SharePoint Performance Monitoring
Webinar
Today’s Presenter

About Me

- Owner and Lead Bitsmith of Bitstream Foundry
- 50% dev, 50% admin = 100% confused
- AR/VR enthusiast and wannabe developer
- Former polymer chemist
- CTO for a non-profit mental health awareness organization (http://www.schizophreniaoralhistories.com)
- Desktop DJ (http://www.bunkertuneage.com)
- Husband to a wonderful woman (Tracy)
- Father to two beautiful twins (Brendan & Sabrina)
- Coffee lover and occasional donut eater ...

Sean McDonough
Bitstream Foundry LLC
What We’ll Be Covering

1. Some Introductory Words
2. Farm Environments
3. Tools and Monitoring Servers
4. Page Performance Monitoring
5. Questions & Answers
6. References
Farm Environments
Yes, I said *farm*, not *stamp*

- Subtle distinction, but it means we’re likely on-premises ...
  - No SharePoint Online / Office 365
  - Unless you’re on a “farm in the cloud”
Yes, I said *farm*, not *stamp*

- Subtle distinction, but it means we’re likely on-premises ...
  - No SharePoint Online / Office 365
  - Unless you’re on a “farm in the cloud”

- Why on-premises?
  - Significant surface reduction for monitoring in the cloud
  - It’s “someone else’s” problem (i.e., a value-add for consumers)
  - Administrative APIs very limited vs. on-premises
  - Limited tools (no perfmon, developer dashboard, etc.)
  - In short: we can’t get at the counters and logs we need!
“An ounce of prevention is worth a pound of cure.”

When you have the luxury of starting from scratch, you can get the basics right.
Without a properly configured SQL Server environment, no amount of SharePoint troubleshooting will amount to anything.

So, some things to bear in mind …

- If virtualizing, then minimize abstractions
- Choose an appropriate storage sub-system
- Don’t skimp on disks!
- Put your I/O where you need it
A Smart Investment

“The Ultimate SharePoint Performance Guide” by Vlad Catrinescu and Gokan Ozcifci

Special link: https://leanpub.com/SharePointPerformanceGuide/c/SysKit
Tools and Monitoring Servers
Why do we monitor performance? Reasons typically fall into one of the following three categories:

- We are seeking to understand why our SharePoint environment is underperforming
  - Troubleshooting!
- We want to ensure that we have enough headroom to scale and grow as desired.
  - Capacity!
- We want to quantify changes we’ve made to our farm in terms of performance
  - Improvements!
We’re looking for the source of a performance problem. Where should we start?

Performance issues typically originate in at least one general sub-system:

- Memory
- Network
- Processor (CPU)
- Storage (Disk)

Of course, SharePoint problems often muddy the waters by spanning more than one category.
Recommendation: start with monitoring the server(s) over time to gain an understanding:

- First understand “the normal state” of a server
- Then observe the server when a problem occurs
Recommendation: start with monitoring the server(s) over time to gain an understanding:

- First understand “the normal state” of a server
- Then observe the server when a problem occurs

Establishing a baseline when your environment is running normally (and non-stressed) is critical.

- Baselines provide a reference point
- Without a baseline, all measurements are simply relative to one another
Many different tools at our disposal:

- Farm Health Analyzer
- Event Viewer
- ULS Viewer
- Fiddler
- Developer Dashboard
- Wireshark
- Diskspd
- CrystalDiskMark
Today’s focus for performance monitoring is on counters

- **Specific** performance counters that can help direct further investigation and keep us out of the weeds
Today’s focus for performance monitoring is on counters

- **Specific** performance counters that can help direct further investigation and keep us out of the weeds

How do we view performance counters?

- **Windows Performance Monitor** (perfmon.exe)
Today’s focus for performance monitoring is on counters

- **Specific** performance counters that can help direct further investigation and keep us out of the weeds.

**How do we view performance counters?**
- Windows Performance Monitor (perfmon.exe)
- Windows Resource Monitor (resmon.exe)
Today’s focus for performance monitoring is on counters
  - **Specific** performance counters that can help direct further investigation and keep us out of the weeds

How do we view performance counters?
  - Windows Performance Monitor (perfmon.exe)
  - Windows Resource Monitor (resmon.exe)
  - A more specialized tool (like SysKit Monitor)

Performance Counters
Performance Counters

Performance Counter Basics

The operating system exposes counters

- Memory, CPU, network, and more

```
Memory
- % Committed Bytes In Use
- Available Bytes
- Available KB/Bytes
- Available MB/Bytes
- Cache Bytes
- Cache Bytes Peak
- Cache Faults/sec

Network Interface
- Bytes Received/sec
- Bytes Sent/sec
- Bytes Total/sec
- Current Bandwidth
- Offloaded Connections
- Output Queue Length
- Packets Outbound Discarded

Processor
- % C1 Time
- % C2 Time
- % C3 Time
- % DPC Time
- % Idle Time
- % Interrupt Time
- % Privileged Time
```
Performance Counter Basics

The operating system exposes counters
- Memory, CPU, network, and more

Applications oftentimes expose their own counters
- For instance, SharePoint alone exposes over 20 categories and hundreds of counters
Performance Counter Basics

The operating system exposes counters
  - Memory, CPU, network, and more

Applications oftentimes expose their own counters
  - For instance, SharePoint alone exposes over 20 categories and hundreds of counters

Bottom line: unless you know what to watch, you’ll suffer a cruel and horrible death at the hands of the Performance Counter Gods.
Configuration

We may need to configure our farm to facilitate better data capture (covered in the references):

- Turn off Event Log Flooding Protection
- Reduce the interval on the SharePoint Foundation Usage Data Import Timer Job
- Enable all diagnostic providers
- Lower `job-diagnostics-performance-counter-###-provider` schedule interval (where ### is “wfe” and “sql”)
- Enable stack tracing for content requests
- Enable the Developer Dashboard
- Enable additional usage data collection.
What should I be watching?

That depends on the role of the server:
- Web Front-End
- Application Server
- SQL Server
WFEs serve-up pages through IIS, so we want low values for all of these counters

- ASP.NET: Requests Queued *(should be “low”)*
- ASP.NET: Requests Rejected *(should be 0)*
- ASP.NET: Request Wait Time *(should be near 0)*
- ASP.NET: Worker Process Restarts *(should be 0)*
WFEs serve-up pages through IIS, so we want low values for all of these counters

- ASP.NET: Requests Queued (should be “low”)
- ASP.NET: Requests Rejected (should be 0)
- ASP.NET: Request Wait Time (should be near 0)
- ASP.NET: Worker Process Restarts (should be 0)

WFEs also use their memory for caching to accelerate web requests.

- ASP.NET Applications: Cache API Trims (should be near 0)
- ASP.NET Applications: Cache API Hit Ratio (should be “high”)
- SharePoint Publishing Cache: Total Number of Cache Compactions (should be near 0)
- SharePoint Publishing Cache: Publishing Cache Hit Ratio (should be “high”)
- SharePoint Publishing Cache: Publishing Cache Flushes / Second (should be 0)
WFEs use disks for BLOB caching

- SharePoint Publishing Cache: BLOB Cache % Full (maintain headroom)
Unless an application server is experiencing issues specific to its function (which might require monitoring specialized counters), consider monitoring the following:

- **Processor: % Processor Time** (>75% - 85% is bad)
- **Memory: Available Mbytes** (<2 GB is bad)
- **Memory: Cache Faults/sec** (>1 is bad)
- **Memory: Pages/sec** (>10 is bad)
- **Disk: Avg. Disk Queue Length** (depends)
- **Disk: % Idle Time** (<90% is bad)
- **Disk: % Free Space** (<30% is bad)
Unless an application server is experiencing issues specific to its function (which might require monitoring specialized counters), consider monitoring the following:

- Processor: % Processor Time (>75% - 85% is bad)
- Memory: Available Mbytes (<2 GB is bad)
- Memory: Cache Faults/sec (>1 is bad)
- Memory: Pages/sec (>10 is bad)
- Disk: Avg. Disk Queue Length (depends)
- Disk: % Idle Time (<90% is bad)
- Disk: % Free Space (<30% is bad)

These also are valid for WFEs, as well!
Consider watching the following:

- SQLServer:Buffer Manager: Buffer Cache Hit Ratio
- SQLServer:Databases: Transactions/sec
- SQLServer:General Statistics: User Connections
- SQLServer:Latches: Average Latch Wait Time (ms)
- SQLServer:Latches: Latch Waits/sec
- SQLServer:Locks: Average Wait Time (ms)
- SQLServer:Locks: Lock Wait Time (ms)
- SQLServer:Locks: Number of Deadlocks/sec
- SQLServer:Plan Cache: Cache Hit Ratio
- SQLServer:SQL Statistics: SQL Compilations/sec
- SQLServer:SQL Statistics: SQL Re-Compilations/sec
Performance troubleshooting is an art, not a science.

There is no prescriptive path to cover every situation.

Plan to spend a little time noodling.
Page Performance Monitoring
We’ve been looking at server-side performance monitoring thus far. It represents only half of the overall equation.
We’ve been looking at server-side performance monitoring thus far. It represents only half of the overall equation.

We need go to put ourselves in the role of the end-user to monitor and diagnose a number of other issues, including page performance issues.
We've been looking at server-side performance monitoring thus far. It represents only half of the overall equation.

We need to put ourselves in the role of the end-user to monitor and diagnose a number of other issues, including page performance issues.

What can we do from the other end of the wire?
The answer is “quite a bit”

Your browser is an amazingly capable performance tool – if you understand how to use it.
The answer is “quite a bit”

Your browser is an amazingly capable performance tool – if you understand how to use it.

Requests and their responses are recorded chronologically – including all sorts of information such as HTTP headers, response codes, cookies, and much more.

<table>
<thead>
<tr>
<th>Name/Path</th>
<th>Protocol</th>
<th>Method</th>
<th>Result/Description</th>
<th>Content type</th>
<th>Received</th>
<th>Time</th>
<th>Indicator/Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>/pageглянуть 15.css</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>text/css</td>
<td>1.14 KB</td>
<td>11.59 ms</td>
<td>link</td>
</tr>
<tr>
<td>/pageглянуть 15.css</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>text/css</td>
<td>47.38 KB</td>
<td>5.30 ms</td>
<td>link</td>
</tr>
<tr>
<td>/controls/15.css</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>text/css</td>
<td>2.73 KB</td>
<td>13.19 ms</td>
<td>link</td>
</tr>
<tr>
<td>/SinaNews/15.css</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>text/css</td>
<td>5.44 KB</td>
<td>4.94 ms</td>
<td>link</td>
</tr>
<tr>
<td>/initString.js</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>application/javascript</td>
<td>6.8 KB</td>
<td>3.99 ms</td>
<td>script</td>
</tr>
<tr>
<td>/init.js 15.css</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>application/javascript</td>
<td>74.95 KB</td>
<td>16.42 ms</td>
<td>script</td>
</tr>
<tr>
<td>ScriptResource.15.css</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>application/javascript</td>
<td>23.01 KB</td>
<td>13.2 ms</td>
<td>script</td>
</tr>
<tr>
<td>blank.js</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>application/javascript</td>
<td>381 B</td>
<td>18.9 ms</td>
<td>script</td>
</tr>
<tr>
<td>ScriptResource.15.css</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>application/javascript</td>
<td>9.75 KB</td>
<td>6.12 ms</td>
<td>script</td>
</tr>
<tr>
<td>xskeman.js</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>application/javascript</td>
<td>30.33 KB</td>
<td>18.75 ms</td>
<td>script</td>
</tr>
<tr>
<td>favicon.ico</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>image/png</td>
<td>1.62 KB</td>
<td>10.42 ms</td>
<td>image</td>
</tr>
<tr>
<td>spamming.png</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>image/png</td>
<td>24.35 KB</td>
<td>10.05 ms</td>
<td>image</td>
</tr>
<tr>
<td>staticicon.png</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>image/png</td>
<td>2.29 KB</td>
<td>5.73 ms</td>
<td>image</td>
</tr>
<tr>
<td>WebResource.15.css</td>
<td>HTTP</td>
<td>GET</td>
<td>200 OK</td>
<td>application/javascript</td>
<td>5.87 KB</td>
<td>12.34 ms</td>
<td>script</td>
</tr>
</tbody>
</table>
**Page Performance Monitoring**

**X-SharePointHealthScore**
- A measure of the front-end’s general load or stress. Values from 0 (no stress) to 10 (max stress). We want this low.
Page Performance Monitoring

X-SharePointHealthScore
- A measure of the front-end’s general load or stress. Values from 0 (no stress) to 10 (max stress). We want this low.

SPRequestDuration
- The amount of time your request spends processing on the server (in ms). Ideally less than three seconds (3000ms)
Page Performance Monitoring

X-SharePointHealthScore
- A measure of the front-end’s general load or stress. Values from 0 (no stress) to 10 (max stress). We want this low.

SPRequestDuration
- The amount of time your request spends processing on the server (in ms). Ideally less than three seconds (3000ms)

SPIisLatency
- The amount of time your request spends waiting on the server (in ms). Should be near zero.
Page Performance Monitoring

Round Trip Time – (SPRequestDuration + SPIisLatency) = Time lost “Elsewhere”
Round Trip Time – (SPRequestDuration + SPIisLatency) = Time Lost “Elsewhere”

For example:
- Round Trip Time = 76.04ms
- SPRequestDuration = 51ms
- SPIisLatency = 0
- Time Lost Elsewhere = 25.04ms
Round Trip Time – (SPRequestDuration + SPIisLatency) = Time lost “Elsewhere”

For example:
- Round Trip Time = 76.04ms
- SPRequestDuration = 51ms
- SPIisLatency = 0
- Time Lost Elsewhere = 25.04ms

This is a high-performance SharePoint farm that is not under load.
- May not reflect real world conditions
This will work for ...  
- SharePoint 2013 on-prem
This will work for ...

- SharePoint 2013 on-prem
- SharePoint 2016 on-prem
I’ve got consistently high SPRequestDuration values

- This is oftentimes where we find questionable dev practices
- May be related to server (over-)load or other factors
- X-SharePointHealthScore can corroborate (or not)
I’ve got consistently high SPRequestDuration values
- This is oftentimes where I find questionable dev practices
- May be related to server (over-)load or other factors
- X-SharePointHealthScore can corroborate (or not)

I’m seeing a lot of “time lost elsewhere”
- Network congestion or failure
- Web proxies inserting themselves between you and SharePoint
- DNS resolution issues
- Routing problems
Questions & Answers
References
References

1. Storage and SQL Server Capacity Planning and Configuration (SharePoint Server)
2. Best Practices for SQL Server in a SharePoint Server Farm
3. Diskspd Utility: A Robust Storage Testing Tool (superseding SQLIO)
   https://gallery.technet.microsoft.com/DiskSpd-a-robust-storage-6cd2f223
4. Github repository for diskspd
   https://github.com/microsoft/diskspd
5. Using Microsoft DiskSpd to Test Your Storage Subsystem
   https://sqlperformance.com/2015/08/io-subsystem/diskspd-test-storage
6. CrystalDiskMark 6.0.0
   https://crystalmark.info/download/index-e.html
7. The Ultimate SharePoint Performance Guide
   https://leanpub.com/SharePointPerformanceGuide/c/SysKit
References

8. Monitoring and maintaining SharePoint Server 2013
10. Capacity management and sizing overview for SharePoint Server 2013
11. SharePoint Performance Monitoring – How and Why?
    http://blog.syskit.com/sharepoint-performance-monitoring
12. Performance Counters for ASP.NET
14. ASP.NET Performance Monitoring, and When to Alert Administrators
References

15. MOSS Object Cache Memory Tuning is not an Intuitive Process

16. High Avg Disk Queue Length and Finding the Cause

17. SharePoint Performance: Best Practices from the Field
   https://www.slideshare.net/jasonhimmelstein/sharepoint-performance

18. ULS Viewer

19. Fiddler
   https://www.telerik.com/download/fiddler

20. Using the Developer Dashboard

21. The Five-Minute Page Performance Troubleshooting Guide for SharePoint Online
References

22. Akamai Reveals 2 Seconds As The New Threshold Of Acceptability For Ecommerce Web Page Response Times

23. How Loading Time Affects Your Bottom Line
https://blog.kissmetrics.com/loading-time/
Thank you
Sean P. McDonough

SharePoint and Office 365 Gearhead, Tinkerer, Microsoft MVP

My Company: bitstream
Twitter: @spmcdonough
Blog: http://SharePointInterface.com
About: http://about.me/spmcdonough