffer the performance hit if not properly monitored and optimized. Language translation is just one usage for variations, however, and is likely of small concern.

SharePoint implementations are often growing to additional regions within the same language context as well. This generates an immediate demand for availability of regional variations of information presented to its intended demographic. Audience controlling functions well but isn't nearly as robust as versioning capabilities.

As an application grows into a requirement for variance, adding this function to the established monitoring strategy is an absolute must. These robust capabilities bring complexity to the monitoring strategy through the additional data watch points at object version counts, content size variations, and associated database optimization.

Of additional concern to most organizations, considerations for replication strategy and optimization thereof often develop with market or demographic expansion and typically pertain to disaster recovery and environmental promotion.

Scaled To Replication

Implementations grow for several reasons. Growth to additional markets, growth to support tier one line of business applications or growth respective to infrastructure architecture all eventually lead to some form of system replication or syndication. True replication is typically done at the SQL backend layer to support Disaster Recovery (DR) and regional implementations. Monitoring of SQL replications is fairly well accommodated for via the SQL Server tool sets, however, SharePoint administrators may monitor timer job queues for indications of impact to farm functionality due to SQL replication issues.

Other replication practices often employed for environmental promotion (e.g. development, to test, to production) resident in SharePoint include content deployments and backup/restore methodologies. Monitoring of these SharePoint functions tends to veer from a quantitative based focus and are largely dependent on qualitative results such as farm logging and time to completion values. Common remedies for poor SharePoint replication performance are often driven by load distribution or dedication of the service to underutilized farm member server(s).

Thoughts About Backups

SharePoint backup methodologies are often a point of contention as companies tend to employ both SQL Server and SharePoint system backup tactics to establish the safety net for short-term retention and catastrophic failure.

This amalgamation of the two schools of thought indeed uses twice the storage space, but more importantly will have varied performance based on design, storage location and schedule.

SQL Server backups are most often scheduled and kept per established business retention policies. As such, they are likely placed in a scraped location and the scraped data sent to an offsite location. This process rarely has performance problems but may have size constraints on the dump location and may justify a monitoring practice with established thresholds to maintain proper functionality.

SharePoint backups are often more prone to performance degradation, likely due to these common factors:

- Backup is dumped to a single low performing drive
- Dump drive may often be heavily fragmented due to repeated scheduled backup activity
- Backups may run from or dump to a machine running parallel farm functions (Search crawl and Indexing, document conversion services, etc.)

All three of these monitoring points are a great start for monitoring SharePoint backup functional performance and ensuring greater data integrity. Log files and system resource counters are the best location for tell tale signs of SharePoint backup performance impact and thus should be included in the monitoring strategy thresholds as a start.

Conclusion

As stated previously, the most successful monitoring strategies are comprised of practices defined by governance structure, in coordination with the business. They nearly always employ several monitoring tool sets and methodologies, polling heartbeat and snapshot information as close to real time as possible from data points that potentially number into the thousands. Even small-scale implementations should keep uptime and performance monitoring in mind.

About Argent Software

As the premier enterprise-level monitoring suite available to SharePoint administrators to date, Argent for SharePoint is designed specifically to handle discussion points outlined throughout this document and more.

Developed to accommodate for implementation scales from single server to global distributions, Argent for SharePoint beautifully embraces threshold mentality to address easily business-defined monitoring, mitigation, and escalation paths. In addition, the product supports business growth within the SharePoint technology platform with expansive configuration flexibility.

There are few products on the market capable of incorporating monitoring facets such as web service communication, object model interpretation, system resource factors, and replication strategy right off the shelf. Argent for SharePoint is the product no SharePoint administrator should be without.

Note: ArgSoft Intellectual Property Holdings Limited has created this White Paper for informational purposes only. ArgSoft Intellectual Property Holdings Limited makes no warranties, express or implied, in this document. The information contained in this document is subject to change without notice. ArgSoft Intellectual Property Holdings Limited shall not be liable for any technical or editorial errors, or omissions contained in this document, nor for incidental, indirect or consequential damages resulting from the furnishing, performance, or use of the material contained in this document, or the document itself. All views expressed are opinions of ArgSoft Intellectual Property Holdings Limited. All trademarks are the property of their respective owners.