

Database Health and Performance

(AKA I can **prove** it's not my system's fault!)

SQL Saturday Lincoln – October 6, 2012

David Klee – Solutions Architect (@kleegeek)



© 2012 House of Brick Technologies, LLC



About House of Brick

- 14 year old Omaha-based company
- Leader: Tier-1 VMware, Database Performance
- Rock-solid reputation for optimizing the entire system stack to maximize Tier-1 performance
- House of Brick key service value components
 - Hybrid/private cloud architectures for complex Tier-1 workloads
 - Legacy to virtualization, and private/hybrid cloud system replatforming
 - SQL Server and Oracle virtualization specialties
 - Short term assessments and proof-of-concept projects
 - Long-term project analysis, PM, implementation, & validation



© 2012 House of Brick Technologies, LLC

About Me



vmware
CERTIFIED
ADVANCED
PROFESSIONAL 5
DATACENTER
DESIGN

David Klee
Twitter: @kleegeek
Blog: davidklee.net

Microsoft
CERTIFIED
IT Professional

Database Administrator 2008
Database Administrator on SQL Server® 2005
Database Developer 2008

- SQL Server on VMware team lead
- Experience in VMware, Microsoft, Linux, networking, security, application development technologies

© 2012 House of Brick Technologies, LLC

Average Day

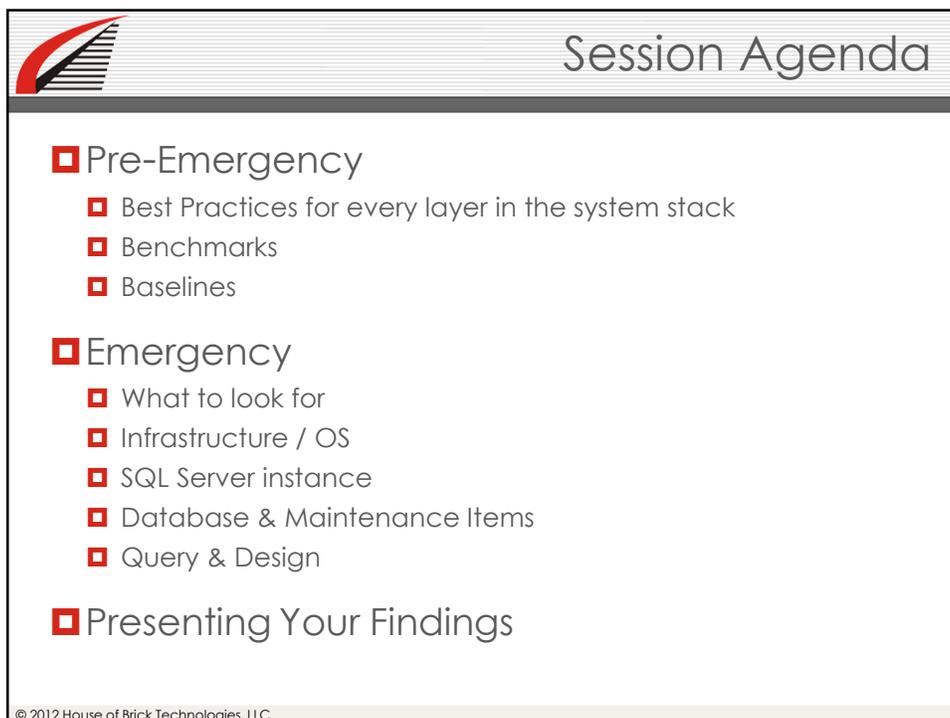
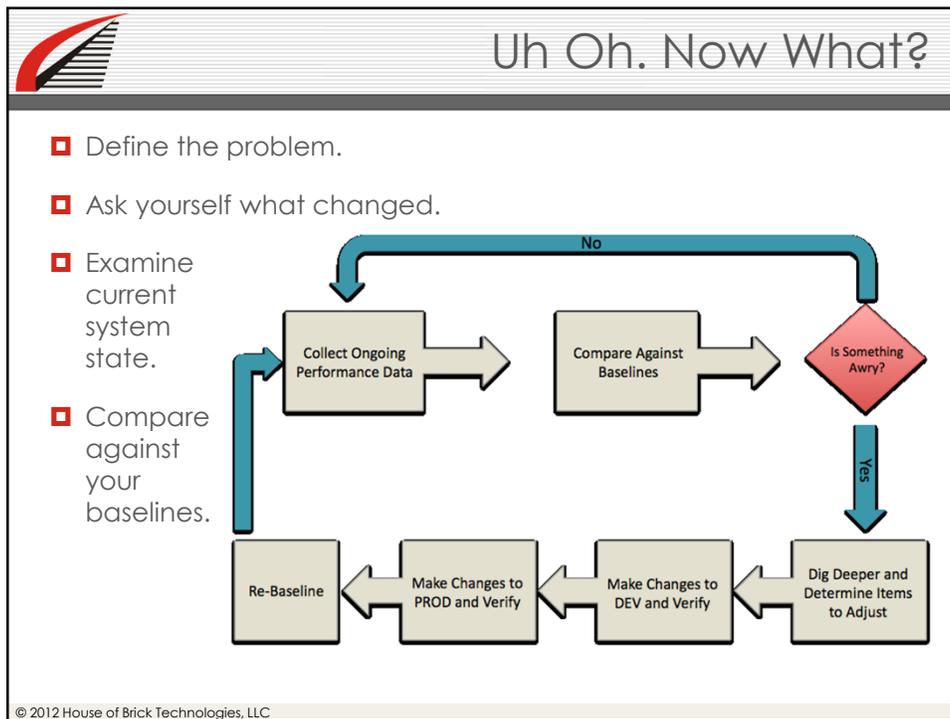
8:15 AM: "My app is running slow! What did you do to it?!"

(App owner and what he wants to do to your systems) →



How can you prove that your systems are running optimally?

© 2012 House of Brick Technologies, LLC





Some Best Practices

- ▣ Hardware
 - ▣ Storage
 - ▣ Interconnects
- ▣ Virtualization
- ▣ Operating System
- ▣ SQL Server
 - ▣ Instance
 - ▣ Database
 - ▣ Maintenance



© 2012 House of Brick Technologies, LLC



Best Practices - Infrastructure

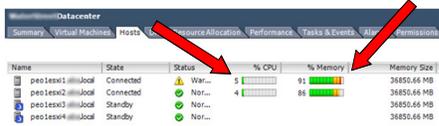
- ▣ Hardware
 - ▣ Set power management to Maximum Performance
 - ▣ Up-to-date BIOS, hardware drivers
- ▣ Storage
 - ▣ Iometer – At least 60MB/s in all tests
 - ▣ SQLIO – latency no greater than 25ms
- ▣ Interconnects
 - ▣ Fastest storage fiber and Ethernet you can get
 - ▣ Multiple paths a requirement
 - ▣ Iperf to determine if you have problems



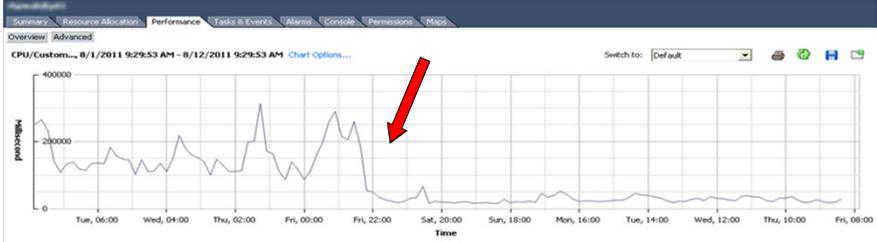
© 2012 House of Brick Technologies, LLC

Best Practices - Virtualization

- ▣ CPU Utilization
 - ▣ CPU Ready time
 - ▣ 300ms average
 - ▣ 500ms high water mark
- ▣ VM Resources
 - ▣ Right-sized
 - ▣ Full memory reservation
- ▣ Memory Utilization
 - ▣ No host overcommitment!
 - ▣ No ballooning!



Name	State	Status	% CPU	% Memory	Memory Size
peo1esw1@localhost	Connected	War...	5	91	36850.64 MB
peo1esw2@localhost	Connected	Nor...	4	86	36850.64 MB
peo1esw3@localhost	Standby	Nor...			36850.64 MB
peo1esw4@localhost	Standby	Nor...			36850.64 MB



Performance Chart Legend

Key	Object	Measurement	Rollup	Units	Latest	Maximum	Minimum	Average
■	Hosts	CPU Ready	Summation	Millisecond	27896	314324	14836	80934.1

© 2012 House of Brick Technologies, LLC

Best Practices - Virtualization

- ▣ Storage
 - ▣ Check stats for path and datastore overload
 - ▣ Latency < 30ms
- ▣ PVSCSI driver for all non-OS drives
- ▣ Partition alignment (not just Windows Server 2003)
- ▣ Multipathing driver



SQL Server
VM



OS VMDK



Data VMDK



Log VMDK



TempDB VMDK

LSI Logic

PVSCSI

© 2012 House of Brick Technologies, LLC



Best Practices - OS

- Disks
 - NTFS 64KB allocation size
 - Check partition alignment (Windows 2003 especially)
 - Disable generation of 8.3 names (fsutil behavior set disable8dot3 1)
 - Disable last file access time tracking (fsutil behavior disableaccess 1)
- System
 - Antivirus exclusions for MDF, NDF, LDF, BAK set (<http://support.microsoft.com/kb/309422>)
- SQL Server Services
 - Enable Lock Pages in Memory (Enterprise ed. only)
 - Instant File Initialization
- Perfmon
 - Set to always collect perf counters every 5m
 - Rotate log files nightly



Windows Server® 2008 R2



© 2012 House of Brick Technologies, LLC



Best Practices – SQL Server

- Set min and max instance memory settings
- Optimize for Ad-hoc Workloads
- Tempdb data files
 - More than one (# cores?)
 - Grow at same rate
- Watch for high log file VLF counts
- Agent system alerts – severities 17-25, 823-825
 - Default Operator set to distribution group



Microsoft®
SQL Server® 2008 R2



Microsoft®
SQL Server® 2012

© 2012 House of Brick Technologies, LLC

Best Practices – SQL Server

- ▣ Proper Maintenance is a Must!
- ▣ Fantastic database maintenance solution – ola.hallgren.com
 - ▣ Backups
 - ▣ Indexes / Statistics
 - ▣ Integrity Checks
 - ▣ Work file cleanup
- ▣ Configure email notifications and set default operator
- ▣ Demo (1)

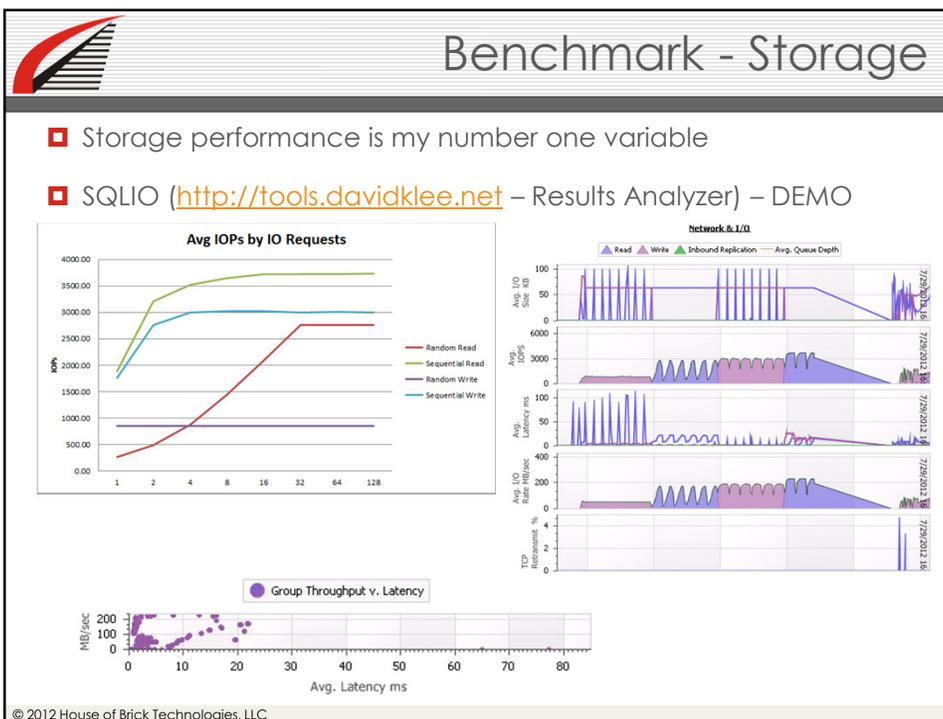
© 2012 House of Brick Technologies, LLC

Getting Started - Benchmarking

- ▣ Must know how to benchmark so you can establish baselines
- ▣ Repeatable process to get point in time performance metrics
- ▣ Benchmarks affect the speed of the system during the test!
- ▣ What changes between tests / iterations?
- ▣ What to benchmark?
 - ▣ Subsystem speed
 - ▣ Objective SQL Server instance speed
 - ▣ Known process / job performance and runtimes
 - ▣ Query runtimes / impact
 - ▣ *Application performance*

vCPU Scheduling Performance Impact

© 2012 House of Brick Technologies, LLC



Benchmark – SQL Server

- ▣ DVDStore
- ▣ <http://linux.dell.com/dvdstore>

Threads	Core x GHz	VM	VM	Itanium Physical	Itanium Physical	8CPU VM to Avg. Itanium Improvement
		8x2GHz	32x2GHz	8x1.6GHz Run 1	8x1.6GHz Run 2	
	MaxDOP	HP Orders	HP Orders	Itanium Orders	Itanium Orders	
2	1	19277	13589	15612	12853	35.44%
2	2	19251	17858	16368	17553	13.50%
2	3	18841	17453	17214	18209	6.38%
2	4	15839	15640	15147	18306	-5.31%
2	5	15953	15779	10201	16866	17.88%
2	6	16263	16055		17596	-7.58%
8	1	76590	63910	62896	60789	23.85%
8	2	76592	70705	69392	71523	8.71%
8	3	75441	69335	65737	69184	11.83%
8	4	57508	61412	44123	71918	-0.88%
8	5	55021	61579	42442 (extrapolated)	67073	0.48%

© 2012 House of Brick Technologies, LLC



Benchmark – DB Performance

- ▣ Every environment is different!
- ▣ Reports / Jobs / Etc.
 - ▣ Average Runtimes
- ▣ Query Performance
 - ▣ CPU Impact
 - ▣ Memory Impact
 - ▣ Storage Impact
 - ▣ TempDB Impact
- ▣ Application owners should be involved in benchmarking process
- ▣ Tools
 - ▣ Perfmon
 - ▣ Extended Events
 - ▣ Simple things:
 - ▣ Set statistics io on/off
 - ▣ Set statistics time on/off
 - ▣ Demo (2)

© 2012 House of Brick Technologies, LLC



Baselines

- ▣ Creating a baseline != Tuning Time
- ▣ Baseline = Averages and peaks during routine activities
- ▣ Can help predict growth and resource contention
- ▣ Establish performance thresholds and high / low water marks
- ▣ Update your baselines after major system changes and/or fixed period of time
- ▣ Be consistent in your approach.
- ▣ Helpful to create a baseline repository
 - ▣ <http://www.sql-server-performance.com/2010/baseline-repository/>

© 2012 House of Brick Technologies, LLC



Running Baselines - Perfmon

- ▣ Create your own constant, running system baseline
- ▣ Record every 5m, cycle log files nightly

Counters:

- Memory – Pages/sec
- Network Interface – Bytes total/sec
- Physical Disk – Disk Transfers/sec
- Processor - % Processor Time
- SQLServer:Access Methods - Full Scans/sec
- SQLServer:Buffer Manager – Buffer Cache Hit Ratio
- SQLServer:Databases Application Database - Transactions/sec
- SQLServer:General Statistics - User Connections
- SQLServer:Latches – Average Latch Wait Time
- SQLServer:Locks - Average Wait Time
- SQLServer:Locks - Lock Timeouts/sec
- SQLServer:Locks - Number of Deadlocks/sec
- SQLServer:Memory Manager - Memory Grants Pending

© 2012 House of Brick Technologies, LLC



Getting Started with Benchmarks & Baselines

- ▣ **Easiest Places to Start:**
- ▣ Glenn Berry – SQL Server Diagnostics Queries
 - ▣ <http://sqlserverperformance.wordpress.com>
- ▣ Brent Ozar – SQL Blitz
 - ▣ <http://www.brentozar.com/sql/blitz-minute-sql-server-takeovers/>
- ▣ You know your systems. Baseline your system performance frequently.
- ▣ What objective application metrics can you repetitively measure?



© 2012 House of Brick Technologies, LLC



Back to the Problem At Hand...

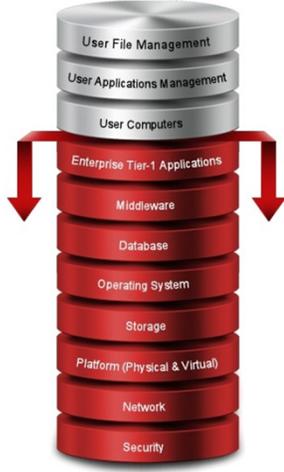


© 2012 House of Brick Technologies, LLC



Performance Triage

- ▣ Performance triage starts with comparing real-time stats against your baselines.
- ▣ Work Down the Stack
 - or*
 - ▣ Work Up the Stack
 - ▣ Hardware, Storage, and Virtualization
 - ▣ Operating System
 - ▣ SQL Server instance
 - ▣ Database(s)
 - ▣ Queries



© 2012 House of Brick Technologies, LLC



System Stack

- ▣ Raw Server Health
 - ▣ Hardware alarms / warnings / failures
 - ▣ Interconnect failures / dead paths
 - ▣ Networking overloaded or degraded network path
 - ▣ Sub-par raw storage performance
 - ▣ IOmeter & SQLIO
 - ▣ SQLIO Analyzer @ tools.davidklee.net
- ▣ Virtualization Health
 - ▣ Host CPU and memory states
 - ▣ High CPU Ready and/or memory ballooning
 - ▣ Host resource overcommitment
- ▣ Operating System
 - ▣ Check Windows Event Log



© 2012 House of Brick Technologies, LLC



CPU Bottlenecks

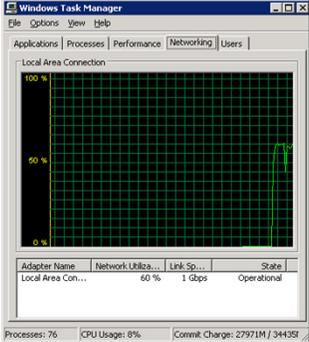
- ▣ Raw CPU usage by instance is very high
- ▣ Signal wait statistic is greater than 25% of total waits
- ▣ Plan re-use is less than 90%
- ▣ Parallel wait statistic CXPACKET is greater than 10% of total waits
- ▣ High work queue count for long duration
- ▣ Demo (3)

Query Source: <http://blog.sqlauthority.com/2009/08/17/sql-server-measure-cpu-pressure-cpu-business/>

© 2012 House of Brick Technologies, LLC

Network Problems

- ▣ You have high network latency
- ▣ You have dropped packets
- ▣ You have maxed out your network bandwidth
- ▣ Wait stats to check
 - ▣ ASYNC_IO_COMPLETION
 - ▣ ASYNC_NETWORK_IO
- ▣ Task manager / perfmon on server
- ▣ Use iperf to check network paths



The screenshot shows the Windows Task Manager Performance tab. The 'Networking' section is selected, showing a graph for 'Local Area Connection' with network usage at 60%. Below the graph is a table with the following data:

Adapter Name	Network Utiliza...	Link Sp...	State
Local Area Con...	60 %	1 Gbps	Operational

At the bottom of the window, system statistics are shown: Processes: 76, CPU Usage: 8%, Commit Charge: 27971M / 34435M.

© 2012 House of Brick Technologies, LLC

Memory Pressure

- ▣ Page life expectancy is low.
 - ▣ 300 is NOT the magic number!
 - ▣ One Suggestion: [Data Cache Size (GB) / 4GB] x 300
 - ▣ See <http://www.salskills.com/blogs/jonathan/post/Finding-what-queries-in-the-plan-cache-use-a-specific-index.aspx> for more details
- ▣ Buffer Cache Hit ratio < 90%
- ▣ High Checkpoint pages / sec & Lazywrites / sec
- ▣ Ring Buffers
 - ▣ http://salskills.com/blogs/jonathan/post/identifying-External-Memory-Pressure-with-dm_os_ring_buffers-and-RING_BUFFER_RESOURCE_MONITOR.aspx
- ▣ Demo (4)
- ▣ Read More:
 - ▣ <http://www.mssqltips.com/sqlservertip/2304/how-to-identify-microsoft-sql-server-memory-bottlenecks/>
 - ▣ <http://blog.sqlauthority.com/2010/12/10/sql-server-finding-memory-pressure-external-and-internal/>

© 2012 House of Brick Technologies, LLC



I/O Performance

- ❑ Exceptionally high average disk seconds per read
- ❑ High volume of disk stalls and/or high disk stall wait times
- ❑ High wait stats around:
 - ❑ ASYNCH_IO_COMPLETION
 - ❑ IO_COMPLETION
 - ❑ LOGMGR
 - ❑ PAGEIOLATCH_*
 - ❑ WRITELOG
- ❑ Disk read / write stalls high
- ❑ Your triage benchmark falls well below your established baseline
- ❑ Get more reports from your storage administrator and investigate further



© 2012 House of Brick Technologies, LLC



Indexes and Statistics

- ❑ No index? Table scans = slow performance.
- ❑ No stats? The optimizer has to guess!
- ❑ Check for:
 - ❑ Index fragmentation is high
 - ❑ Statistics are out of date
 - ❑ High number of 'bad' indexes
 - ❑ Execution plan missing index warnings / bad stats
 - ❑ Is estimated number of rows WAY off from actual?
 - ❑ (Bad) DTA query index recommendations that **someone** applied
 - ❑ Is Auto Create Statistics set to off?
- ❑ Demo (5)
- ❑ Read More:
 - ❑ <http://www.simple-talk.com/sql/database-administration/brads-sure-guide-to-indexes/>
 - ❑ <http://www.simple-talk.com/sql/performance/sql-server-statistics-problems-and-solutions/>

© 2012 House of Brick Technologies, LLC



Bad Queries

- ▣ Tools: DMVs and Profiler
- ▣ Obvious places to start:
 - ▣ High CPU utilization
 - ▣ High memory consumption
 - ▣ High I/O requirements
- ▣ Missing / bad indexes and/or statistics
- ▣ Fetching more columns / rows than needed (*select * from dbo.x*)
- ▣ Bad execution plan

 **Include Actual Execution Plan** Ctrl+M

 - ▣

```
select plan_handle, creation_time, last_execution_time, execution_count, qt.text
FROM sys.dm_exec_query_stats qs
CROSS APPLY sys.dm_exec_sql_text (qs.[sql_handle]) AS qt
```
 - ▣ DBCC FREESYSTEMCACHE(plan handle) to clear it

© 2012 House of Brick Technologies, LLC



Locking & Blocking

- ▣ A top wait statistic is LCK_M_XX AND the average time is high
 - ▣ Sys.dm_os_wait_stats
- ▣ High number of deadlocks
- ▣ Block process report reports long blocks
- ▣ High average row lock or latch waits
- ▣ Long running transactions
- ▣ Index contention
 - ▣ Sys.dm_db_index_operational_stats
- ▣ Read More:
 - ▣ <http://www.sqlskills.com/BLOGS/PAUL/post/Wait-statistics-or-please-tell-me-where-it-hurts.aspx>

© 2012 House of Brick Technologies, LLC



Database Origins

- ▣ Is the database that is performing poorly a third-party database, or one where you have control over the design?
- ▣ If third-party, will the vendor support you changing things?
 - ▣ Doubtful.
 - ▣ Will application support be dropped if you change anything?
- ▣ If custom, can you change things?
 - ▣ Maybe. It depends.
- ▣ If third-party, is your organization willing to 'void the warranty'?



© 2012 House of Brick Technologies, LLC



Bad Database Design

- ▣ Too many table joins for frequent queries
- ▣ Table too wide (many null columns without sparse columns)
- ▣ Not enough appropriate indexes
- ▣ Too many inappropriate ones
- ▣ Indexes too huge – check for GUIDs in primary key
- ▣ No / bad normalization
- ▣ No foreign keys / primary keys / relationships
- ▣ Inappropriate isolation level

© 2012 House of Brick Technologies, LLC



Put Up or (Don't) Shut Up

- ▣ Document your findings thoroughly and accurately
- ▣ Don't just point another finger if it is not your system's fault
 - ▣ Provide evidence of the problem
 - ▣ Identify the system stack component at fault
 - ▣ Assist with remediation as needed
 - ▣ Provide validation when resolved
- ▣ If the problem is in your system, acknowledge it and continue
 - ▣ Remediate at first possible opportunity
 - ▣ Develop long term plan for solution, not just a patch
 - ▣ Append monitoring to try to proactively alert if it happens again
- ▣ Remember
 - ▣ This is an art AND a science. Know SQL Server internals. Know your environment, apps, and how they are used.

© 2012 House of Brick Technologies, LLC



SQL Server on VMware Boot Camp

VIRTUALIZING SQL SERVER ON VMware BOOT CAMP

Tuesday November 6th, 2012
HELD AT SQL PASS SUMMIT
Aspen Room
Sheraton Seattle Hotel
1400 Sixth Avenue, Seattle, WA 98101

11:00 – 12:00 Registration, Lunch, and Networking
12:00 – 4:30 Boot Camp Intensive



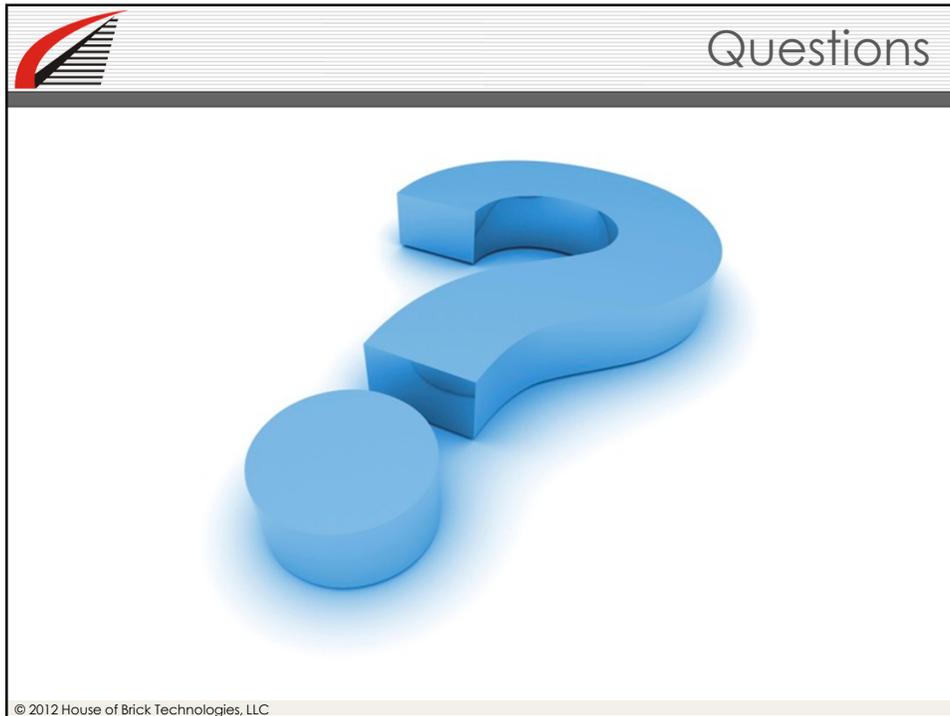
BOOT CAMP COURSE OVERVIEW

- Intro to SQL Server Virtualization**
 - Virtualization Trends •
 - Common Objections and Misconceptions •
- Physical Stack Fundamentals**
 - SQL Server Licensing Concepts •
 - Storage, vSphere Host, and Networking •
- Virtual Machine Layer**
 - VM and Guest Operating System Customization •
 - Virtual Storage Presentation Options •
 - Installation and Optimally Configure SQL Server •
- SQL Server on VMware Prototype**
 - Benchmarking and Baselining Performance •
 - Workload Selection •
- Beyond the Prototype**
 - Disaster Recovery and HA Options •
 - SQL Server Clustering •
 - SQL Server 2012 AlwaysOn •

House of Brick
EDUCATION

More information at <http://bit.ly/Plne7f>. Register at <http://bit.ly/PUKkbO>.

© 2012 House of Brick Technologies, LLC



Database Health and Performance
(AKA I can *prove* it's not my system's fault!)

SQL Saturday Lincoln – October 6, 2012

David Klee – Solutions Architect (@kleegeek)



House of Brick[®]
TECHNOLOGIES

© 2012 House of Brick Technologies, LLC