SQL intersection

SQL Server Architecture

Preconference Workshop #06

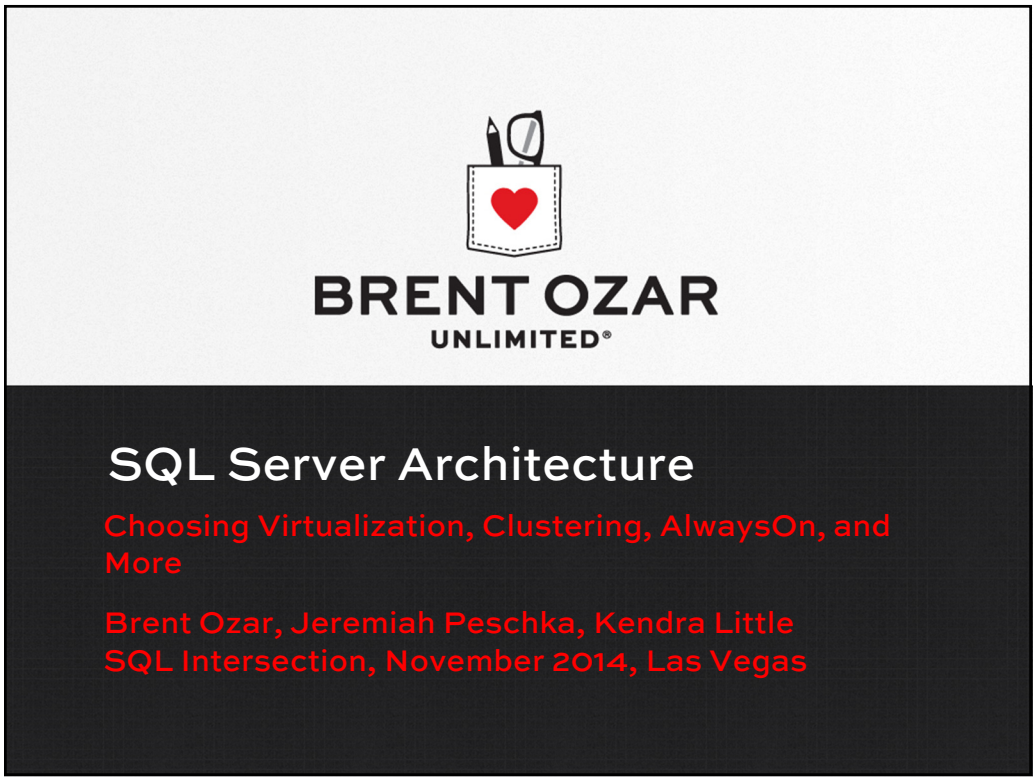
Choosing Virtualization, Clustering, AlwaysOn, and More

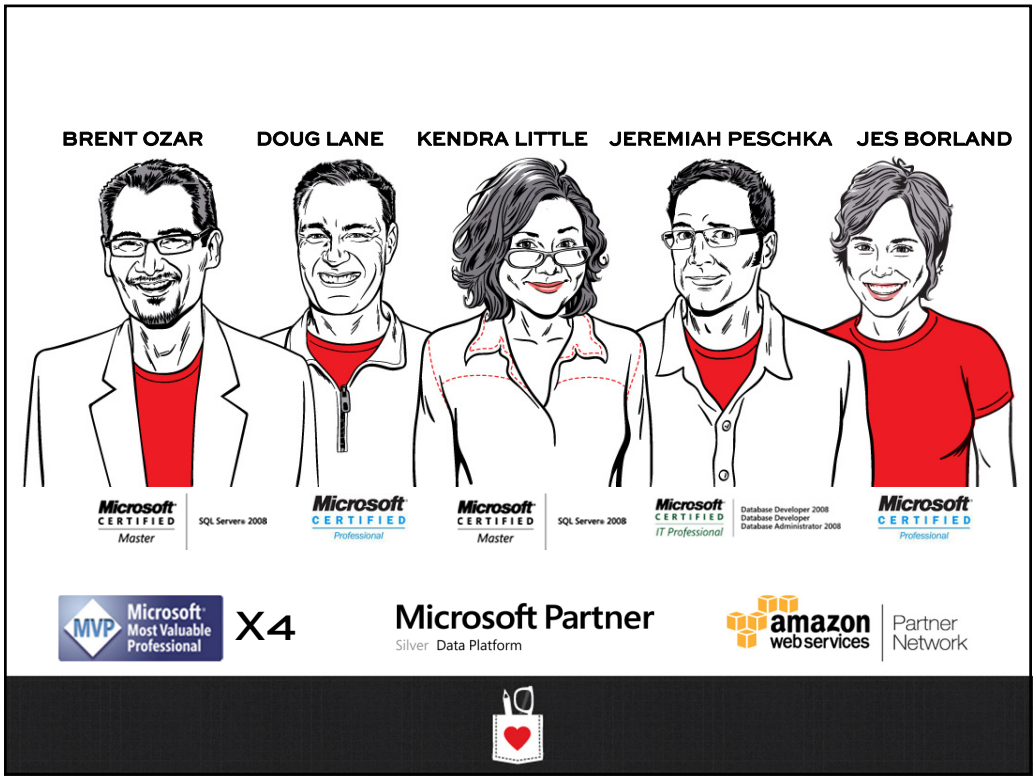
SQLintersection Fall 2014

Workshop: Sunday, November 9, 2014

written and presented by
Brent Ozar, Jeremiah Peschka, Kendra Little
http://www.brentozar.com

BRENT OZAR
UNLIMITED®


BRENT OZAR
UNLIMITED®
SQL Server Architecture
Choosing Virtualization, Clustering, AlwaysOn, and More
Brent Ozar, Jeremiah Peschka, Kendra Little
SQL Intersection, November 2014, Las Vegas


BRENT OZAR
DOUG LANE
KENDRA LITTLE
JEREMIAH PESCHKA
JES BORLAND
Microsoft CERTIFIED Master SQL Server 2008
Microsoft CERTIFIED Professional
Microsoft CERTIFIED Master SQL Server 2008
Microsoft CERTIFIED IT Professional Database Developer 2008 Database Developer Database Administrator 2008
Microsoft CERTIFIED Professional
MVP Microsoft Most Valuable Professional
X4
Microsoft Partner
Silver Data Platform
amazon web services Partner Network

About you

You need to recommend the right architecture for your company's SQL Servers.

The CTO is interested in the cloud.

The SAN Administrator wants to virtualize everything in-house.

One of the business users believes that SQL Server Availability Groups can work with both technologies and solve all your problems.

You've got input, but you need help.

Today, you'll learn...

When virtualization is the right fit (and when it's not)

How to mix platforms and features to keep SQL Server available

How to choose the right technology to scale out reads and writes

When the cloud is right for you

When to invest in application technology like caching

Today's agenda

Introduction

3 Numbers Every Architect Needs to Know – Brent

When the Cloud is Right for You – Jeremiah

How to Choose the Right Scale-Out Tech – Brent

Reworking Existing Architecture – Kendra

When to Invest in App Technology Like Caching – Jeremiah

Designing a New Solution - Kendra

SQL intersection

5

FAQ, BBQ, LOL

We're at level 200-300 today

- But ask your questions no matter what!
- Look for your "Montel"

These chairs get old!

- Get up and walk around the back/sides of the room
- We will take frequent breaks (not just the scheduled ones)
- Watch the timer to know when to get back in your seat

We love questions.

Asking questions can win you a prize.

SQL intersection

6

Tell us how you really feel.

Use EventBoard to fill out a session evaluation.

We want to improve, and your feedback helps.

Questions before we get started?

Agenda: 3 numbers every architect should know

- How busy is your server?
- How hard is it working?
- How much data do you have?

SQL intersection

6

How busy is your server?

- **Perfmon - SQLServer: SQL Statistics – Batch Requests/sec**
- **Not Transactions/sec**
- **Trend on an hourly basis and break it out by:**
  - Weekday vs weekend
  - Business hours vs after hours
  - Maintenance windows (backups, DB maintenance)

SQL intersection

7

When Batch Requests / Sec changes

- **Going up? More users, more API calls, new functionality in the application**
- **Going down? Less users or API calls, or code has been streamlined so that more work is done in less queries**
- **More data could mean an increase – but only if you have things like functions that are called row-by-row**

SQL intersection

8

Batch Requests / Sec seen in the field

- 0 – 1,000 – easy to handle with commodity hardware.
- 1,000-5,000 – be careful, because one bad change to a query can knock a commodity server over.
- 5,000-25,000 – if you're growing, you should be making a scale-out or caching plan.
- Over 25,000 – doable, but requires a lot of attention. Our record at the moment: ~150k sustained.

SQL intersection

9

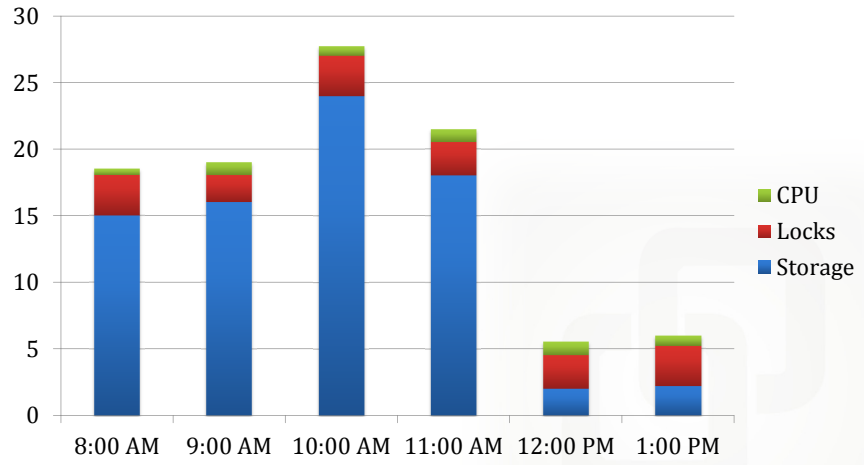
Agenda: 3 numbers every architect should know

- How busy is your server? Batch Requests / Time
- How hard is it working?
- How much data do you have?

SQL intersection

10

Server work = hours of wait time per hour



SQL intersection

11

How hard is your server working?

- **Dynamic Management View (DMV) sys.dm_os_wait_stats**
- **Tracked cumulatively over time, so snapshots are all you**
- **Start with: http://BrentOzar.com/go/getwaits**

- **Trend on an hourly basis and break it out by:**
  - Weekday vs weekend
  - Business hours vs after hours
  - Maintenance windows (backups, DB maintenance)

SQL intersection

12

When Wait Time per Hour changes

- **This metric is linked to Batch Requests: the more work the server is asked to do, the more it waits.**
- **Going up? Batch Requests could be going up, or the server is responding slower to the existing workload.**
  - Storage got slower, indexes were dropped, queries were tuned poorly, other processes are running on the server
- **Going down? Batch Requests could be too, or the server is responding faster to the workload.**
  - Indexes or queries were tuned, more memory was added

SQL intersection

13

Wait Time per Hour seen in the field

- **0 – Your server isn't doing anything.**
- **1 hour of waits – You're still not doing much. (each CPU core could have a query waiting, and your server has multiple CPU cores, right?)**
- **1 hour of waits X your number of cores – now we're working! And we should probably be tuning.**

SQL intersection

14

Agenda: 3 numbers every architect should know

- How busy is your server? Batch Requests / Time
- How hard is it working? Wait Time Per Hour
- How much data do you have?

SQL intersection

15

Total data size in GB/TB

- Rough guess: size of MDF/NDF data files on the server
- More accurate: used space in files. For each user database: SELECT SUM(size)FROM sys.database_files WHERE type = 0;
- Over time for the past: backup sizes in msdb.dbo.backupfile
- In SaaS environments, chart the number of databases too

SQL intersection

16

When Database Size changes

- Going up? Make sure it's related to user counts. If not, we started adding new features and storing new data, or storing backups inside the database.
- Going down? Hopefully people are dropping scratch tables. (Note: we're not distinguishing active vs logged data here)

SQL intersection

17

Database size seen in the field

- 1-150GB – easy to handle with SQL Server Standard Edition on commodity hardware.
- 150-500GB – easy to handle with Enterprise Edition
- Over 500GB – it starts to matter if this is active data, and how it's accessed (OLTP vs analytical)
- Over 1TB OLTP data – starts to get very challenging
- Our record at the moment: ~100TB per SQL Server, and ~10k-~12k databases per server.

SQL intersection

18

Recap: 3 numbers every architect should know

- How busy is your server? Batch Requests / Time
- How hard is it working? Wait Time Per Hour
- How much data do you have? Data Size in GB/TB

SQL intersection

19

Agenda

Defining "the cloud"

Separating responsibilities

Server requirements

Potential bottlenecks

Offloading opportunities

SQL intersection

2

What is "the cloud"?

The cloud is:
Someone else's servers
In a pool of shared resources
Where we don't have total control.

A cloud is probably virtual

Similar considerations as virtualization.

- Shared resources.
- Measure HA and DR.

Less control over features and functionality.

Examples of clouds

Public cloud

- AWS
- Azure
- Rackspace

Private cloud

- Large VM farm
- Internal IaaS (OpenStack, Eucalyptus)

Agenda

Defining "the cloud"

Separating responsibilities

Server requirements

Potential bottlenecks

Offloading opportunities

SQL intersection

Separating Responsibilities

SQL intersection

What is the cloud responsible for?

Your cloud provider has limited responsibility.

They provide:

- Facilities
  - Power
  - Cooling
- Infrastructure
  - Physical machines
  - Network equipment
  - Virtualization infrastructure

SQL intersection

What if hardware breaks?

That's not your problem.
You're renting it.

SQL intersection

What are you responsible for?

Everything else.

Does an SLA apply to me?

We don't need an SLA, it's the cloud!

If you don't know...

SLA: Service Level Agreement

RPO: how much data can we lose?
(Recovery Point Objective)

RTO: how long do we have to get it back?
(Recovery Time Objective)

This can also be related to performance:

- Users must authenticate in < 15ms
- The shopping cart must display in 250ms
- Any catalog page must render in 400ms
- Reports must perform < 100,000 on the OLTP system.

It sounds like we really do need an SLA!

If you don't have an SLA, get one!

SLAs will govern everything you do.

- What happens if your provider loses power?
- What happens if the eastern seaboard loses power?
- What happens if someone drops the storage offline?

Losing control of the underlying architecture will make you painfully aware that these things can happen.

Helpful SLA tips

Head over to http://brentozar.com/go/fail to get our HA/DR worksheet.

Talk to the users to put together a performance SLA.

- Start with 5 queries or pages.
- Time the whole application stack.
- You may need new monitoring tools to support this.

SQL intersection

Agenda

Defining "the cloud"

Separating responsibilities

Server requirements

Potential bottlenecks

Offloading opportunities

SQL intersection

Server Requirements

Current hardware?

What does the current hardware look like:

- CPU count & speed
- Memory
- Network
- Disks

CPU

Examine CPU usage over time.

Carefully monitor the source of CPU consumption.

- Try to tune for CPU.
- Tune indexes. Tune queries.
- Offload work.

Do you *really* need extra CPU?

- More CPU costs more money **per hour**.

CPU Costs

AWS – $0.15 / CPU / hour

Azure – between $0.16 / CPU / hour
and $0.30 / CPU / hour

Rackspace – between $0.17 / CPU / hour
and $0.34 / CPU / hour

Identifying CPU Consumers

Find your top CPU consumers with sp_BlitzCache™
http://brentozar.com/blitzcache

Identify missing indexes with sp_BlitzIndex®
http://brentozar.com/blitzindex

SQL intersection

Memory

How much data do you have today?

How much of that data are you querying right now?

How fast is your database growing?

What else is going to run on this server?
The correct answer is "nothing".

Do you *really* need Enterprise Edition?

SQL intersection

What about disks?

I/O is difficult to measure

What's an IOP?

- Amazon says it's 16KB.
  Unless it's an SSD. Those use 256KB IOPS.
- Azure is "one IOP is one IOP".
- Rackspace is eerily silent.
- SQL Server does 8KB or 64KB or, sometimes, 2MB IOPS.

Check out our memory sizing post for I/O measuring techniques.

- http://brentozar.com/go/memory
- Remember your requirements, they help you pick the size of your cloud server.

Everything has a measurable cost

CPU is by the hour.

Memory increases require CPU increases.

Disk costs along multiple dimensions:

- Capacity.
- Throughput.
- IOPS.

Network has ingress/egress charges.

Plan wisely, monitor carefully

Every provider lets you monitor and alert on usage conditions. Use them.

How should you size cloud servers?

SQL intersection

Measured guesswork

1. Evaluate your current utilization.
2. Extrapolate requirements.
3. Decide on your tolerance for overage.
4. Size appropriately.

SQL intersection

Two ways to move

Method 1: Forklift

Typically start with a 1:1 match and tune later.

You should still tune first, but…

- Requires additional time and effort.
- No guarantee of fixing any problems.
  But you probably will.

Tune after you move to the cloud.

Method 2: Redesign

Set aside an architecture period.

Redesign your application to work with the cloud.

Downsides:

- Time consuming.
- Still no guarantee that it will work.
- You'll have to tune after you get there.

SQL intersection

33

One last thing about servers...

SQL intersection

34

Don't think of it as a server

What do you have to do to get a new physical server?

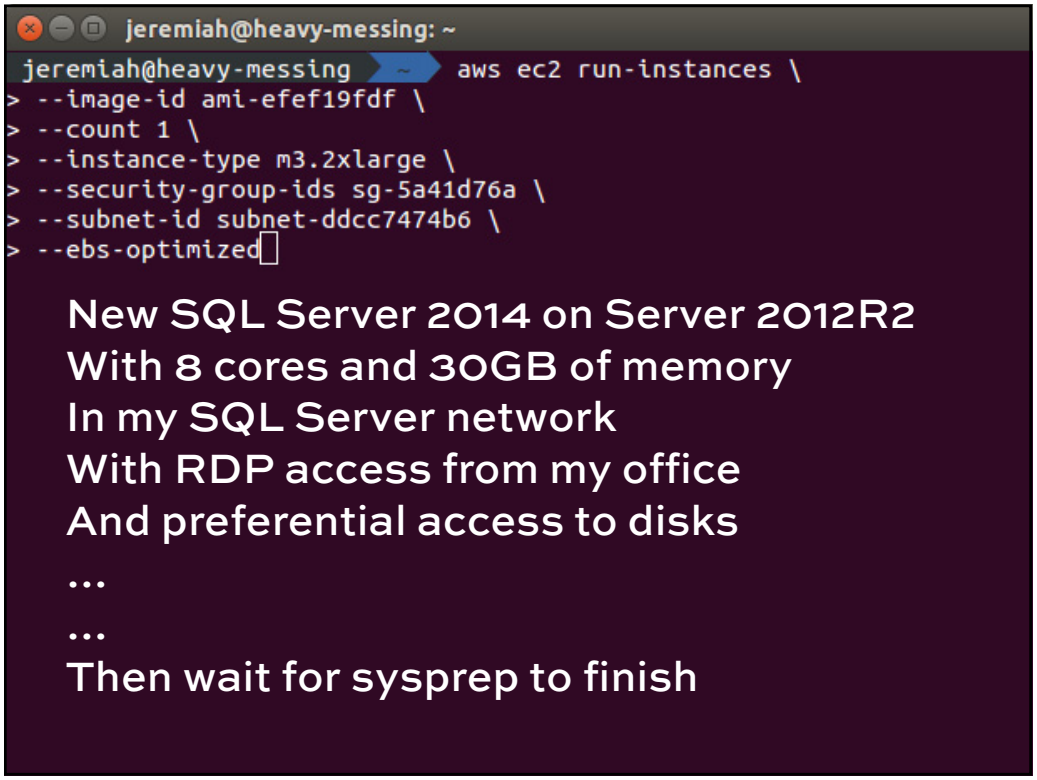
SQL intersection

35

Don't think of it as a server

What do you have to do to get a new physical server?

What do you have to do to get a new cloud server?



Scripting helps ensure success

Script SQL Server deployment and configuration.

Deploy new environments rapidly.

- Test changes
- Stand up new customers
- Disaster recovery

Sometimes, it's easier to rebuild a system from scratch than it is to troubleshoot.

Three Deployment Options

PaaS: SQL as a Service	IaaS Standard: SQL in an EC2 VM	IaaS Enterprise: SQL in an EC2 VM
Zero data loss Automatic failover Reduced features, datatypes, control	1 hour data loss 1 hour manual failover Same SQL Server you already know and love but less features	1 minute data loss 1 minute manual failover Data compression Snapshots
$4.4k per month per server	$4k per month per server	$56k up front, $1.5k per mo

* Per server means "per mirrored pair of servers"

Agenda

Defining "the cloud"

Separating responsibilities

Server requirements

Potential bottlenecks

Offloading opportunities

Potential Bottlenecks

Measure your top waits!

This will point you to your SQL Server bottlenecks.

http://www.brentozar.com/go/waits

Figure out when you have your worst waits:

- Storage waits after hours may not be bad.
- There are always mitigations for problems you find.

Don't get the fastest storage you can – try to mitigate storage issues first.

SQL intersection

43

But what about disks?!?

SQL intersection

44

Storage is terrible.

Storage is usually the slowest part of SQL Server.

The cloud is no exception.

Plan for narrow throughput to storage.

- Storage throughput is often limited by server size.
- Even fast servers typically have 1GbE.
  That's 125 MB/s
- If you need faster, you're going to pay for it.

SQL intersection

But what about IOPS?!?

SQL intersection

If you really need to know IOPS...

Break out perfmon and measure:

- Physical Disk – Disk Reads/sec
- Physical Disk – Disk Writes/sec

This is about as accurate as it's going to get.

Make some guesses from there about your IOPS requirements.

Accurately measuring disks

Look at dm_io_virtual_file_stats

- Average out your disk throughput.
- Look for high stalls.

Record metrics with perfmon

- Physical Disk – Disk Reads/sec
- Physical Disk – Disk Writes/sec
- Physical Disk – Avg. Disk sec/Read
- Physical Disk – Avg. Disk sec/Write

Use cloud provider metrics

Record your key metrics, as the provider sees them:

- Data throughput to disks.
- Disk latency.
- IOPS

Yes, you collected this at the OS level, but also collect this at the infrastructure level.

What if the server won't fit in the cloud?

Hypothetically…

You have a SQL Server that has a resource intensive workload.

- You have more than 32 CPUs.
- You have an all SSD SAN.

Management is all in on the cloud.

What do you do?

a) Ragequit
b) Sob quietly until the problem goes away
c) Deal with it

SQL intersection

51

Dealing with it: offloading work

Do less work in SQL Server.

Move functionality out of SQL Server:

- Cache common queries.
- Move search to external services.
- Sort in the application.
- Perform paging anywhere else.

Split the workload

- Report on server A
- Transactional work on server B
- Single sign on/user profiles on server C

Your cloud provider can help

Let someone else host other applications.

- Need caching? Spin up an Amazon ElastiCache or Azure Cache server.
- Need bulk data processing? HDInsight or Elastic MapReduce

Focus on your core skills and functionality.

You can always create specialized VMs later.

Common Blockers

What stops a move to the cloud?

Unwillingness to change.

Unrealistic expectations.

Lack of planning.

What should I measure?

Waits: http://brentozar.com/go/waits

Memory & I/O: http://brentozar.com/go/memory

Worst Queries: http://brentozar.com/blitzcache

SQL intersection

Wrapping Up

SQL intersection

In summation…

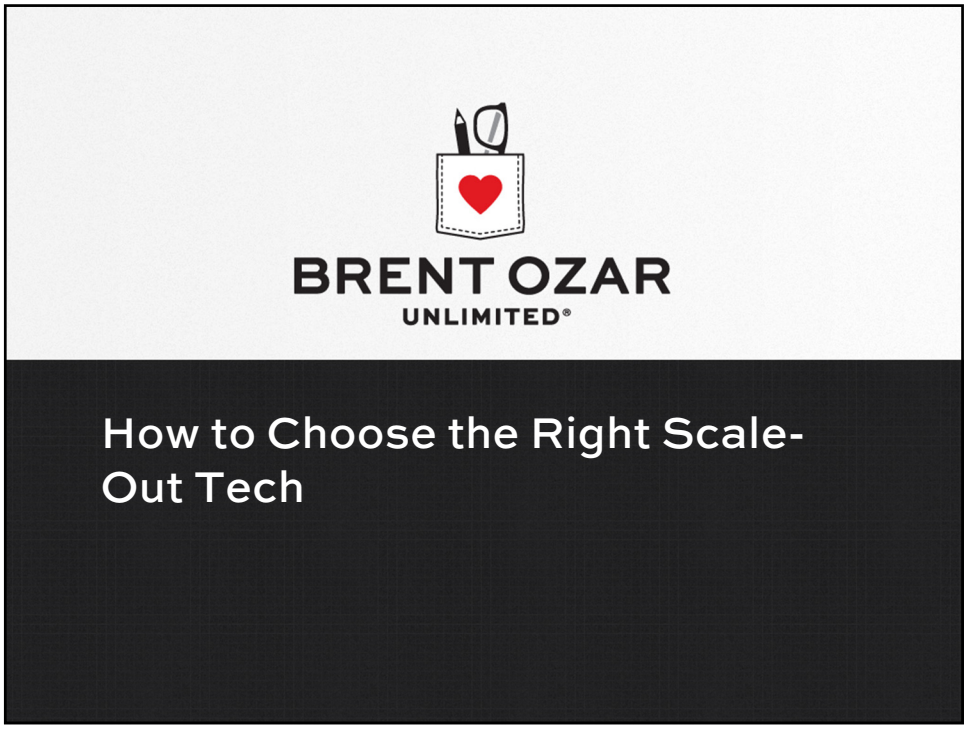
There's no hard and fast "This will work, that won't"

Measure your systems:

- Know load patterns.
- Understand SLAs.
- Understand the deployment platform.

SQL intersection

59



Matching up use cases to scale-out technology is hard.

- Many concurrent readers of small data (product inventory): Hekaton
- Many concurrent writers, but writing to different groups of records: sharding bidirectional transactional replication, merge replication
- Many readers of data that is constantly updated, but readers can deal with data that's a few minutes old: AlwaysOn AG readable secondaries, transactional replication
- A few readers that write horrifically bad queries that tend to block others, and they only need a snapshot of the data as of 9AM this morning: log shipping, database mirroring snapshots

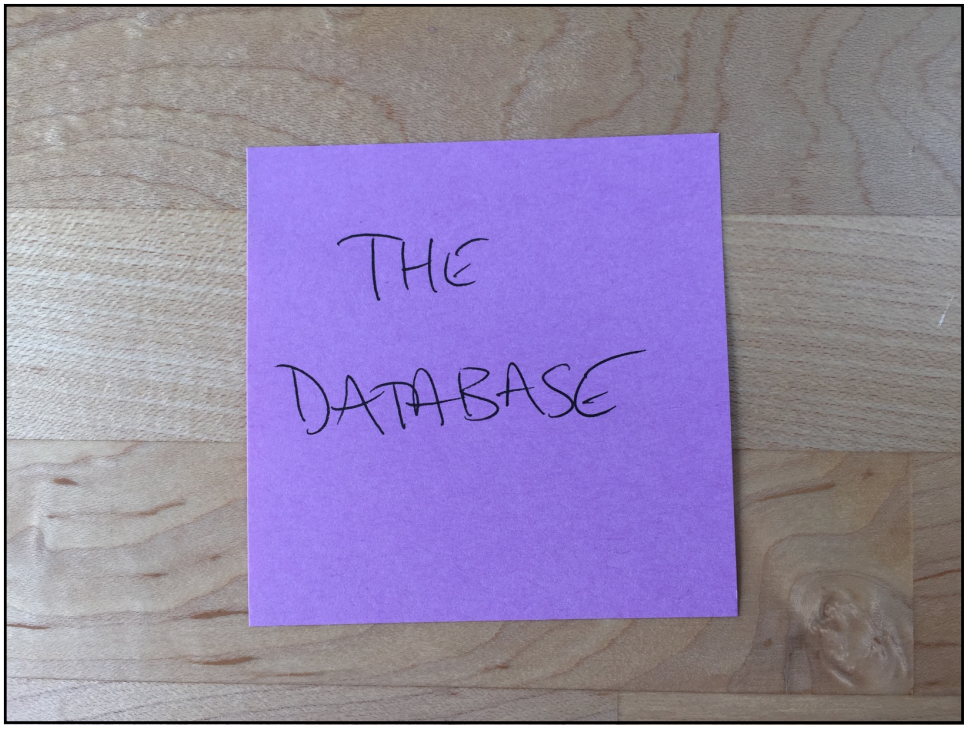
SQL intersection

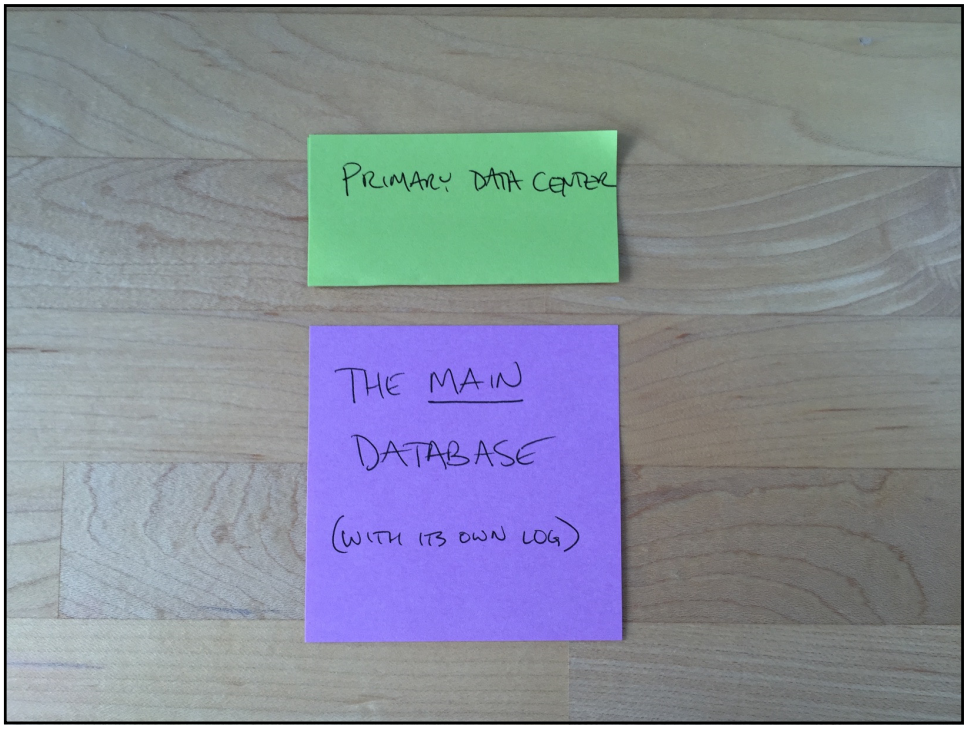
2

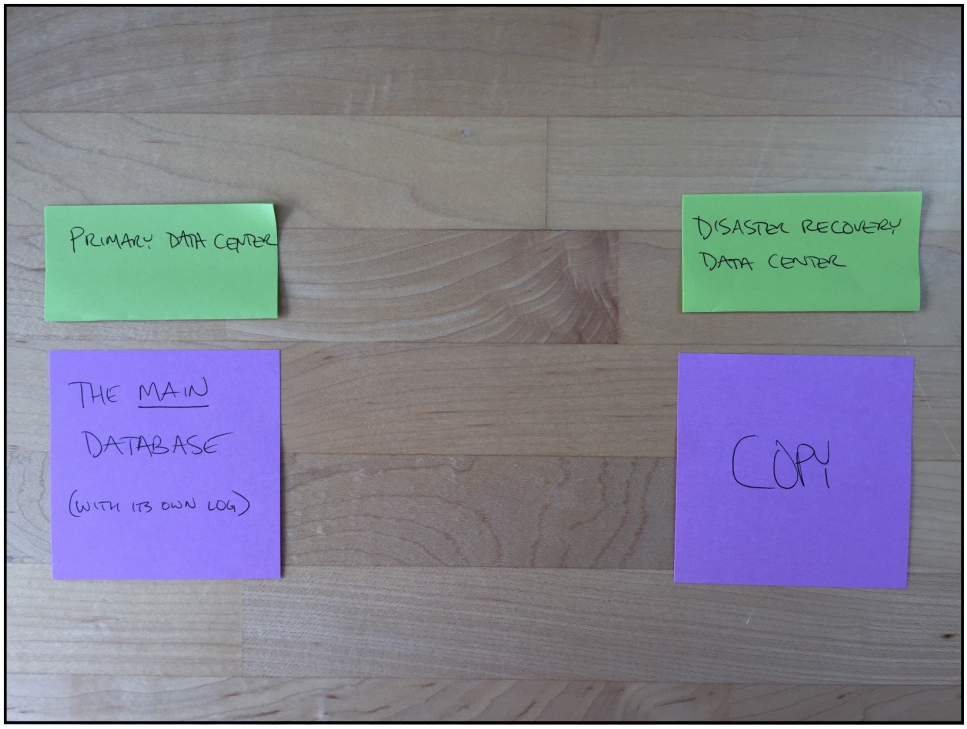
Session agenda

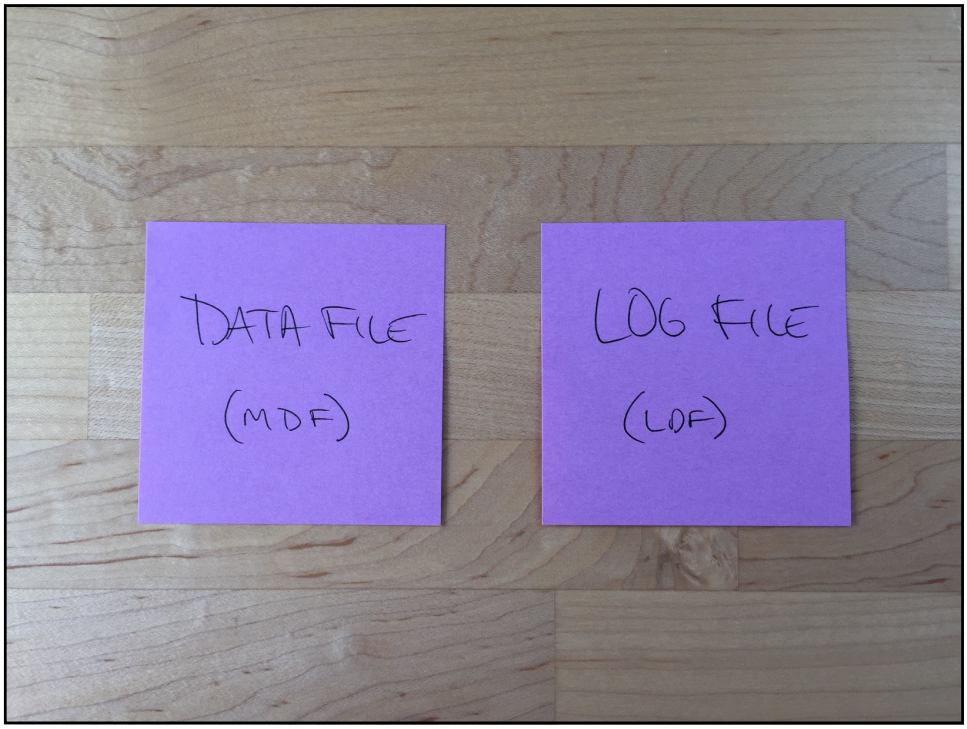
- Explain why scale-out starts with separation for HA/DR
- Lay out common data groupings with examples
- Designing your own data groupings
- Picking the right scale-out tech for your data groupings

SQL intersection




PRIMARY DATA CENTER
THE MAIN DATABASE
(WITH ITS OWN LOG)


PRIMARY DATA CENTER
THE MAIN DATABASE
(WITH ITS OWN LOG)
DISASTER RECOVERY DATA CENTER
COPY

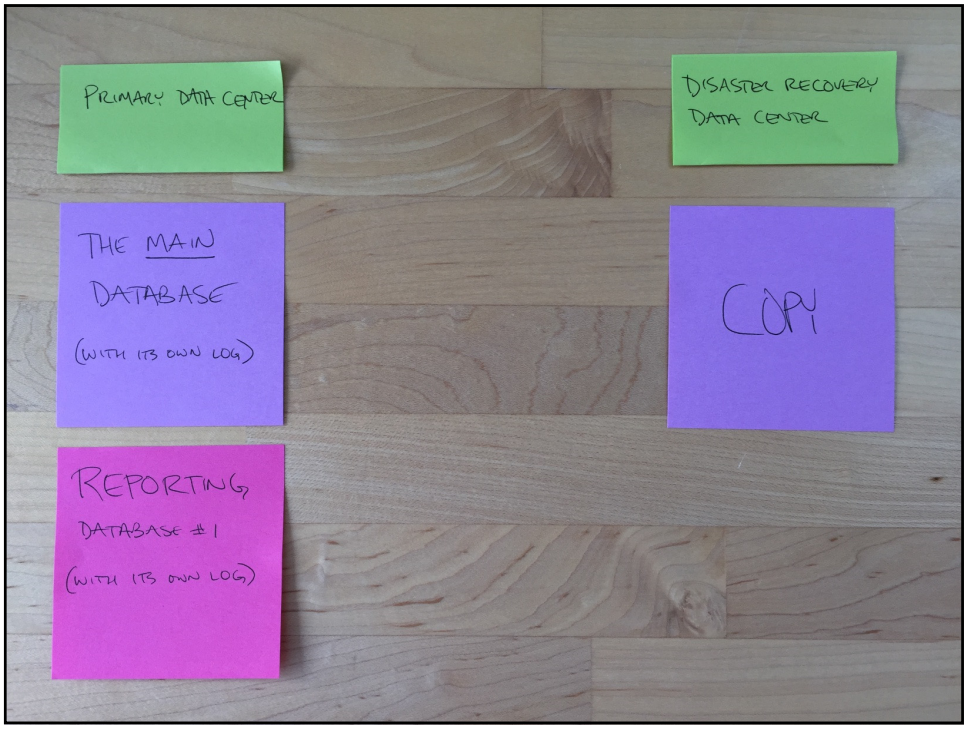


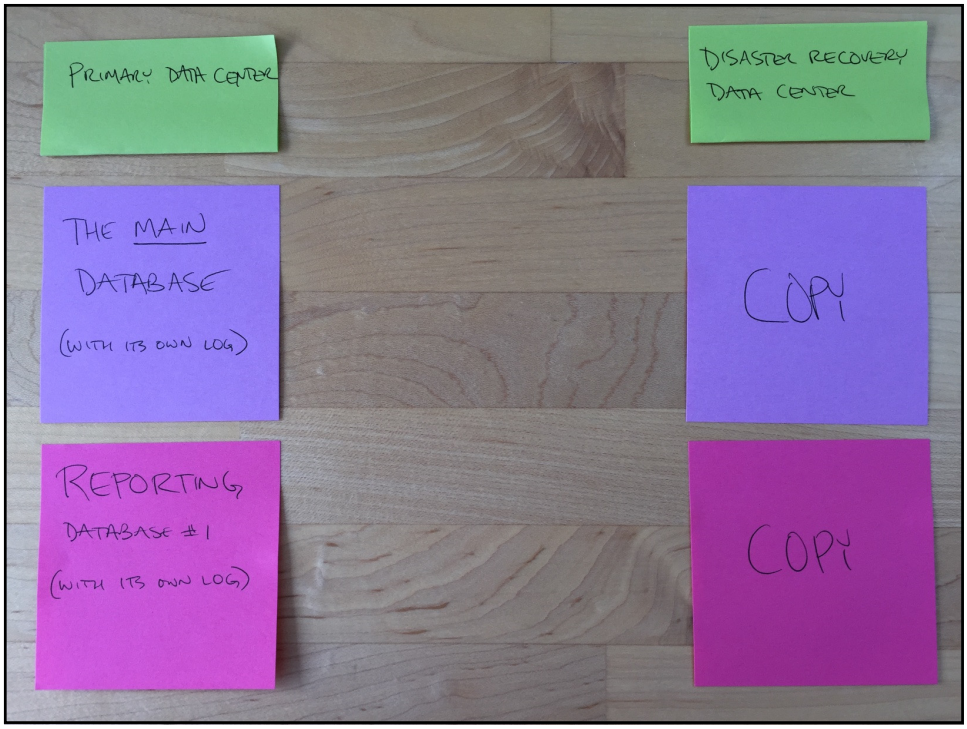
The problems with one log file

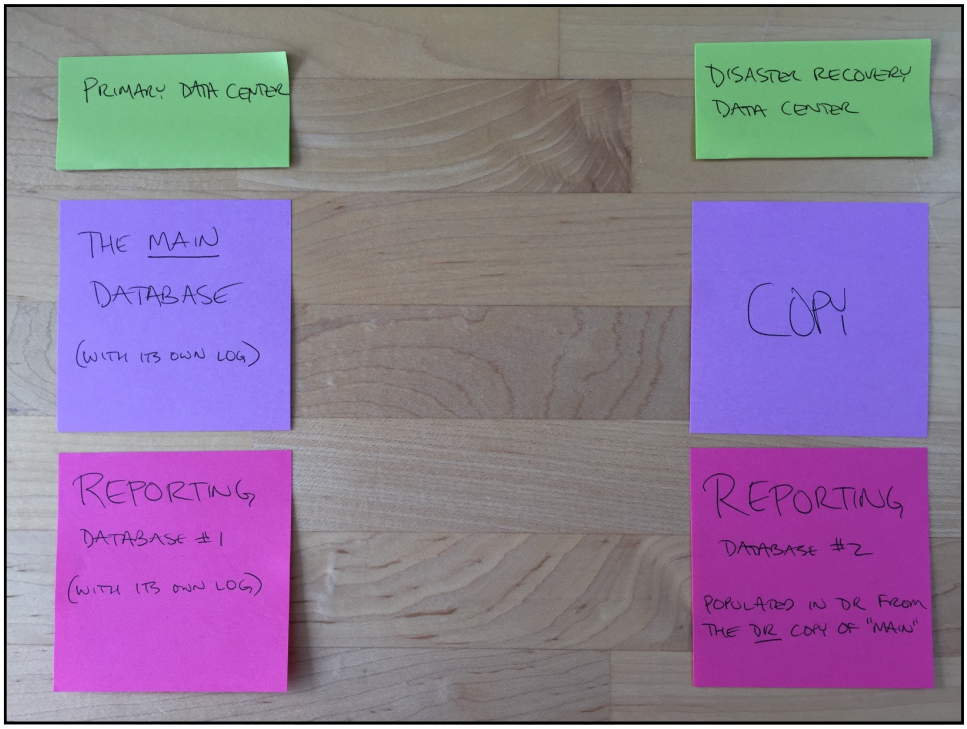
- We can't prioritize which data needs to get over to DR first
- Almost everything we do in the database gets logged (including maintenance tasks like rebuilding indexes)
- We can't set limits on how far behind log copies get: either we're synchronous, or we're asynchronous
- We can't really specify a low-value table to be non-logged*

SQL intersection

8


PRIMARY DATA CENTER
DISASTER RECOVERY DATA CENTER
THE MAIN DATABASE (WITH ITS OWN LOG)
COPY
REPORTING DATABASE #1 (WITH ITS OWN LOG)


PRIMARY DATA CENTER
DISASTER RECOVERY DATA CENTER
THE MAIN DATABASE (WITH ITS OWN LOG)
COPY
REPORTING DATABASE #1 (WITH ITS OWN LOG)
COPY

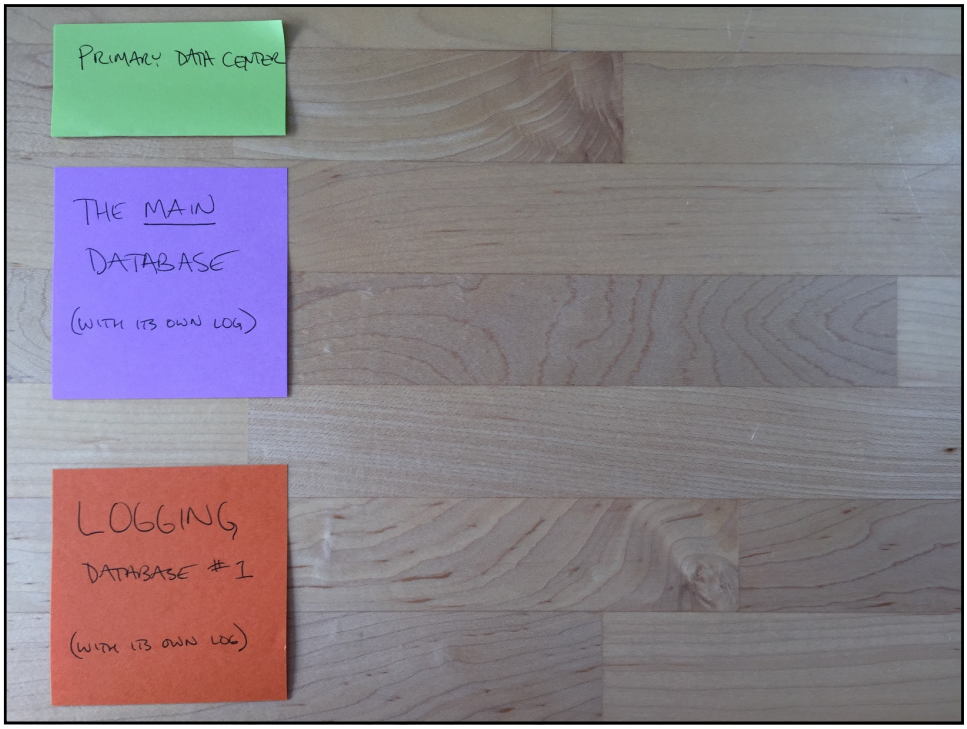


Considerations for splitting databases

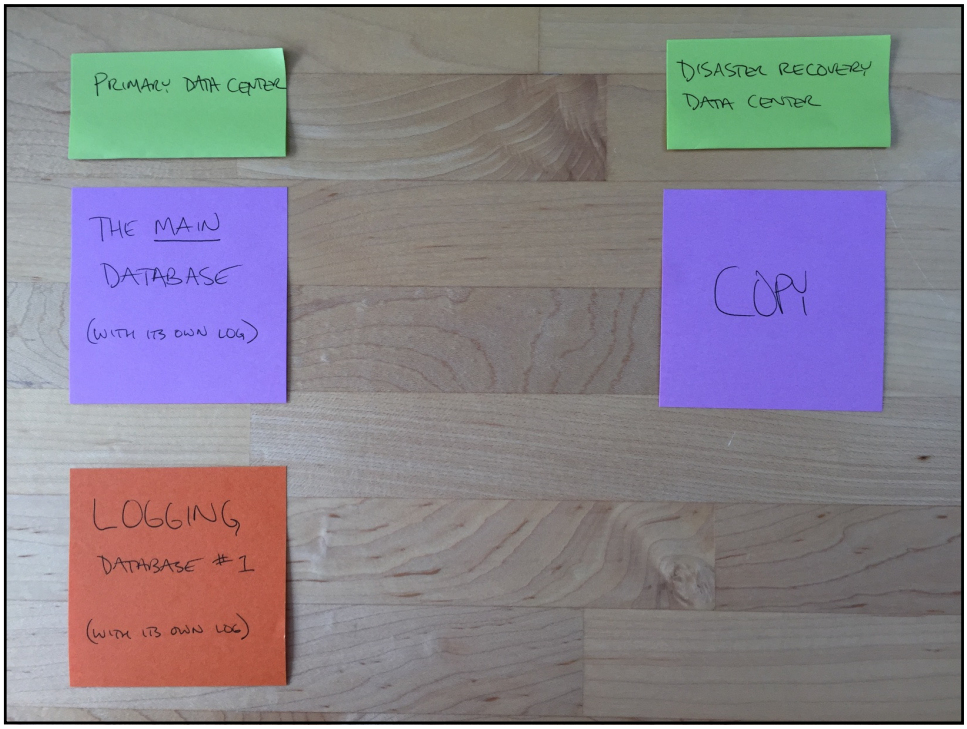
- **Could this table be easily recreated from other tables?**
- **Could the app be up without this table?**
- **Does this table have to be exactly in sync with other tables?**
- **If the table already exists, can I swap in a synonym or a view?**

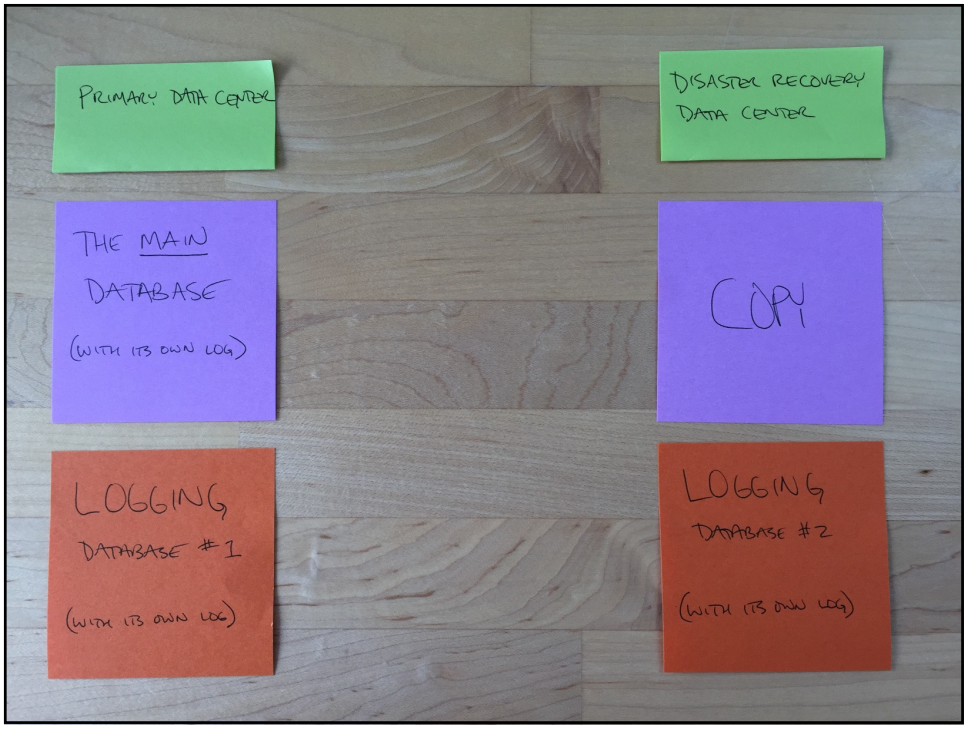
SQL intersection

12


PRIMARY DATA CENTER
THE MAIN DATABASE
(WITH ITS OWN LOG)
LOGGING DATABASE #1
(WITH ITS OWN LOG)


PRIMARY DATA CENTER
DISASTER RECOVERY DATA CENTER
THE MAIN DATABASE
(WITH ITS OWN LOG)
LOGGING DATABASE #1
(WITH ITS OWN LOG)


PRIMARY DATA CENTER
DISASTER RECOVERY DATA CENTER
THE MAIN DATABASE
(WITH ITS OWN LOG)
COPY
LOGGING DATABASE #1
(WITH ITS OWN LOG)


PRIMARY DATA CENTER
DISASTER RECOVERY DATA CENTER
THE MAIN DATABASE
(WITH ITS OWN LOG)
COPY
LOGGING DATABASE #1
(WITH ITS OWN LOG)
LOGGING DATABASE #2
(WITH ITS OWN LOG)

Common data groupings

Common data groupings

- **Main: our transactional data. The app won't run without this.**
- **Logging: low-value, heavy-write, rarely-read. App would still work with an empty version of these tables.**
- **Reporting: aggregated data that's easily recreated from other transactional data. App could be down without it, or could live with a stale copy of this data.**
- **Files/text/XML: voluminous data. We do point lookups of individual rows or groups of rows based on the OLTP data, but we don't directly join this data to the above data.**
- **Other examples: session state, product inventory, staging, scratch**

SQL intersection

18

For each grouping:

- **Backup and recovery**
  - What's the acceptable data loss and downtime?
  - How close does this have to be to other data groups?
- **Change rate**
- **Geographic or legal restrictions**
- **Consumers – who's querying the data, from where?**
- **Type of data, natural home**
  - Is this data best served from SQL Server, or another platform
- **Performance bottleneck:**
  - Do you need to scale this group out?
  - What's the bottleneck requiring scale-out?

SQL intersection

19

Wikipedia*

* No, of course Wikipedia doesn't run on Microsoft SQL Server, but let's pretend.

	Main OLTP (Users/Security)	Article Contents	Web Clicks	Web Click Analysis
RPO (Data Loss) for HA scenarios	0	1 day	1 day	(Unlimited, can reload from clicks)
RTO (Downtime) for HA scenarios	1 minute	1 minute	1 minute	1 day
Data producers/consumers location	Web/app servers	Web/app servers	Web/app servers	SSIS servers
How it's queried	Small point queries, insert/update/deletes	99% reads	99% writes	Loaded nightly in batches, then read with reports
Performance bottleneck	Concurrent reads/writes	Lots of read connections	Transaction log writes	Storage reads b/c too large for RAM
Change rate	.01% per day	.01% per day	1% per day	1% per day
Type of data, natural home	Relational	Text, XML	Relational, cubes	Relational, cubes

Amazon*

* But first, a few words about "scale" and "real time data."

	Main OLTP (Orders)	Product Catalog	Inventory	Shopping carts
RPO (Data Loss) for HA scenarios	0	1 day (can regenerate from source, partners)	?	Unlimited
RTO (Downtime) for HA scenarios	0	0	1 hour (assume all items are in stock)	0
Data producers/consumers location	Web/app servers	Produced by ETL, consumed by web servers	Web/app servers	Web/app servers
How it's queried	Small point queries, inserts only	99.999% reads	Small point queries, readers/updaters	Small point queries, readers/inserts
Performance bottleneck	Transaction log writes	Many concurrent readers	Many concurrent readers	Transaction log writes
Change rate	1% per day	1% per day	100% per day	100% per day
Type of data, natural home	Relational	XML	Relational or XML	Relational or XML

Hotel Survey Company

Hotel survey company

- Hotel managers design the guest feedback questions to ask
- Hotels export their list of guests every night and we import it
- We email the guests a link to their customized survey
- Guest takes the survey
- We work on the comments (translation, sentiment analysis)
- We alert the hotel managers on truly bad comments that need help
- Hotel managers can analyze their survey feedback over time

SQL intersection

26

	Main OLTP (The Database)
RPO (Data Loss) for HA scenarios	1 minute
RTO (Downtime) for HA scenarios	1 minute
Data producers/consumers location	Web/app servers, ETL processes
How it's queried	Small point queries, insert/update/deletes, plus analytical reports
Performance bottleneck	Many: locking, reads, transaction log writes
Change rate	1% per day
Type of data, natural home	Everything!

27

	Main OLTP (Hotels, Guests, Surveys)	Survey Servers (Emails, Questions)	Survey Answers	Hotel Reports
RPO (Data Loss) for HA scenarios	1 minute	(Unlimited, can reload from OLTP)	1 minute	(Unlimited, can reload from OLTP answers)
RTO (Downtime) for HA scenarios	1 day (as long as people can still take surveys)	1 minute	1 minute	1 day
Data producers/consumers location	ETL process: hotels sent text files in, plus answers	Web/app servers around the world	Web/app servers around the world	SSIS servers
How it's queried	Small point queries, insert/update/deletes	99% reads	99% writes, plus reads to push data to OLTP	Loaded nightly in batches, then read with reports
Performance bottleneck	None	Lots of read connections in bursts	Transaction log writes	Storage reads b/c too large for RAM
Change rate	1% per day	100% per day	100% per day	10% per day
Type of data, natural home	Relational	Relational	Relational and text comments	Relational, cubes

Hotel survey company: scaling out = separating.

- For us, after simply splitting out the data and building some ETL processes, we didn't need database server scale-out technologies.
- But how would you do this today with AWS and Azure services?
- Sending a survey notification, answering a survey, or translating a survey could really be done off a queue service instead.

Typical Multi-Purpose Web App with Reports

	Main OLTP	Logging	Reporting (DW)	Staging
RPO (Data Loss) for HA scenarios	0	1 day	(Unlimited, can regenerate from source)	(Unlimited, can restart the nightly process)
RTO (Downtime) for HA scenarios	1 minute	1 minute	1 hour	1 minute
Data producers/consumers location	Web/app servers	Web/app servers	End users nationwide w/Excel, Access	SSIS servers
How it's queried	Small point queries, insert/update/deletes	99% inserts, rarely queried	Loaded in nightly batches, then queried all day	Huge loads of writes at night
Performance bottleneck	Transaction log writes	Transaction log writes	Reading data from disk for reports	Storage writes/reads, CPU
Change rate	1% per day	1% per day	10% per day	100% per day
Type of data, natural home	Relational	XML	Relational, cubes	Relational

How this impacts tuning and scale-out technologies

- **Many concurrent readers of small data (product inventory): Hekaton**
- **Many concurrent writers, but writing to different groups of records: sharding bidirectional transactional replication, merge replication**
- **Many readers of data that is constantly updated, but readers can deal with data that's a few minutes old: AlwaysOn AG readable secondaries, transactional replication**
- **A few readers that write horrifically bad queries that tend to block others, and they only need a snapshot of the data as of 9AM this morning: log shipping, database mirroring snapshots**

SQL intersection

32

Separate before you scale.

Scaling out is seriously hard.

Don't scale out more data than you have to.

It's your turn.

	Main OLTP (Your current DB)			
RPO (Data Loss) for HA scenarios				
RTO (Downtime) for HA scenarios				
Data producers/consumers location				
How it's queried				
Performance bottleneck				
Change rate				
Type of data, natural home				

I don't expect you to have these answers yet,
and as you separate the data, the answers may change.

	Main OLTP (Your current DB)			
RPO (Data Loss) for HA scenarios				
RTO (Downtime) for HA scenarios				
Data producers/consumers location				
How it's queried				
Performance bottleneck				
Change rate				
Type of data, natural home				

Choosing the right scale-out tech for a data group

Wikipedia*

* No, of course Wikipedia doesn't run on Microsoft SQL Server, but let's pretend.

	Main OLTP (Users/Security)	Article Contents	Web Clicks	Web Click Analysis
RPO (Data Loss) for HA scenarios	0	1 day	1 day	(Unlimited, can reload from clicks)
RTO (Downtime) for HA scenarios	1 minute	1 minute	1 minute	1 day
Data producers/consumers location	Web/app servers	Web/app servers	Web/app servers	SSIS servers
How it's queried	Small point queries, insert/update/deletes	99% reads	99% writes	Loaded nightly in batches, then read with reports
Performance bottleneck	Concurrent reads/writes	Lots of read connections	Transaction log writes	Storage reads b/c too large for RAM
Change rate	.01% per day	.01% per day	1% per day	1% per day
Type of data, natural home	Relational	Text, XML	Relational, cubes	Relational, cubes

	Main OLTP (Users/Security)
RPO (Data Loss) for HA scenarios	0
RTO (Downtime) for HA scenarios	1 minute
Data producers/consumers location	Web/app servers
How it's queried	Small point queries, insert/update/deletes
Performance bottleneck	Concurrent reads/writes
Change rate	.01% per day
Type of data, natural home	Relational

Users/Security details

- **Query volume isn't all that high, but we're running into locking/blocking**
- **Writes alone could easily be handled by 1 database server, and reads could be handled by 2-3**
- **We just need to split out writers and readers (and we already tried RCSI)**

41

	Article Contents
RPO (Data Loss) for HA scenarios	1 day
RTO (Downtime) for HA scenarios	1 minute
Data producers/consumers location	Web/app servers
How it's queried	99% reads
Performance bottleneck	Lots of read connections
Change rate	.01% per day
Type of data, natural home	Text, XML

Article Contents details

- When querying the data for display on web pages, the data can be stale (a few minutes old, even hours)
- Even when the primary copy of the data fails, the readable copies need to stay online for read-only web pages
- A couple readable servers won't be enough: we may need dozens to handle reads

44

	Web Clicks
RPO (Data Loss) for HA scenarios	1 day
RTO (Downtime) for HA scenarios	1 minute
Data producers/consumers location	Web/app servers
How it's queried	99% writes
Performance bottleneck	Transaction log writes
Change rate	1% per day
Type of data, natural home	Relational, cubes

Web Clicks details

- Clicks: mostly insert-only
- Also read nightly by ETL process that builds analytics cubes for the Web Click Analysis column
- One server alone can't handle the write workloads

47

Wait – why are we leaving SQL Server?

**When you've separated your data out into groups,
the easy fix is isolating the problem group onto its own database server.**

You can isolate it with minimal changes to code or connection libraries.

But when one scaled-up server still isn't enough for it, and you split the data across multiple servers, you may need to:

- **Re-combine the data for reporting**
- **Build conflict resolution processes**
- **Change your application to connect to the right server**

And if you're doing all that, it's a great time to think about the data's home.

SQL intersection

	Web Click Analysis
RPO (Data Loss) for HA scenarios	(Unlimited, can reload from clicks)
RTO (Downtime) for HA scenarios	1 day
Data producers/consumers location	SSIS servers
How it's queried	Loaded nightly in batches, then read with reports
Performance bottleneck	Storage reads b/c too large for RAM
Change rate	1% per day
Type of data, natural home	Relational, cubes

Analysis details

- **Analytical data can be independent from the source data format**
- **Well-known reporting front ends: Excel, SSRS, BusinessObjects**
- **Great fit for relational or cubes (but we still may need to scale out)**

Amazon*

* But first, a few words about "scale" and "real time data."

	Main OLTP (Orders)	Product Catalog	Inventory	Shopping carts
RPO (Data Loss) for HA scenarios	0	1 day (can regenerate from source, partners)	?	Unlimited
RTO (Downtime) for HA scenarios	0	0	1 hour (assume all items are in stock)	0
Data producers/consumers location	Web/app servers	Produced by ETL, consumed by web servers	Web/app servers	Web/app servers
How it's queried	Small point queries, inserts only	99.999% reads	Small point queries, readers/updaters	Small point queries, readers/inserts
Performance bottleneck	Transaction log writes	Many concurrent readers	Many concurrent readers	Transaction log writes
Change rate	1% per day	1% per day	100% per day	100% per day
Type of data, natural home	Relational	XML	Relational or XML	Relational or XML

	Shopping carts
RPO (Data Loss) for HA scenarios	Unlimited
RTO (Downtime) for HA scenarios	0
Data producers/consumers location	Web/app servers
How it's queried	Small point queries, readers/inserts
Performance bottleneck	Transaction log writes
Change rate	100% per day
Type of data, natural home	Relational or XML

Shopping Cart details

- **dbo.Cart with:**
  - CustomerID INT
  - CartContents XML
- **<cart>**
  - <customerid>12346</customerid>
  - <item>
    - <productid>7891350</productid>
    - <quantity>1</quantity>
  - </item>
  - <item>
    - <productid>735616</productid>
    - <quantity>10</quantity>
  - </item>
- **</cart>**

56

	Inventory
RPO (Data Loss) for HA scenarios	?
RTO (Downtime) for HA scenarios	1 hour (assume all items are in stock)
Data producers/consumers location	Web/app servers
How it's queried	Small point queries, readers/updaters
Performance bottleneck	Many concurrent readers
Change rate	100% per day
Type of data, natural home	Relational or XML

Inventory: let's pretend.

- **All our order data resides on a single SQL Server, not sharded out.**
- **We don't need pinpoint accuracy: if we're off by a few items, that's OK.**
- **Our developers aren't smart enough to implement caching, and we have to query the database every time.**

60

Recapping how to decide

Session agenda

- Lay out your app's data groupings (ideally before you run into scaling problems, because these changes can take time)
- Set RPO/RTO goals for each data grouping
- Define the performance bottleneck for the grouping
- Separate and scale up first: don't scale out anything you don't have to
- Pick the right home and scale-out method for an individual group, not a database full of unrelated objects

SQL intersection

63

We're going to work in teams

A sick SQL Server needs your help!

Your team will get information about the current architecture and the problem

- Work together for 20 minutes
- Diagram out architecture changes you recommend

We'll meet back and go over what you came up with, and what I came up with

There's more than one answer to each problem.

SQL intersection

2

The teams (and mentors)

Team 1 - Jeremiah

- Sad scale-up SQL Server

Team 2 - Kendra

- Scale out SQL Server pains (replication)

Team 3 - Brent

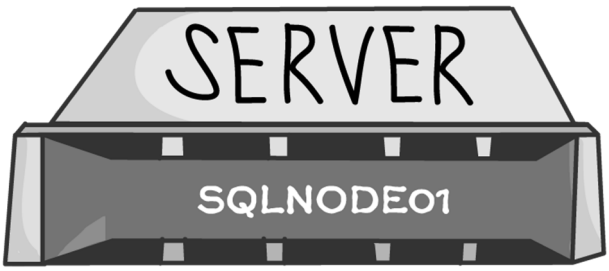
- SQL Server Availability Group is not-so-available

When recommending architecture, consider...

Hardware

You can recommend changes to:

- Memory
- CPUs (but consider licensing)
- Storage
- Other physical components

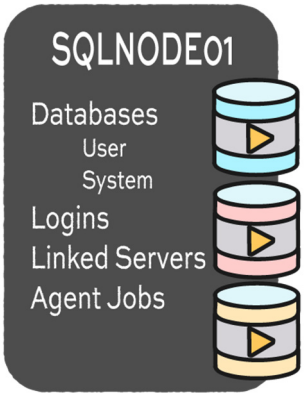


SQL intersection

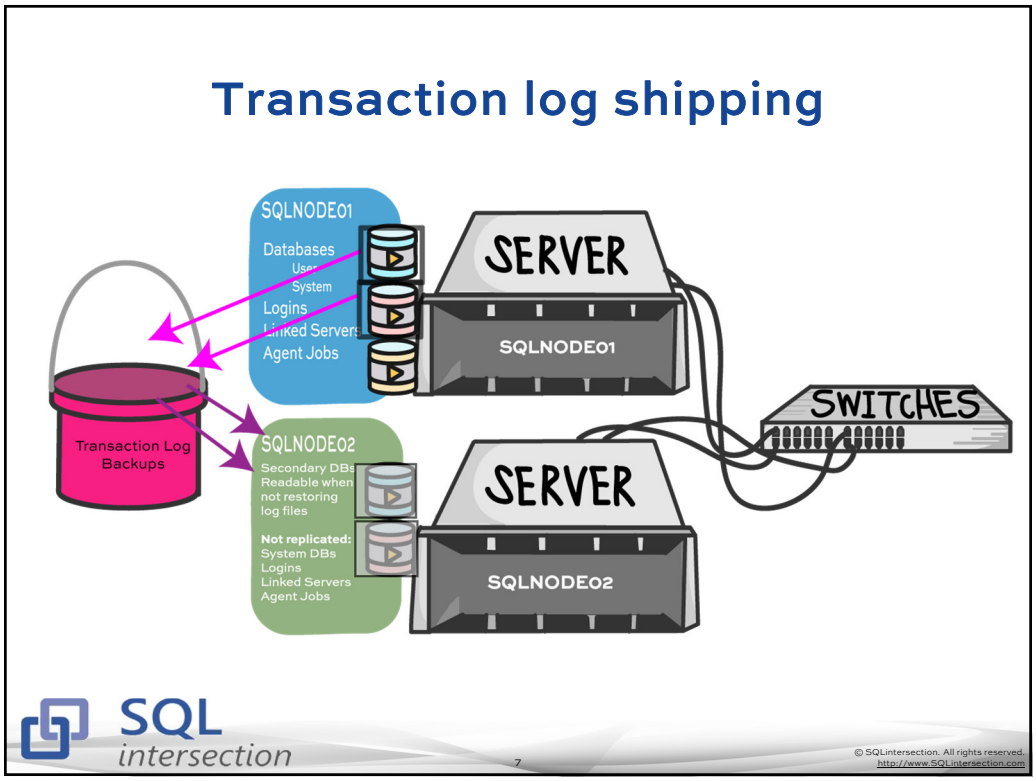
SQL Server instance and code

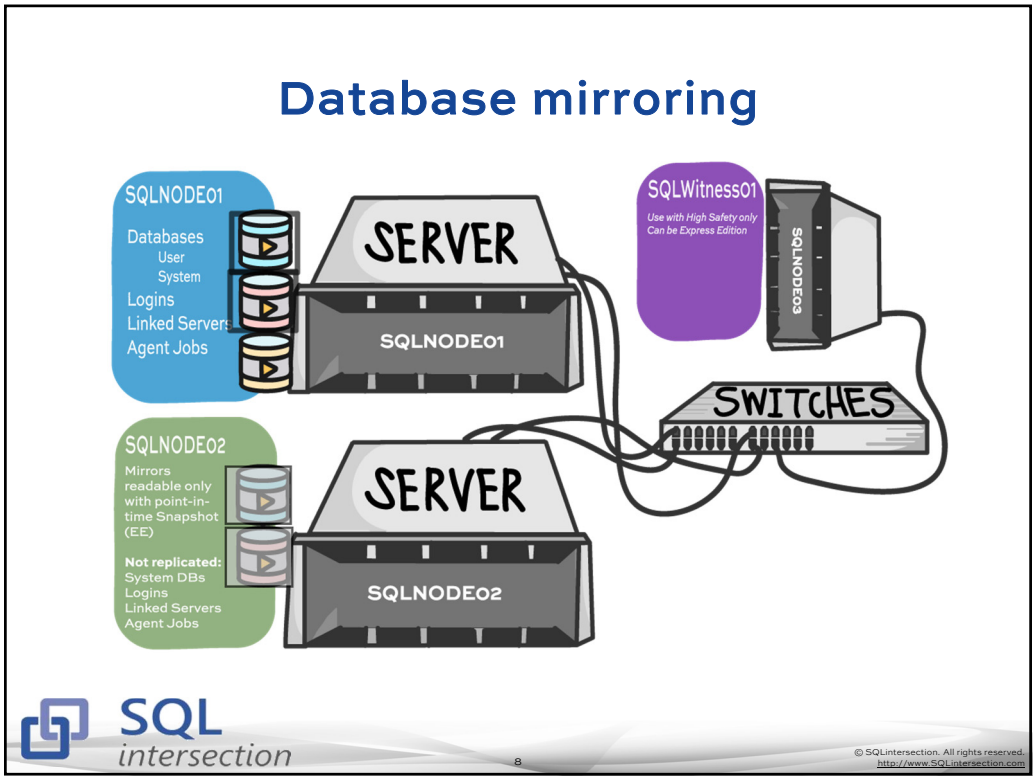
You can recommend changes to:

- Isolation levels
- SQL Server instance settings
- Application patterns

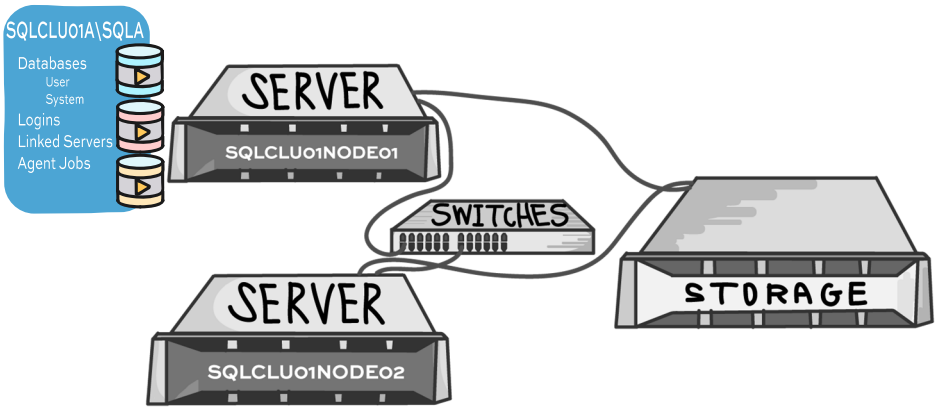


SQL intersection


Transaction log shipping
SQLNODE01
Databases
User
System
Logins
Linked Servers
Agent Jobs
SERVER
SQLNODE01
SWITCHES
Transaction Log Backups
SQLNODE02
Secondary DBs
Readable when not restoring log files
Not replicated:
System DBs
Logins
Linked Servers
Agent Jobs
SERVER
SQLNODE02
SQL intersection
7
© SQLintersection. All rights reserved.
http://www.SQLintersection.com


Database mirroring
SQLNODE01
Databases
User
System
Logins
Linked Servers
Agent Jobs
SERVER
SQLNODE01
SQLWitness01
Use with High Safety only
Can be Express Edition
SQLNODE03
SWITCHES
SQLNODE02
Mirrors readable only with point-in-time Snapshot (EE)
Not replicated:
System DBs
Logins
Linked Servers
Agent Jobs
SERVER
SQLNODE02
SQL intersection
8
© SQLintersection. All rights reserved.
http://www.SQLintersection.com

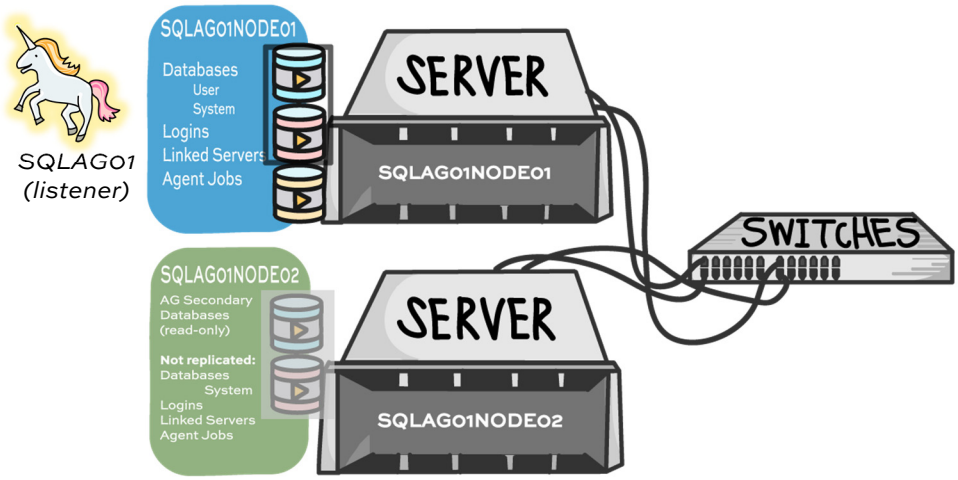
Failover cluster



SQL intersection

9

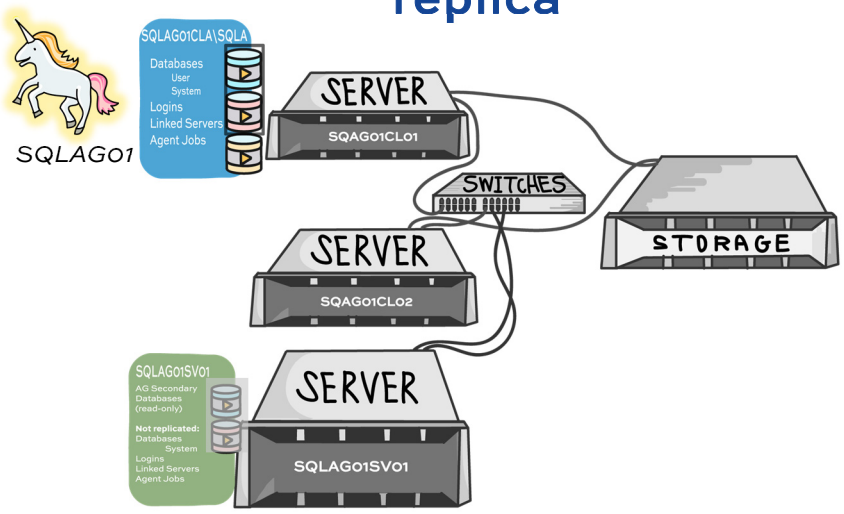
AlwaysOn availability group



SQL intersection

10

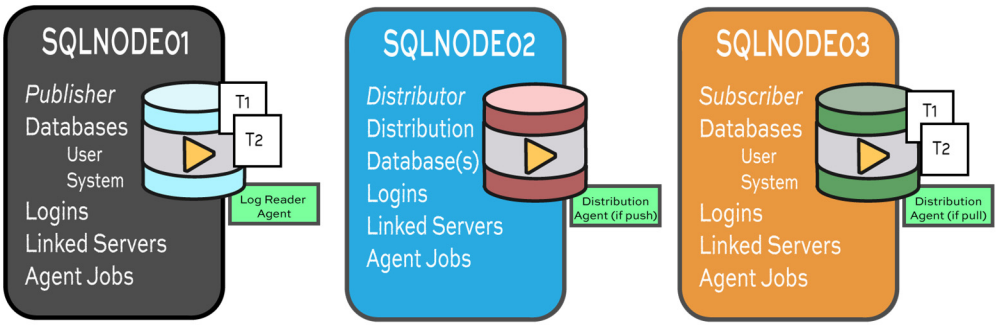
AlwaysOn failover cluster instance + availability group replica



SQL intersection

11

SQL Server transactional replication



Changes to tables in a replication publication are read from the transaction log

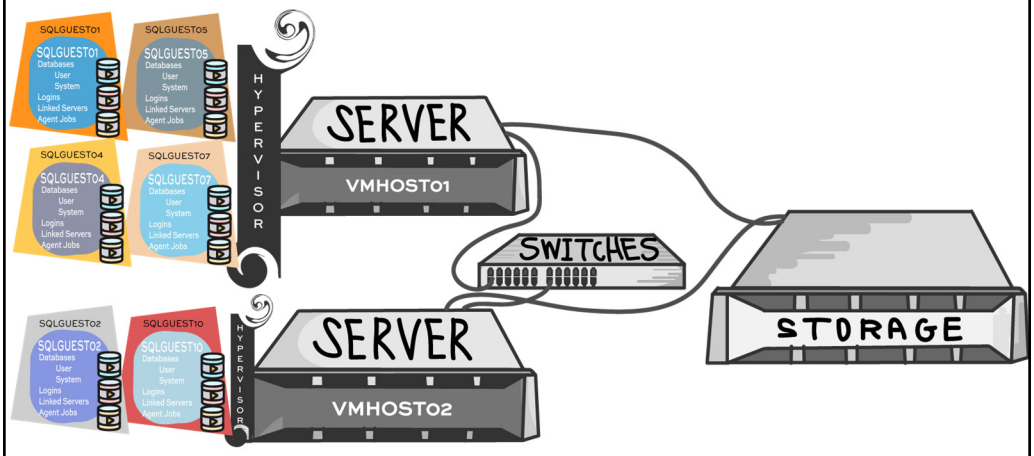
Commands to update subscribers are generated for each change and stored in the distribution database

Changes are applied for tables defined in the subscription

SQL intersection

12

Virtualized SQL Server instances



SQL intersection

People

You can recommend changes to:

- Monitoring tools
- Hiring / team members
- Processes

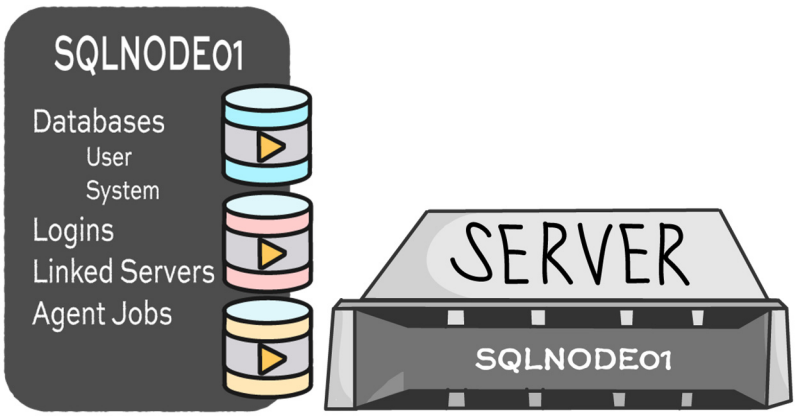
SQL intersection

Team 1
Sad scale-up
SQL Server

SQL intersection

15

Scale up SQL Server



SQL intersection

16

Team 1 RPO/RTO

Team:

- 1 full time DBA
- 9 developers

Budget: "We can spend whatever it takes to make it right."

	Server Offline	Corrupt data	"Oops" deletes	Datacenter offline
RPO: Minutes of data loss allowed	5 minutes	5 minutes	5 minutes	1 hour
RTO: Minutes of downtime allowed	5 minutes	2 hours	4 hours	1 week

SQL intersection

17

SQL Server 2008R2 SP3, Enterprise Edition

SQL Server data sizes:

- 500 GB of data across 8 related databases
- 100 GB tempdb

Server basics:

- 768 GB of memory
- 32 logical processors (hyperthreading)
- SAN storage

Backups

- Back up to a file share, then copied to the cloud
- Full backups nightly
- Transaction log backups every 1 minute

SQL intersection

18

Here's the problem...

Application profile:

- Used constantly by many web services and applications (OLTP)
- Large amounts of data are imported multiple times a day. Insert, update, delete speed is critical.
- Reporting queries also run against the same instance. Reports must be accurate and real-time.

Bottlenecks:

- Row and page level lock waits
- Deadlocks

Usually, things are fine

When large reports run, the server grinds to a halt

Answer three questions

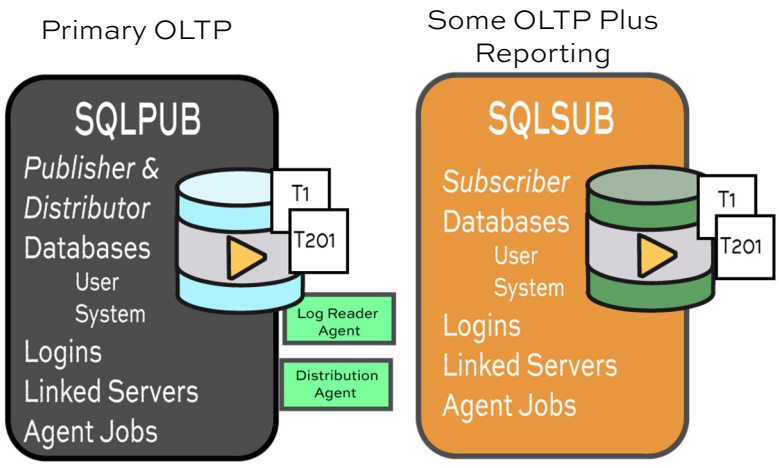
1. What are the top changes you would explore?

1. What are the benefits to your approach?

1. What are the risks?

Team 2
Scale out pains

Scale out SQL Server



All 201 tables are replicated

Team 2 RPO/RTO

Team:

- Two full time DBAs
- 25 developers

Budget: "We can spend some, but we really hope we don't have to buy Enterprise licenses."

	Server Offline	Corrupt data	"Oops" deletes	Datacenter offline
RPO: Minutes of data loss allowed	N/A. The data is completely from other sources, this is a pure secondary instance.			
RTO: Minutes of downtime allowed	1 hour	1 hour	12 hours	1 day

SQL intersection

23

SQL Server 2012 RTM, Standard Edition

The publisher and subscriber are set up the same way

Replication was put in place to spread the load out to more than one server

SQL Server data sizes:

- 450 GB of data (and growing) in one database
- 60 GB tempdb

Server basics:

- 128 GB of memory
- 16 logical processors (hyperthreading)
- SAN Storage (an old, overburdened SAN)

SQL intersection

24

Here's the problem...

Application profile:

- Replication is constantly bringing data over
- Some application connection strings that require real-time data point to the instance
- Internal business users run reports against the instance during daytime hours.
  - They are used to real-time data, but would be willing to have data as of the prior midnight if it would stop the pain.

The problem:

- When activity on the primary server gets high, replication gets very far behind
- The application queries pointing to the secondary then have data that is wrong
- Sometimes replication has to be rebuilt and that disrupts the report users

SQL intersection

25

Answer three questions

1. What changes do you recommend? (Diagram it out)

1. What are the benefits?

1. What are the risks?

SQL intersection

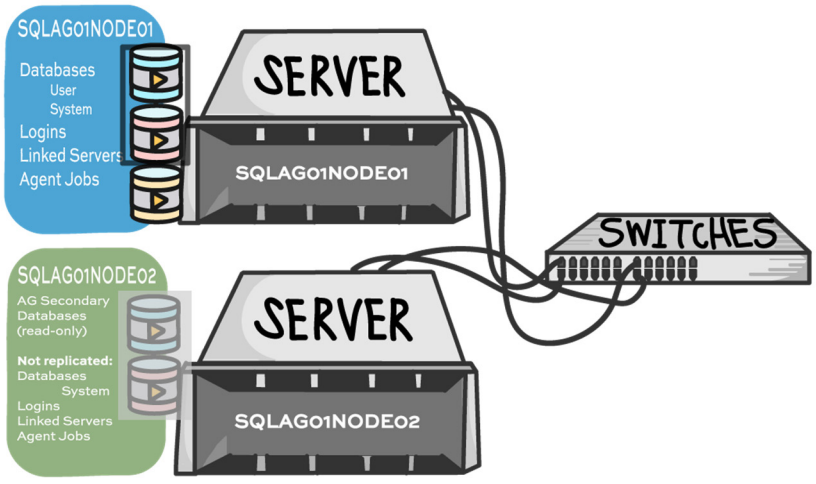
26

Team 3:
"Our Availability Group is bringing us down. Literally."

SQL intersection

27

Availability group



SQL intersection

28

Team 3 RPO/RTO

Team:

- 5 developers

Budget: "We've invested a lot. We don't want to spend much more."

	Server Offline	Corrupt data	"Oops" deletes	Datacenter offline
RPO: Minutes of data loss allowed	5 minutes	5 minutes	5 minutes	??? (No DR plan)
RTO: Minutes of downtime allowed	5 minutes	5 minutes	5 minutes	??? (No DR plan)

SQL intersection

29

SQL Server 2012 RTM, Enterprise Edition

SQL Server data sizes:

- 120 GB of data (and growing) in one database
- 20 GB tempdb

Server basics:

- 256 GB of memory
- 16 logical processors (hyperthreading)
- Local SSD Storage (PCI Express Cards)
- Dual NICs with network teaming
- Windows Server 2008R2 Enterprise

Availability Group configuration

- Synchronous AG with automatic failover
- Two nodes, no disk or fileshare witness
- One node licensed, not reading from the other node

SQL intersection

30

Everyone's unhappy

Application profile:

- Web applications doing internet retail

The problem:

- SQL Server is periodically unresponsive
- The SQL Server databases sometimes go offline
- No similar dev environment

Answer three questions

1. What changes do you recommend? (Diagram it out)

1. What are the benefits?

1. What are the risks?

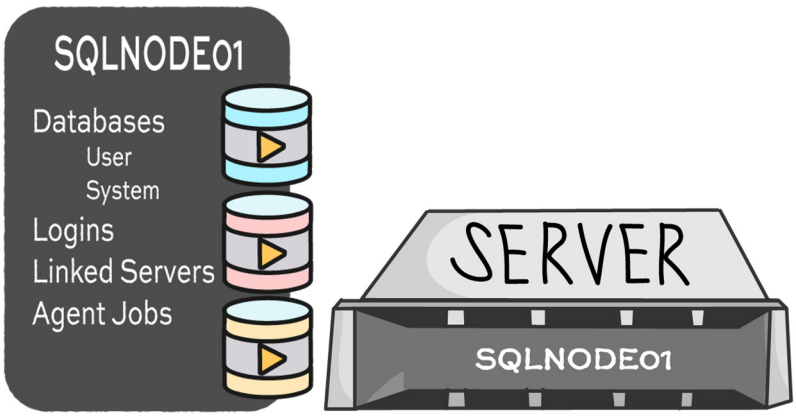
Time to work!

Possible solutions

Team 1
Scale-up sadness

SQL intersection

What did you recommend?

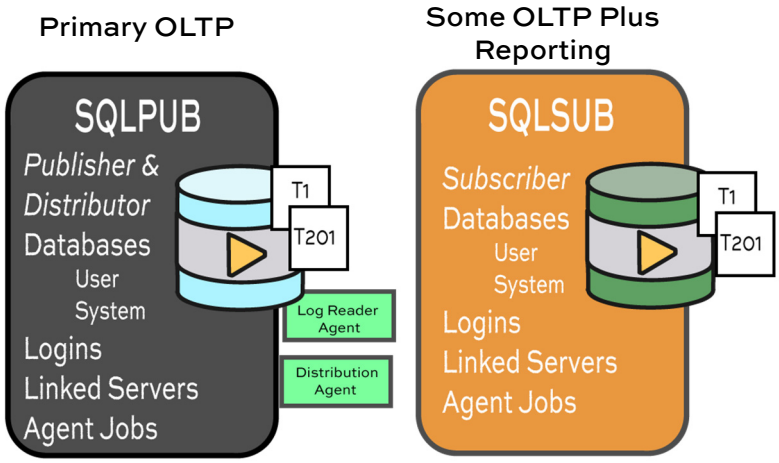


SQL intersection

Team 2
Scale-out pains

SQL intersection

How did you approach this problem?



All 201 tables are replicated

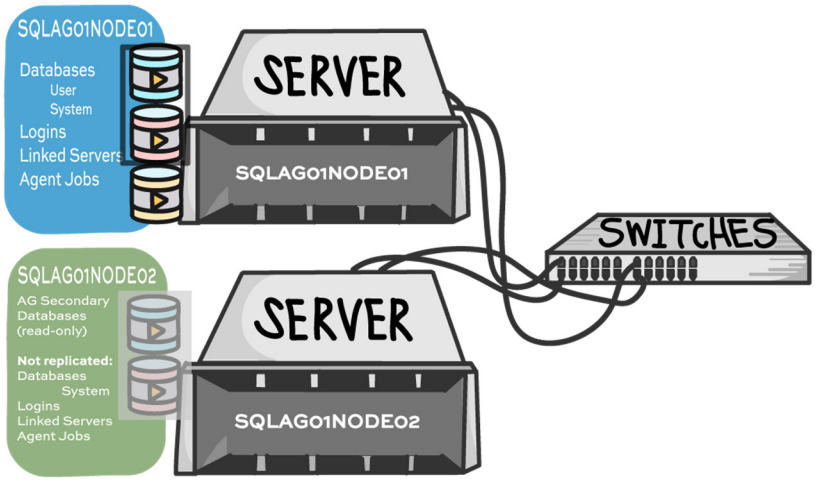
SQL intersection

Team 3
Availability group not-so-available

SQL intersection

41

Can this AG be saved?



SQL intersection

42

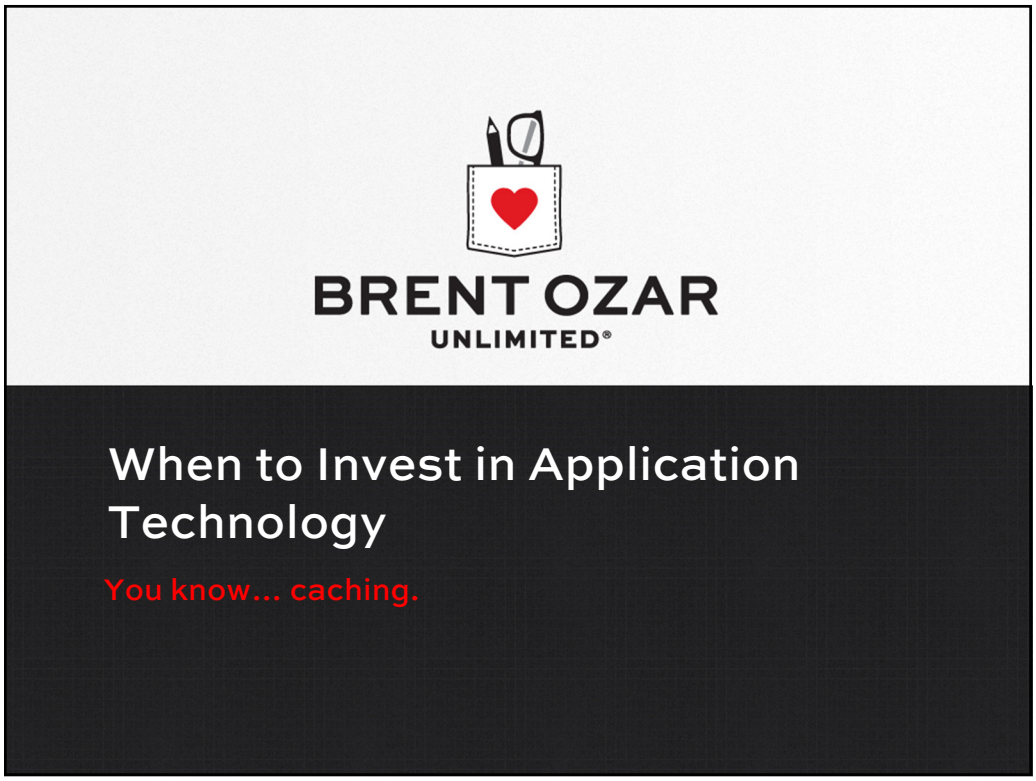
The real world is full of hard choices

It's not *only* about technology

It's also about

- The team
- How they work together

There's something to that old trio of "People, Process, Technology"


BRENT OZAR
UNLIMITED®
When to Invest in Application Technology
You know... caching.


Your first server...
SQL intersection
2
© SQLintersection. All rights reserved.
http://www.SQLintersection.com

After a while, there's load. And you decide to buy a new server.

And you got a bigger server...

When do you stop looking for a bigger server?

Why do we need a bigger server?

Execute queries faster.

Support more users.

Run multiple complex searches at once.

The fastest query is the one you never run

SQL intersection

7

What is cache?

A cache should transparently store data so future requests are served faster.

SQL intersection

8

Latency is why we need cache

Table 2.2 Example Time Scale of System Latencies

Event	Latency	Scaled
1 CPU cycle	0.3 ns	1 s
Level 1 cache access	0.9 ns	3 s
Level 2 cache access	2.8 ns	9 s
Level 3 cache access	12.9 ns	43 s
Main memory access (DRAM, from CPU)	120 ns	6 min
Solid-state disk I/O (flash memory)	50–150 µs	2–6 days
Rotational disk I/O	1–10 ms	1–12 months
Internet: San Francisco to New York	40 ms	4 years
Internet: San Francisco to United Kingdom	81 ms	8 years
Internet: San Francisco to Australia	183 ms	19 years
TCP packet retransmit	1–3 s	105–317 years
OS virtualization system reboot	4 s	423 years
SCSI command time-out	30 s	3 millennia
Hardware (HW) virtualization system reboot	40 s	4 millennia
Physical system reboot	5 m	32 millennia

Data from Systems Performance by Brendan Gregg

Examples of Cache

CPU L1 cache

CPU L2 cache

Main memory (RAM)

OS file system cache

SQL Server's buffer pool


These cache raw data.
SQL intersection
11
© SQLintersection. All rights reserved.
http://www.SQLintersection.com


These are out of our direct control.
SQL intersection
12
© SQLintersection. All rights reserved.
http://www.SQLintersection.com

These don't cache query results.

Why add another cache?

Control

- Cache duration
- Data format – data pages vs query results

Why add another cache?

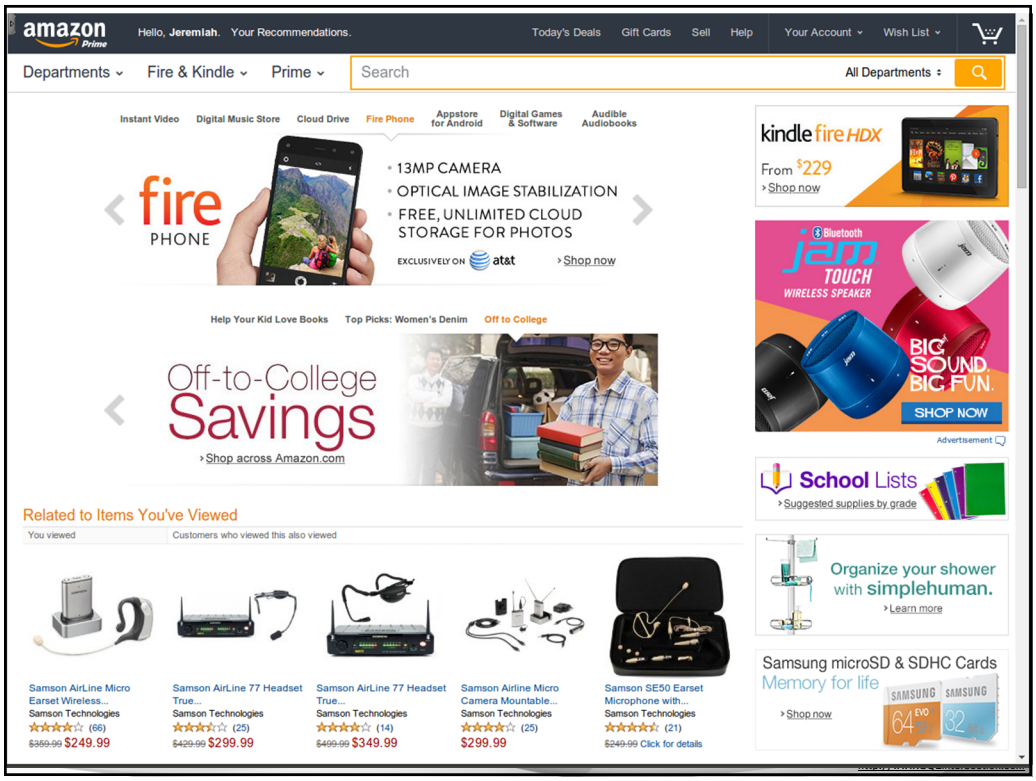
Control

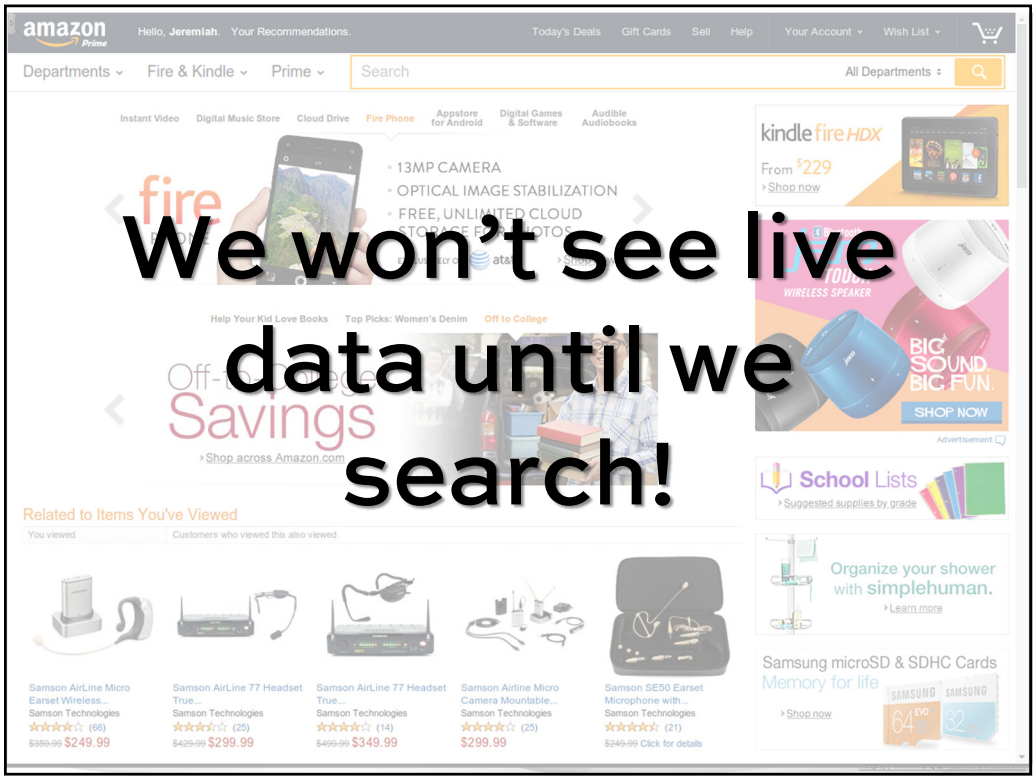
- Cache duration
- Data format – data pages vs query results

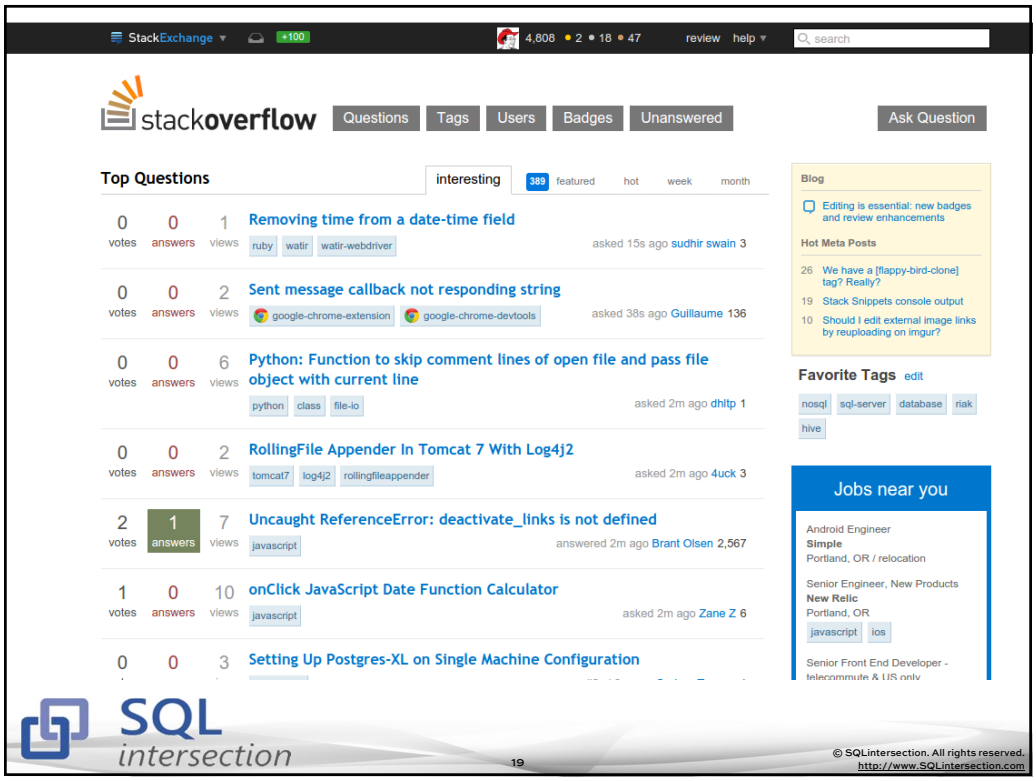
Scalability

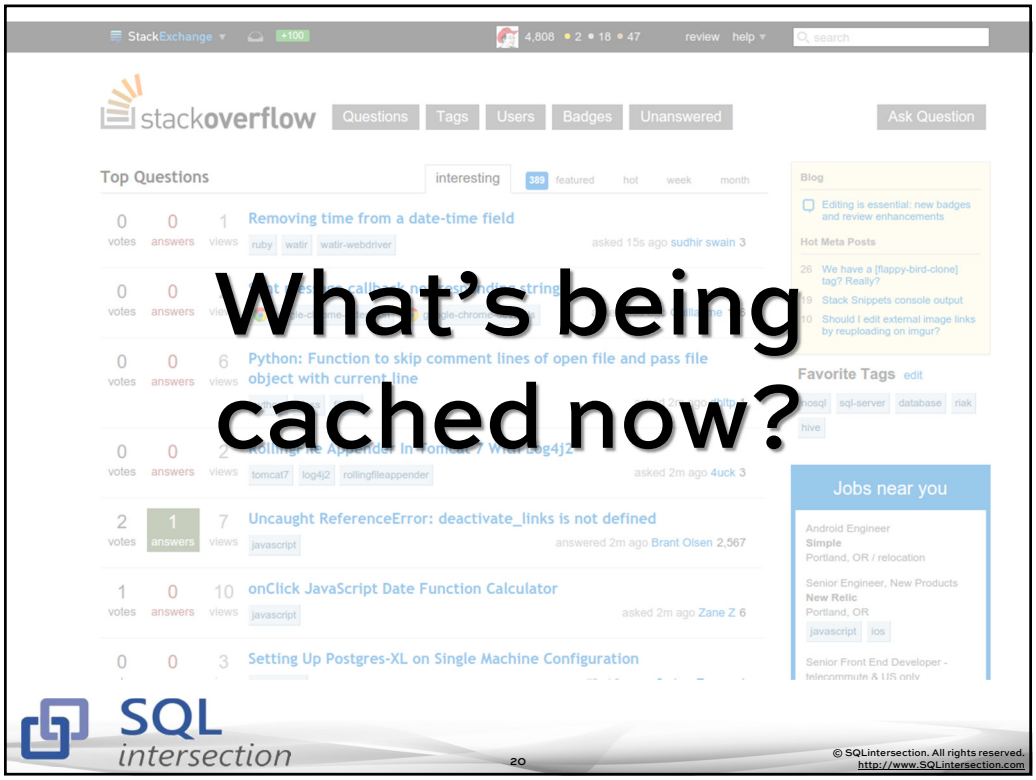
- Cache should scale separately

An example of caching

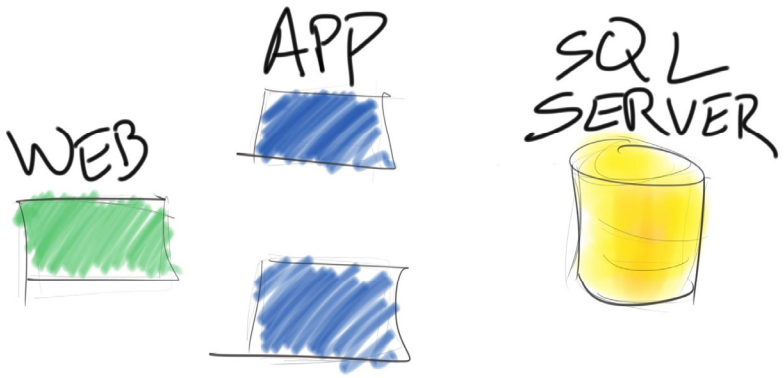

amazon
Hello, Jeremiah. Your Recommendations.
Today's Deals
Gift Cards
Sell
Help
Your Account
Wish List
Departments
Fire & Kindle
Prime
Search
All Departments
Instant Video
Digital Music Store
Cloud Drive
Fire Phone
Appstore for Android
Digital Games & Software
Audible Audiobooks
fire PHONE
13MP CAMERA
OPTICAL IMAGE STABILIZATION
FREE, UNLIMITED CLOUD STORAGE FOR PHOTOS
EXCLUSIVELY ON at&t
Shop now
kindle fire HDX
From $229
Shop now
Bluetooth
jam TOUCH
WIRELESS SPEAKER
BIG SOUND. BIG FUN.
SHOP NOW
Advertisement
Help Your Kid Love Books
Top Picks: Women's Denim
Off to College
Off-to-College Savings
Shop across Amazon.com
School Lists
Suggested supplies by grade
Related to Items You've Viewed
You viewed
Customers who viewed this also viewed
Organize your shower with simplehuman.
Learn more
Samson AirLine Micro Earset Wireless...
Samson Technologies
(66)
$359.99 $249.99
Samson AirLine 77 Headset True...
Samson Technologies
(25)
$429.99 $299.99
Samson AirLine 77 Headset True...
Samson Technologies
(14)
$499.99 $349.99
Samson Airline Micro Camera Mountable...
Samson Technologies
(25)
$299.99
Samson SE50 Earset Microphone with...
Samson Technologies
(21)
$249.99 Click for details
Samsung microSD & SDHC Cards
Memory for life
Shop now


We won't see live data until we search!


StackExchange
stackoverflow
Questions
Tags
Users
Badges
Unanswered
Ask Question
Top Questions
interesting
Removing time from a date-time field
Sent message callback not responding string
Python: Function to skip comment lines of open file and pass file object with current line
RollingFile Appender In Tomcat 7 With Log4j2
Uncaught ReferenceError: deactivate_links is not defined
onClick JavaScript Date Function Calculator
Setting Up Postgres-XL on Single Machine Configuration
Blog
Editing is essential: new badges and review enhancements
Hot Meta Posts
Favorite Tags
Jobs near you
SQL intersection
19
© SQLintersection. All rights reserved.
http://www.SQLintersection.com


What's being cached now?
SQL intersection
20
© SQLintersection. All rights reserved.
http://www.SQLintersection.com

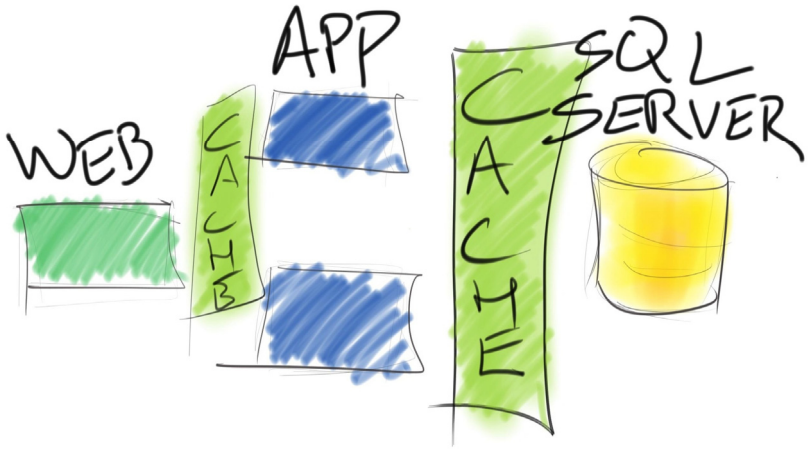
Where should you cache?



SQL intersection

21

Cache Everywhere



SQL intersection

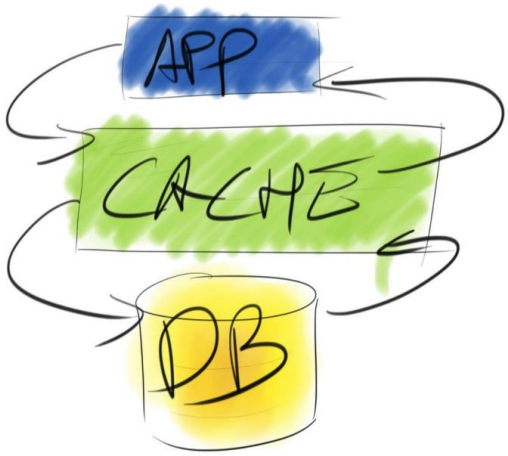
22

There are two forms of cache

Write Through Cache

Write Aside Cache

Write Through Cache



Write Through Cache: Benefits

Cache coherence

Fast updates

Simple code

Write Through Cache: Drawbacks

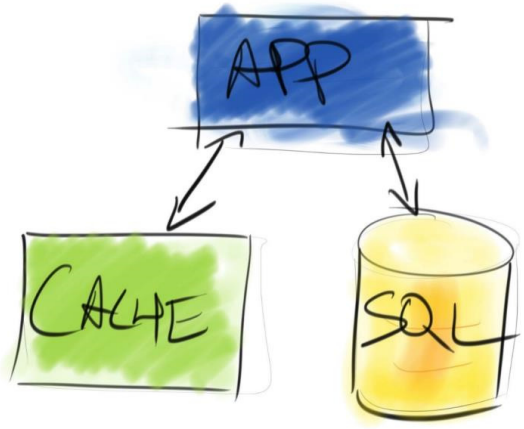
Cache now needs fault tolerance

Strict dependency on language or database

Cache coherence is a myth

May not have complete control over what is cached

Cache Aside



SQL intersection

Cache Aside: Benefits

Application retains full control

Result sets can be easily cached

Intermediate work can be stored in cache

Easy to get started

SQL intersection

Cache Aside: Drawbacks

We're responsible for the hard work

- Cache invalidation
- Determining what to cache

Changes are required to app code

Problems with Both Methods

Cache consistency

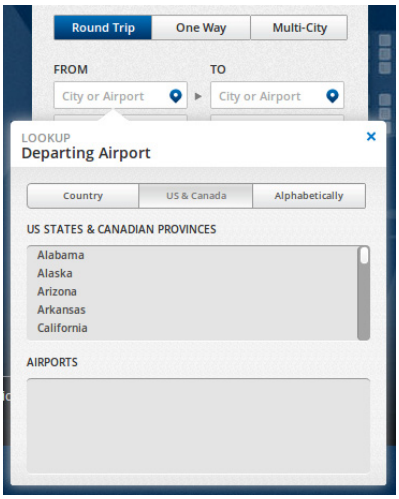
Cache invalidation

Performance during cache start up

Either Option Reduces Work Done

What Should You Cache?

- Dropdown lists
- Commonly queried data
- Paged reports
- Work in progress



Getting Started with Caching

Identify frequently run queries

Categorize these by:

- Risk
- Frequency
- Frequency of Updates
- Data Longevity

Finding Queries to Cache

SQL Server is already tracking this!

The plan cache can get cleared by many things:
Not all queries will be present.

http://www.brentozar.com/go/plans/

Which Queries Should We Cache?

# Execution	Avg CPU	% CPU	Avg Duration	% Duration	Avg Reads	Exec / Min
1	19,055,745,180	32	20,733,815	24	750,443,342	1
1	12,837,730,475	22	7,316,989	8	4,467,732,165	1
201,152,295	22	7	0	5	2	193,190,291
1	3,142,895,504	5	3,266,072	4	2,456,252	1
91,556,589	18	3	0	2	2	88,859,883
80,806,339	19	3	0	2	2	79,444,929
1	1,099,366,224	2	2,153,089	2	167,224,435	1
1	1,014,948,232	2	2,406,363	3	418,868,695	1
1	918,130,859	2	985,988	1	566,702,305	1

SQL intersection

Which Queries Should We Cache?

# Execution	Avg CPU	% CPU	Avg Duration	% Duration	Avg Reads	Exec / Min
1	19,055,745,180	32	20,733,815	24	750,443,342	1
1	12,837,730,475	22	7,316,989	8	4,467,732,165	1
201,152,295	22	7	0	5	2	193,190,291
1	3,142,895,504	5	3,266,072	4	2,456,252	1
91,556,589	18	3	0	2	2	88,859,883
80,806,339	19	3	0	2	2	79,444,929
1	1,099,366,224	2	2,153,089	2	167,224,435	1
1	1,014,948,232	2	2,406,363	3	418,868,695	1
1	918,130,859	2	985,988	1	566,702,305	1

SQL intersection

Why?

High executions/minute

- Probably a common lookup – e.g. list of states
- Even parameterized queries can be cached

High reads/CPU

- Ability to cache depends on use.
- Requires further investigation and business discussion.

Disk as a Cache

Remember our definition of cache?

A cache should **transparently store data** so future requests are **served faster**.

SQL Server's Disk as a Cache

Indexed Views

Filtered Indexes

Indexed Views

Very strict implementation rules.

Materialize a view to disk.

Best used when:

- Query patterns are well known.
- Aggregations are common.
- Read heavy workload.
- Storage is abundant.

Filtered Indexes

Move the predicate to the index definition.

Indexes are smaller and more selective.

Strict implementation rules.

Best used with:

- Common searches (e.g. IsActive = 1).
- Careful planning.

References

Picking Your Cache
http://www.brentozar.com/archive/2012/05/picking-your-cache/

The Fastest Query is the One You Never Make
http://www.brentozar.com/archive/2013/02/the-fastest-query-is-the-one-you-never-make/

Caching in the Distributed Environment
http://msdn.microsoft.com/en-us/library/dd129907.aspx

Working with 154mm Records in Azure Table Storage
www.troyhunt.com/2013/12/working-with-154-million-records-on.html

Creating a Flat File database with Amazon S3
http://seroter.wordpress.com/2013/05/06/creating-a-flat-file-shared-database-with-amazon-s3-and-node-js/

SQL intersection

This time it'll be a little different

1. You'll pick one of two different design problems
2. You may work alone *or* team up with others
   - Work on the problem for 20 minutes
   - Diagram out architecture changes you recommend
3. We'll meet back and go over different solutions

Notes:

- There may not be a perfect solution given every restriction listed
- You do NOT have to design the hardware (no core counts/ memory amounts)

SQL intersection

2

Choose your problem

Make your choice now!

Problem 1: A thousand databases (and counting)

Problem 2: Designing AlwaysOn Availability Groups to stay "On"

Three components to each solution

People

Process

Technology

Problem 1: 1,000 databases (and counting)

The application

Application has been prototyped already and is in beta use

- It can be changed, but that's a ton of work and re-coding

The design:

- One database per client
- There will be many client databases: estimated 1,000 in the first year
- Expect to increase by 3x in the first year thereafter

Each client database will be small (< 200MB each)

One centralized database, "Metadata" that holds information that all the other databases will access

- Client databases will all have queries that join to the Metadata database
- Metadata will also be small (100 MB)

The team

Invested heavily in virtualization

- Everything in the current environment is virtualized
- Have multiple SANs hosting a variety of applications already

No experience with Windows Failover Clustering or Availability Groups

Team consists of:

- 10 developers
- 1 “accidental” DBA (who is one of the 10 developers)
- 2 system engineers who work on storage, virtualization, Active Directory, networking, printers, phones, and get stuck cleaning people’s keyboards

1,000 databases - RPO and RTO

The business has signed off on these numbers for the initial architecture

The numbers apply both to the client databases and the “metadata” database

	Server Offline	Corrupt data	“Oops” deletes	Datacenter offline
RPO: Minutes of data loss allowed	10 minutes	10 minutes	10 minutes	4 hours (to be implemented later)
RTO: Minutes of downtime allowed	5 minutes	4 hours*	4 hour*	1 week (to be implemented later)

** If data is incorrect or corrupt, would rather have the system offline.*

Licensing and requirements

Can have an hour of planned downtime each month for maintenance

- The RTO of 5 minutes is for unplanned downtime

Want to minimize licensing costs

- Prefer Standard Edition strongly, especially if they need multiple instances

No current DR site

- They don't expect you to solve that problem

Your mission

1. What technologies do you use to achieve the RPO and RTO (you don't have to solve DR)?

1. What are the benefits to your approach?

1. What are the risks?

Problem 2: Keeping AlwaysOn Availability Groups "On"

The business problem

Torosaurus Tech has purchased Agilisaurus, Inc.*

- All databases will be migrated to new servers and storage in the Torosaurus datacenter

Torosaurus is ready to invest

- Expects to increase the number of customers by 50% in the next year
- Grow the Agilisaurus team as needed to support the environment

There's no huge rush on the new environment. They just want to do it right.

** These are fake companies, but their story isn't unusual.*

Agilisaurus application

Popular website

- 200GB database
- 20GB tempdb currently
- Inactive customers are archived
- Slow to moderate data growth

Current hardware

- Physical server with 256GB of memory and 8 logical cores (4 physical). No HA or DR.
- Performance is very high and consistent
- Team has invested in application caching, index tuning, query tuning

Current teams

Torosaurus Tech – new parent company

- 4 system engineers (SAN, Networking, Active Directory)
- 5 busy database administrators who manage a mix of SQL Server and Oracle databases (Mostly SQL Server)
- 40 developers working on a variety of products

Agilisaurus, Inc – the purchased company

- 5 developers

Torosaurus is planning to

- Hire a Senior SQL Server DBA
- Have them specialize in Availability Groups and train the rest of the team to work in an on-call rotation

Torosaurus - RPO and RTO

The business has signed off on these numbers for the initial architecture

	Server Offline	Corrupt data	"Oops" deletes	Datacenter offline
RPO: Minutes of data loss allowed	0 minutes	0 minutes	0 minutes	0 minutes
RTO: Minutes of downtime allowed	1 minute	1 minute	1 minute	1 hour

Torosaurus' infrastructure

Both virtualized and bare metal
- Most production databases are not virtualized
- Dev environments are all virtualized

Want to use the best technology for each purpose
- Use SAN technology heavily
- Open to using commodity SSDs in local storage when it makes sense for price and speed

Primary Datacenter: 20 minutes from the main office
- Hire datacenter support for hands-on support
- Sysadmins go to the datacenter regularly and rack, cable their own machines

DR Datacenter: 60 miles away

Licensing and requirements

Want to avoid unplanned downtime

- Customers around the world, 24 x 7 use
- Published goal is 99.9% uptime (43 minutes per month)
- Internal process is report when downtime exceeds 5 minutes each month for “gold” level applications

Backups and failovers:

- Want to take backups in both production and DR site
- DR servers should be as powerful as production, but don’t need automatic failover WITHIN the DR site
- Want automatic protection from corruption in the primary site

SQL intersection

17

1. Design the solution:

- Number of SQL Server instances /servers: _____
- Virtualized or Bare Metal? _____
- Is there a Windows Failover Cluster Instance? _____
- Is there an AlwaysOn Availability Group?
  - Number synchronous replicas: _____
  - Number asynchronous replicas: _____
- How many Enterprise licenses are needed with your design? _____

SQL intersection

18

Your mission (continued)

2. What processes will be key to making this a success?

3. What are the risks?

Possible solutions

Not every problem has a perfect solution. Often, you have to change more than just the SQL Server.

SQL intersection

35

Surprises?
Objections?

SQL intersection

36

After a day of architecture… what's most exciting?

SQL intersection

Questions?

BRENT OZAR UNLIMITED®

Don't forget to complete an online evaluation on EventBoard!

Session: PRECON06

SQL Server Architecture:

Choosing Virtualization, Clustering, AlwaysOn, and More

Your evaluation helps organizers build better conferences and helps speakers improve their sessions.

Thank you!

SQL intersection