Evaluating SMB2 Performance for Home Directory Workloads

Dan Lovinger, David Kruse
Development Leads
Windows Server / File Server Team

Agenda

- Monitoring File Server Performance
- Latency Model
- HomeFolder Workload
- SMB Protocol Options
- Results
- Q&A

Goal

- Consider new information to expose in your server implementation for performance/scalability evaluation
- Estimate impact of a high cost protocol or system component change
- What happens if we invest in directory content leasing?

Analyzing File Server Performance

Performance Counters

Positive

- Provide a summarized view of resource usage across time
  - CPU
  - Disk Queue Length
  - Network Utilization
  - Server Items Queued
- Minimal overhead
  - Computation done inside of the module itself

Negative

- Provides limited insight into performance of an individual task in a set. (Counters are aggregates)
- Static level of detail requires further tools if question can't be answered.
  - Is the IO badly aligned (random) or is the disk simply slow?

Network Captures

Positive

- Can be captured by a 3rd party, so no load on client or server.
- Easily portable file format
- Finer-grained insight based on representation of each individual request or response
  - Operation occurrence, timing, relationship
  - Per-operation latency
- Tools can provide aggregate results

Negative

- Cannot provide insight into local processing.
  - What is the bottleneck in request execution if the latency spikes for some requests?

Windows Performance Analysis Tools

Positive

- Helps identify hot code paths and hot locks – very useful for CPU limited scenarios.
- Integration with storage events provides insight into disk subsystem.

Negative

- Requires significant knowledge of the code.
- Extremely verbose system-wide data can result in difficulty identifying what is relevant.

Answering Deeper Questions

- Current tools answer high level questions
  - Am I storage, network, or CPU bound?
  - What request type was sent most often for this workload?
  - Am I seeing a high level of lock contention?
- Next level of questions is difficult to answer
  - If I installed a faster storage system, how much of a performance gain might my client see?
  - In a complex workload, which component (CPU, file system, etc.) contributes the most to perceived slowness on a per-operation basis?
  - Can I extrapolate how system behavior might be affected if I reduce the number of Create's on the wire?

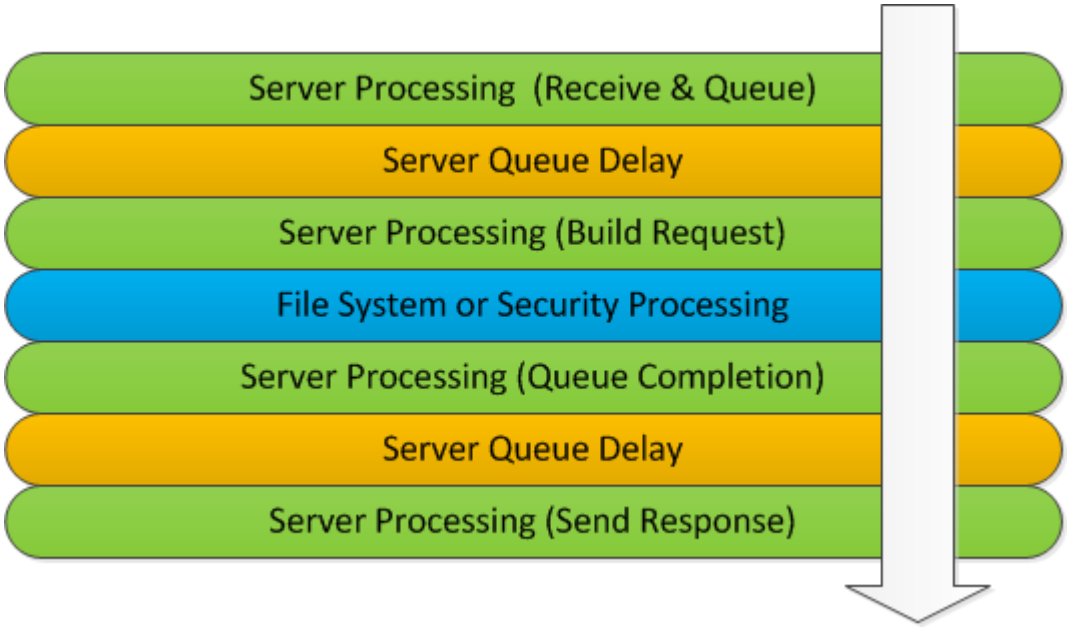
Requirements

- A deep, flexible view of requests processed by the server
  - Allow pivoting on shares, requests, connections, file type, desired access, etc.
- An understanding of what components contributed to completion of the request, and at what cost
- Can be combined with existing performance tools to correlate protocol behavior with system state.

Event-based Model

On receipt, log request information including client, share, request parameters, etc. via an eventing infrastructure	During execution, record the time spent across a slice of components	On completion, log the response information (status, read length, etc.) along with execution information, and correlate with request event.

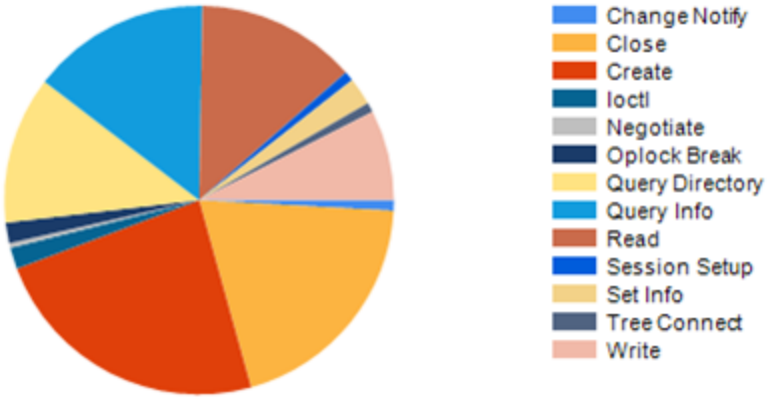
Execution Slicing



Data Mining

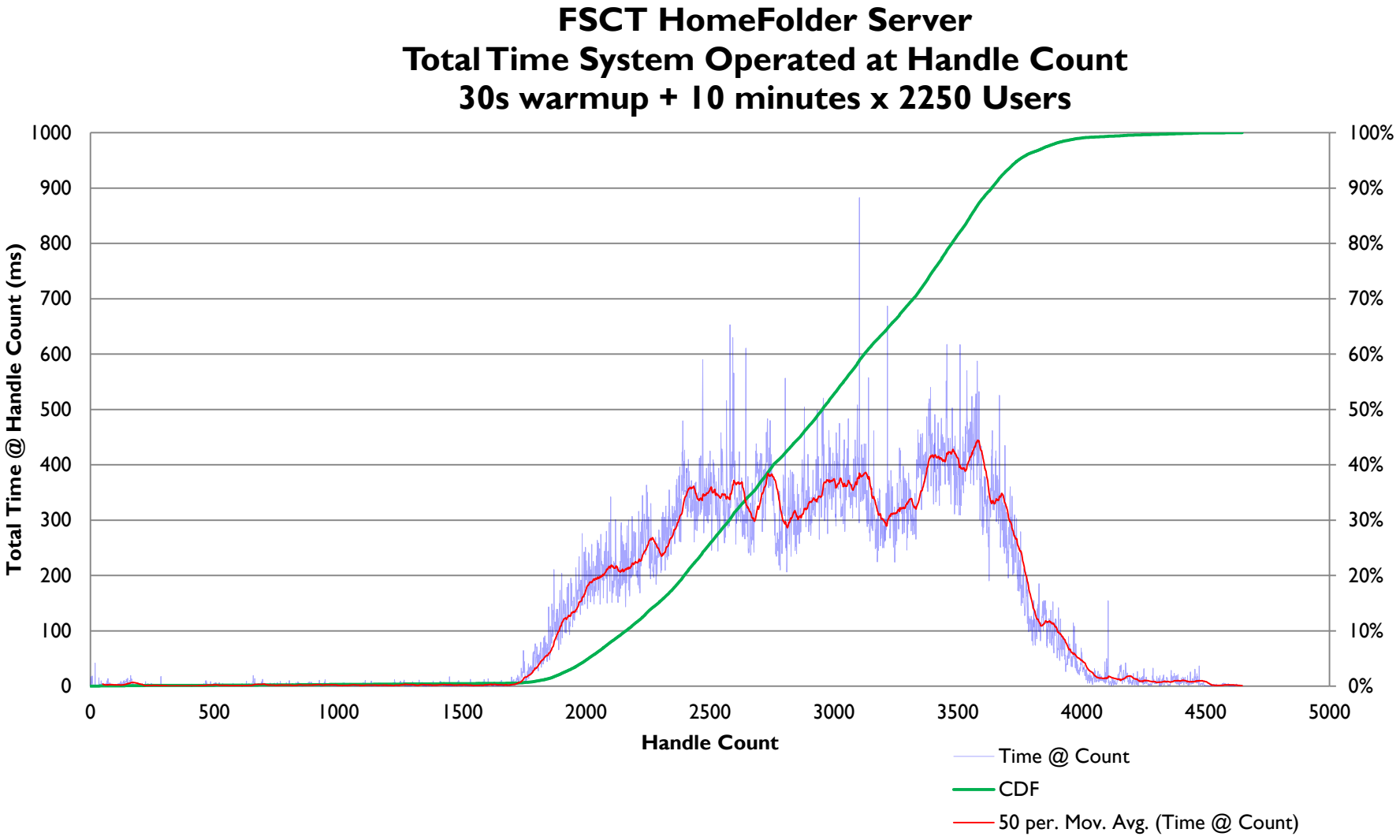
- Detailed analysis is done via post-processing
  - Performance Monitor is a better tool for live viewing as calculations are done in-proc
- Simple queries can execute against a log-file directly
- Complex queries benefit from integration with a database

Command Usage Summary

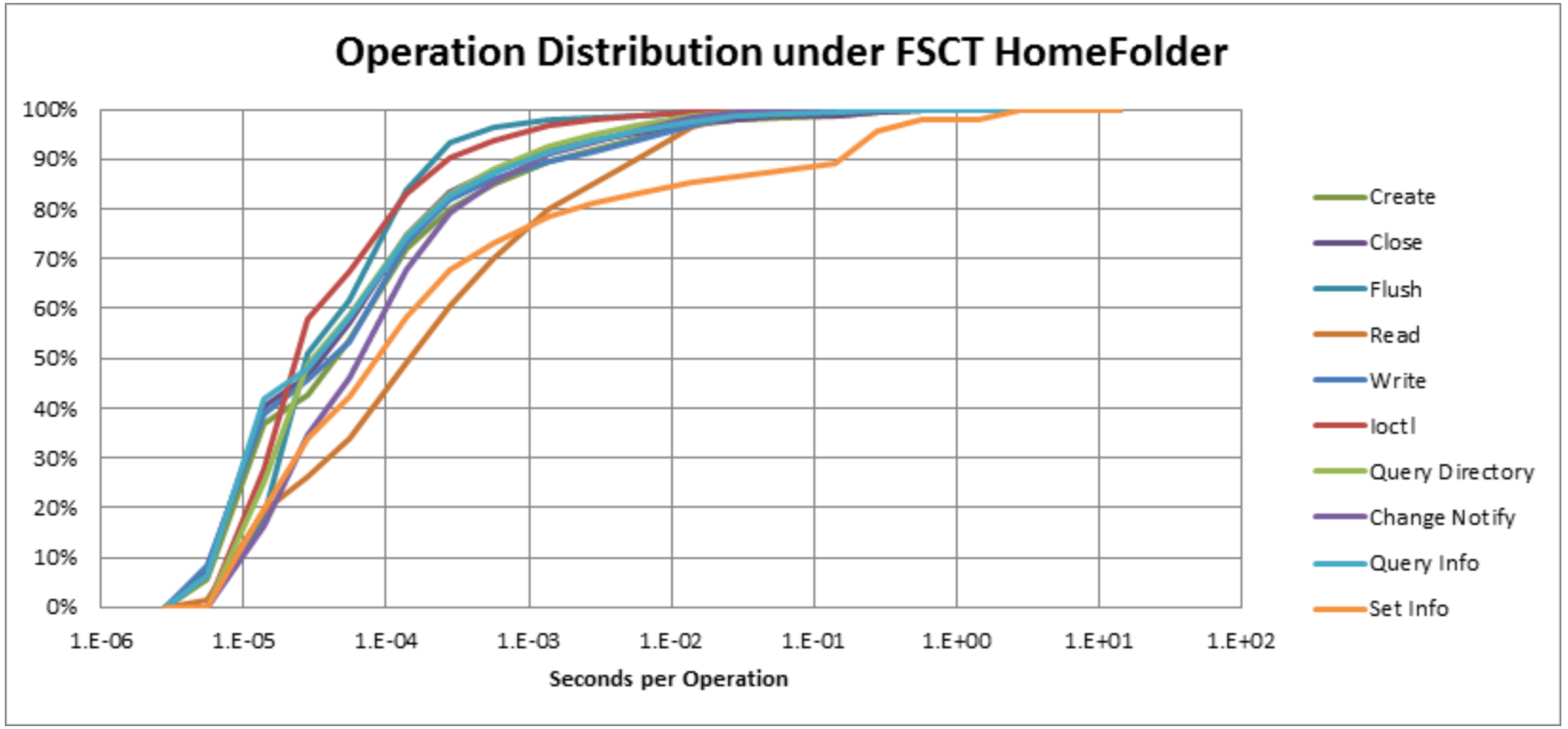


Command Name	Request Count
Change Notify	2256
Close	50579
Create	60184
Ioctl	4517
Negotiate	1025
Oplock Break	4367
Query Directory	31335
Query Info	37938
Read	33990
Session Setup	2050
Set Info	5848
Tree Connect	2049
Write	19167

Local Handle Count across Time



Distribution & Execution Time



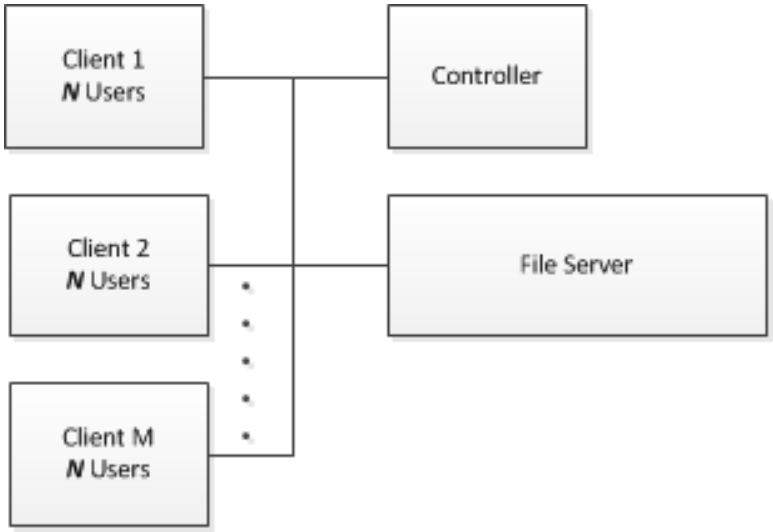
Evaluating the Home Folders Workload

What is FSCT

- File Server Capacity Tool
- First announced at SDC 2008
- Focused on CIFS/SMB/SMB2 File Servers
  - Capacity planning
  - Identifying bottlenecks
- Targeted at
  - IT Professionals
  - Storage Solution Providers
- Results include
  - Maximum number of users for a server configuration
  - Throughput (in scenarios per second) for a server configuration
  - Scenario response time
  - Performance counters for servers and clients

FSCT History

- Origins
  - Internal tool developed by the Windows Fundamentals Performance Team
  - Useful for capacity planning and identifying bottlenecks
  - Used internally every day in Windows Performance Labs
  - Focused on server quality gates and regression testing
- Highlights
  - Each client test machine simulates multiple users
  - Independent TCP connections and FS sessions
  - Exercises sequences of file operations to simulate applications - scenarios
- Version 1.0 released in September 2009
- Available in the Plugfest lab



FSCT HomeFolder Workload

- Simulates Windows user home directories
  - Core file server workload
- Non goals to date
  - DB OLTP, VM image hosting, CAD/CAM, Compile, …
- Properties
  - No data sharing between clients
  - Metadata intensive
  - Mix of navigation and file up/download
- Derived from live traces of Microsoft IT File Servers circa 2006
  - Office 2003
  - Windows XP clients
  - Windows 2003 Servers
  - Mapped to Office 2007/Vista
- Still close to current systems: Windows 7/2008 R2 & Office 2010
  - Re-verified by comparing wire app traffic
- Scenarios repeat the API level actions of the app
  - Twelve scenarios based on CMD, Explorer, Word
  - vs. file sets as seen on the MSIT servers

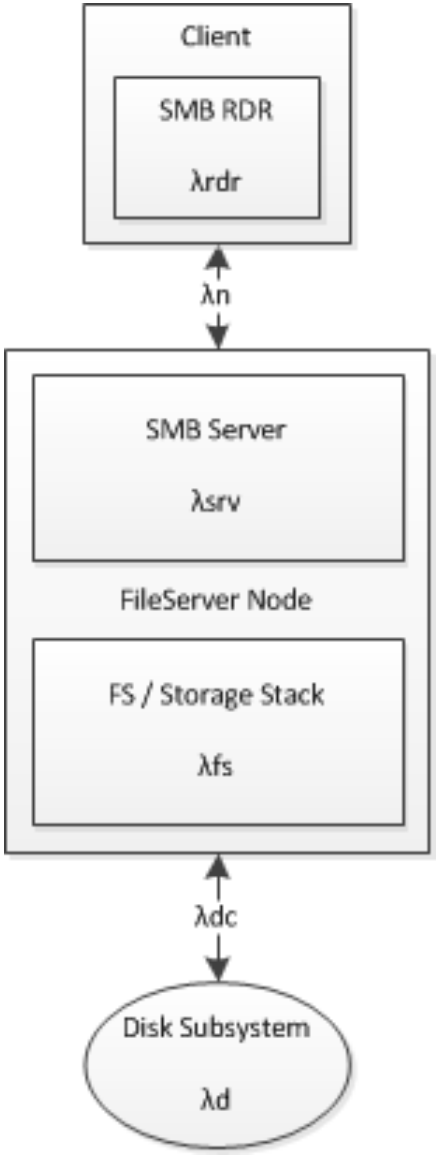
HomeFolder Scenarios

- Each user follows a frequency distribution of scenarios
- ~1 scenario every 11 seconds
- Capacity Metric
  - How many users can the server satisfy without *overload*, where users cannot maintain scheduling rate

FSCT Scenario	FSCT Runs Per Hour	FSCT Run%
CmdLineFileDelete	7	2.1%
CmdLineFileDownload	150	45.3%
CmdLineFileUpload	40	12.1%
CmdLineNavigate	15	4.5%
ExplorerDragDropFileDownload	50	15.1%
ExplorerDragDropFileUpload	15	4.5%
ExplorerFileDelete	5	1.5%
ExplorerNavigate	15	4.5%
ExplorerSelect	15	4.5%
WordEditAndSave	5	1.5%
WordFileClose	7	2.1%
WordFileOpen	7	2.1%
	331	100.0%

Initial Physical Model

- Can we just count operations?
- Wireframe paths for a given operation
- Each component adds some latency, λ, adding to total latency
  - $\lambda = \lambda rdr + 2\lambda n + \lambda srv + \lambda fs + 2\lambda dc + \lambda d$
- Note carefully
  - Latency input will be per workload & scale
  - Need to navigate tradeoff of looking at all operation subtypes
- Correlating the disk subsystem activity is difficult … and rolls into the FS stack
- Current goal – model changes in protocol to remove operations
- Estimate that the client processing would occur regardless
- Simplified model
  - $\lambda = 2\lambda n + \lambda srv + \lambda fs$
- Build vectors Λ subdivided by command type
  - $\Lambda = (2\Lambda n + \Lambda srv + \Lambda fs) \times \mathrm{Ops}$



Model With Compounding

- Compound ops incur a single network trip
  - Create + QueryDir + QueryDir
  - Query FS Volume Info + FS Attributes
- Operations don't compound in the server, so …
- Method
  - Count packets – network latency
  - Count operations – component latency
- One way to account for packets while maintaining structure
  - Packet: by the first/lead operation within them
  - Compound: the second, third, etc. compounded commands
- Final latency equation
  - $\Lambda = 2\Lambda n \times Packet + (\Lambda srv + \Lambda fs) \times (Packet + Compound)$

Measuring HomeFolder

- SMB 2.1 total wire ops (+ non-lead compound)
- Defer breakdown of op subtypes – refinement
- IO averaged over the HomeFolder file distribution

	Cancel	Change Notify	Close	Create	Ioctl	Lock	Query Dir	Query Info	Set Info	Read Avg	Write Avg
CmdLineFileDelete	-	-	3	4	-	-	2 (1)	2 (1)	1	-	-
CmdLineFileDownload	-	-	4	4	-	-	2 (1)	3 (1)	-	5.57	-
CmdLineFileUpload	-	-	3	4	-	-	-	2 (1)	3	-	17.15
CmdLineNavigate	-	-	3	4	-	-	4 (3)	3 (1)	-	-	-
ExplorerDragDropFileDownload	-	-	6	9	-	-	2 (1)	8 (2)	-	6.73	-
ExplorerDragDropFileUpload	-	-	7	9	-	-	4 (2)	4 (2)	2	-	17.15
ExplorerFileDelete	2	4 (2)	16	17	-	-	10 (5)	3 (2)	1	10.75	-
ExplorerNavigate	2	2 (2)	8	11	-	-	4 (2)	3 (2)	-	-	-
ExplorerSelect	-	-	3	3	-	-	2 (1)	6 (1)	-	2.58	-
WordEditAndSave	-	-	13	16	-	-	-	15 (1)	10	2.00	8.95
WordFileClose	1	4	21	23	2 (2)	-	8 (4)	4 (2)	1	-	2.00
WordFileOpen	-	-	28	32	12 (7)	-	14 (8)	13 (2)	-	6.57	-

SMB Investigation: Protocol Options

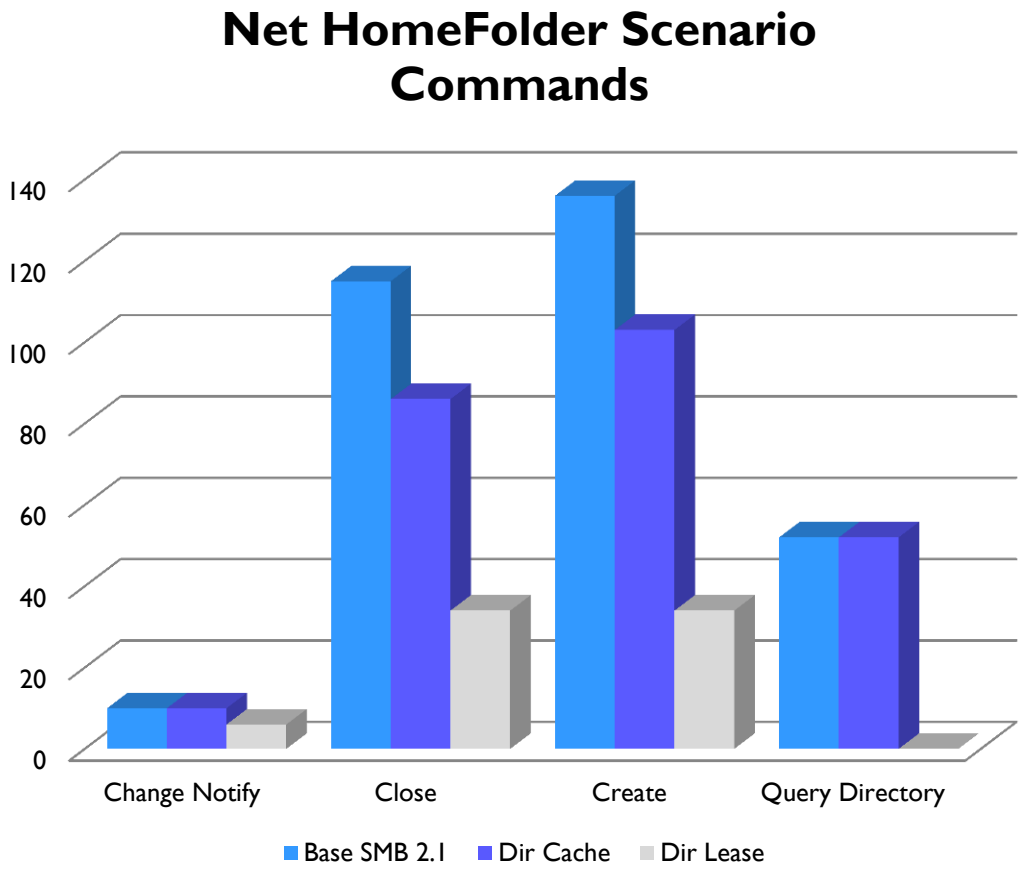
- Modeling of Directory Cache Improvements v. Directory Leases
- Do it better: full collapse of opens accessing cache
- Lease: take a 10s non-coherent cache and make it coherent/durable
- HomeFolder scenario traces analyzed by developer

Directory Cache Imp.	Close	Create
CmdLineFileDelete		
CmdLineFileDownload		
CmdLineFileUpload		
CmdLineNavigate		
ExplorerDragDropFileDownload	-1	-2
ExplorerDragDropFileUpload	-3	-4
ExplorerFileDelete	-5	-5
ExplorerNavigate	-3	-4
ExplorerSelect		
WordEditAndSave		
WordFileClose	-9	-9
WordFileOpen	-8	-9

Directory Leases	Change Notify	Close	Create	Query Dir
CmdLineFileDelete		-2	-3	-2
CmdLineFileDownload		-2	-2	-2
CmdLineFileUpload		-1	-2	
CmdLineNavigate		-3	-4	-4
ExplorerDragDropFileDownload		-4	-7	-2
ExplorerDragDropFileUpload		-6	-8	-4
ExplorerFileDelete	-2	-13	-14	-10
ExplorerNavigate		-8	-11	-4
ExplorerSelect		-1	-1	-2
WordEditAndSave		-3	-6	
WordFileClose	-2	-17	-19	-8
WordFileOpen		-21	-25	-14

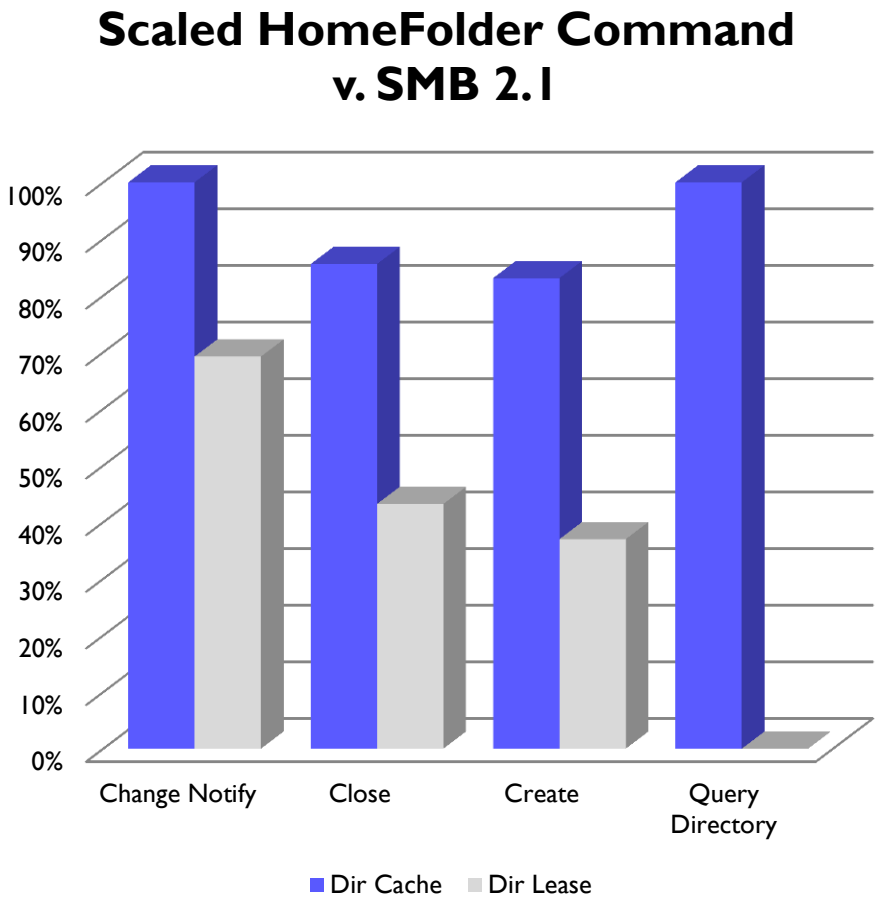
SMB Investigation: Protocol Options

- Per scenario view
- Run each scenario, once
- Impact intuitively expensive operations
- Directory Leases go *much* further



SMB Protocol Options - Scaled

- Scale onto the HomeFolder scenario distribution
  - 150 CmdLineFileDL
  - 7 WordFileOpen
  - …
- This is now an actual command pattern in an HomeFolder run



Latency Estimates

- Use the operation and latency model to extrapolate to real hardware
- FSCT used to load low-end reference server
- SMB Server event counters result in operation latency estimates
- Reference System
  - overload @ 2300
  - measured @ 2250

Reference Server Configuration
2GHz 4 cores
4GB RAM
1Gbe Ethernet
4Gb Fibre Channel
RAID 5 SAS 10K RPM Drives

Latency Estimates

		Server Queue Time (s)			Server Processing Time (s)			Filesystem Time (s)		
	Count	Mean	StdDev	95th %ile	Mean	StdDev	95th %ile	Mean	StdDev	95th %ile
Create	904638	0.008470	0.099420	0.007667	0.000062	0.000131	0.000110	0.003162	0.021645	0.018506
Close	897718	0.007944	0.099567	0.004853	-	-	-	-	-	-
Flush	6725	0.001209	0.037210	0.000391	0.000028	0.000041	0.000044	0.027126	0.049994	0.108172
Read	514862	0.001939	0.006059	0.010790	0.000016	0.000065	0.000023	0.006726	0.011108	0.022162
Write	341346	0.002069	0.025154	0.008328	0.000016	0.000060	0.000025	0.001429	0.010783	0.004775
Ioctl	38541	0.000319	0.002617	0.163206	0.000017	0.000031	0.000044	0.001252	0.009843	0.007777
Query Directory	385783	0.000770	0.004588	0.005842	0.000016	0.000043	0.000043	0.000177	0.005303	0.000371
Change Notify	25132	0.000950	0.004572	0.007616	0.000029	0.000135	0.000085	0.042266	0.111234	0.360713
Query Info	1532866	0.004754	0.075035	0.008879	0.000015	0.000052	0.000042	0.000025	0.001494	0.000089
Set Info	102854	0.066584	0.295014	0.518083	0.000020	0.000123	0.000050	0.000443	0.011081	0.000517

- Input to the latency model
- Queue time has become significant, on the order of filesystem time
- High variance
  - read/write – expected (op v. bandwidth)
  - create – large tails
  - set info – an operation to break down
- Specific to this system, of course.

Note: CLOSE processing time not measured due to a limitation in SMB SRV

Model Results – Scaled Scenarios

- Summary results for a single scaled execution
  - Extrapolate from LAN to WAN impact
  - 2 - 20ms client-server network latency

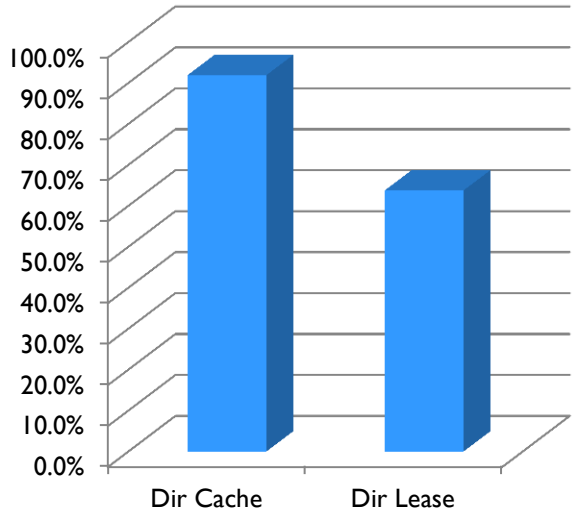
Scaled Single Scenario Execution (seconds) – 2ms			
	SMB 2.1	DirCache	DirLease
CmdLineFileDelete	1.353	1.353	0.796
CmdLineFileDownload	32.647	32.647	23.142
CmdLineFileUpload	18.819	18.819	17.035
CmdLineNavigate	2.001	2.001	0.343
ExplorerDragDropFileDownload	18.971	16.741	10.534
ExplorerDragDropFileUpload	8.34	6.818	5.101
ExplorerFileDelete	4.641	3.93	2.114
ExplorerNavigate	6.049	4.527	1.835
ExplorerSelect	2.641	2.641	2.117
WordEditAndSave	6.812	6.812	6.143
WordFileClose	6.784	4.993	2.383
WordFileOpen	7.989	6.284	3.066
Total	117.045	107.565	74.609

20ms		
SMB 2.1	DirCache	DirLease
3.873	3.873	1.804
122.114	122.114	85.609
59.355	59.355	53.251
7.401	7.401	1.423
70.689	63.059	40.652
29.481	24.179	17.602
14.496	11.985	6.209
19.009	13.707	4.535
10.512	10.512	8.368
18.323	18.323	16.034
21.400	15.073	6.919
30.308	24.319	12.281
406.959	373.899	254.687

Model Results - Summary

Relative FSCT Latency			
Network Latency	SMB 2.1	DirCache	DirLease
200us	100.0%	91.9%	64.3%
2ms	100.0%	91.9%	63.7%
20ms	100.0%	91.9%	62.6%
200ms	100.0%	91.9%	62.2%

- Converges across varying network latency
  - Not necessarily intuitive. Build model, check model … ultimately, trust model.
- Collapse cache opens
  - 8% to SMB 2.1
- Directory Leases
  - 36-38%
- Note …
  - Effect modeled on a single client
  - Processing cost of implementation?
  - Practiced estimation



Summary

- Consider performance counters and events in your design process
- New protocols - SMB2 & NFS 4 - are opportunities to revisit your performance infrastructure
- More data = More insight = Better designs, earlier
- Extrapolation models can be an interesting input to the design process

Questions?

Resources

- Windows Performance Analysis Tools (WPT/XPerf)
  - http://msdn.microsoft.com/en-us/performance/cc825801.aspx
- File Server Capacity Tool 1.0
  - http://www.microsoft.com/downloads/details.aspx?FamilyID=b20db7f1-15fd-40ae-9f3a-514968c65643
- Microsoft Network Monitor 3.4
  - http://www.microsoft.com/downloads/details.aspx?FamilyID=983b941d-06cb-4658-b7f6-3088333d062f
  - See slides from talk earlier today (Tuesday 1PM): Interoperability Tools for CIFS/SMB/SMB2