Building Applications for Azure: Lessons from Scale

Mark Russinovich
CTO, Azure

Microsoft

Stuff Everyone Knows About Cloud Development

- Automate
- Scale out
- Test in production
- Deploy early, deploy often

But there are many more rules…

Stories From the Trenches

- All of these cases are real
  - Customer cases from Azure Customer Advisory Team (CAT) engagements
  - Azure cases caused outages in test or production
- The names of the customers have been omitted to protect the guilty

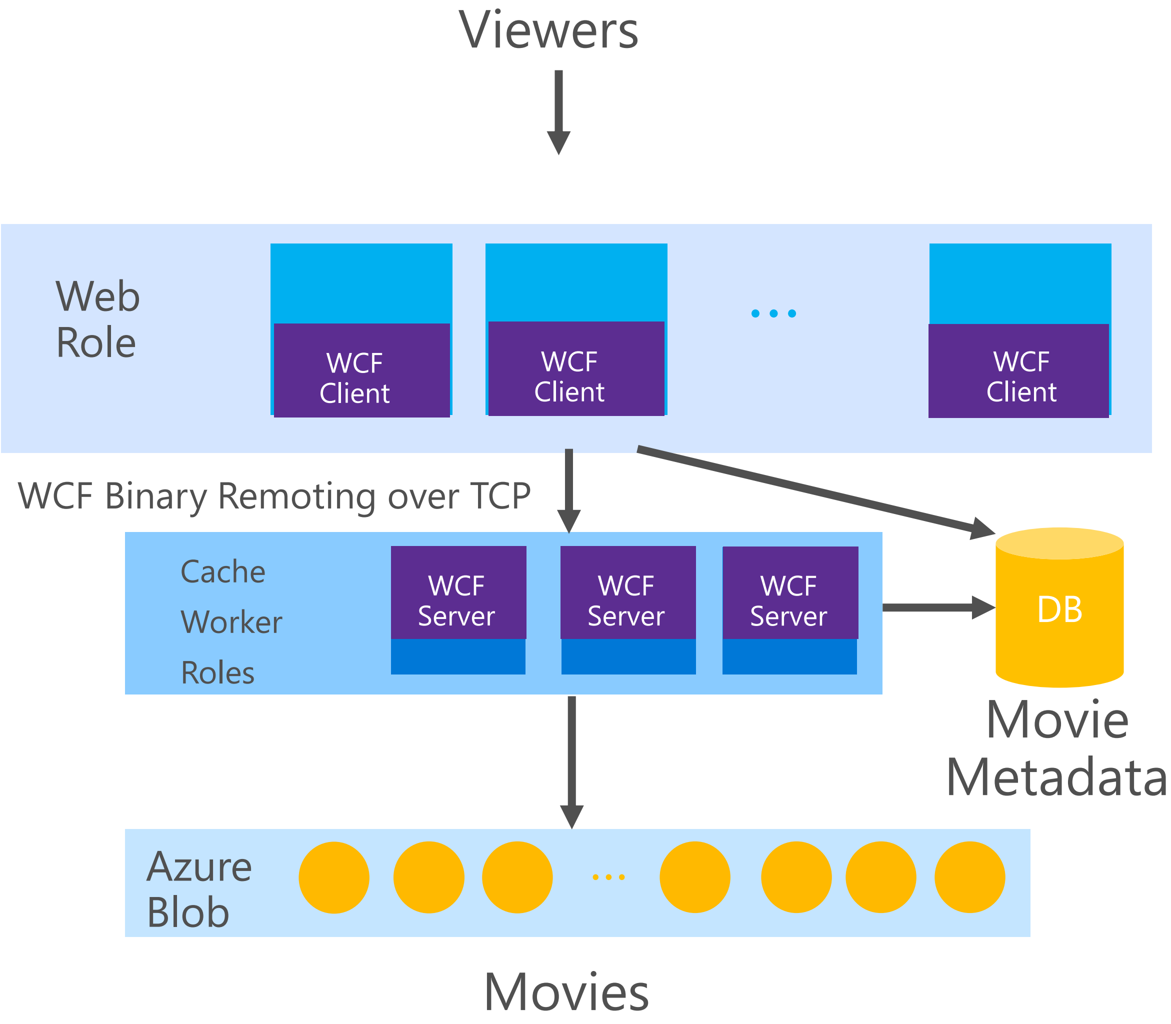
Customer Lessons

Movie Streaming

“Now showing on Azure”

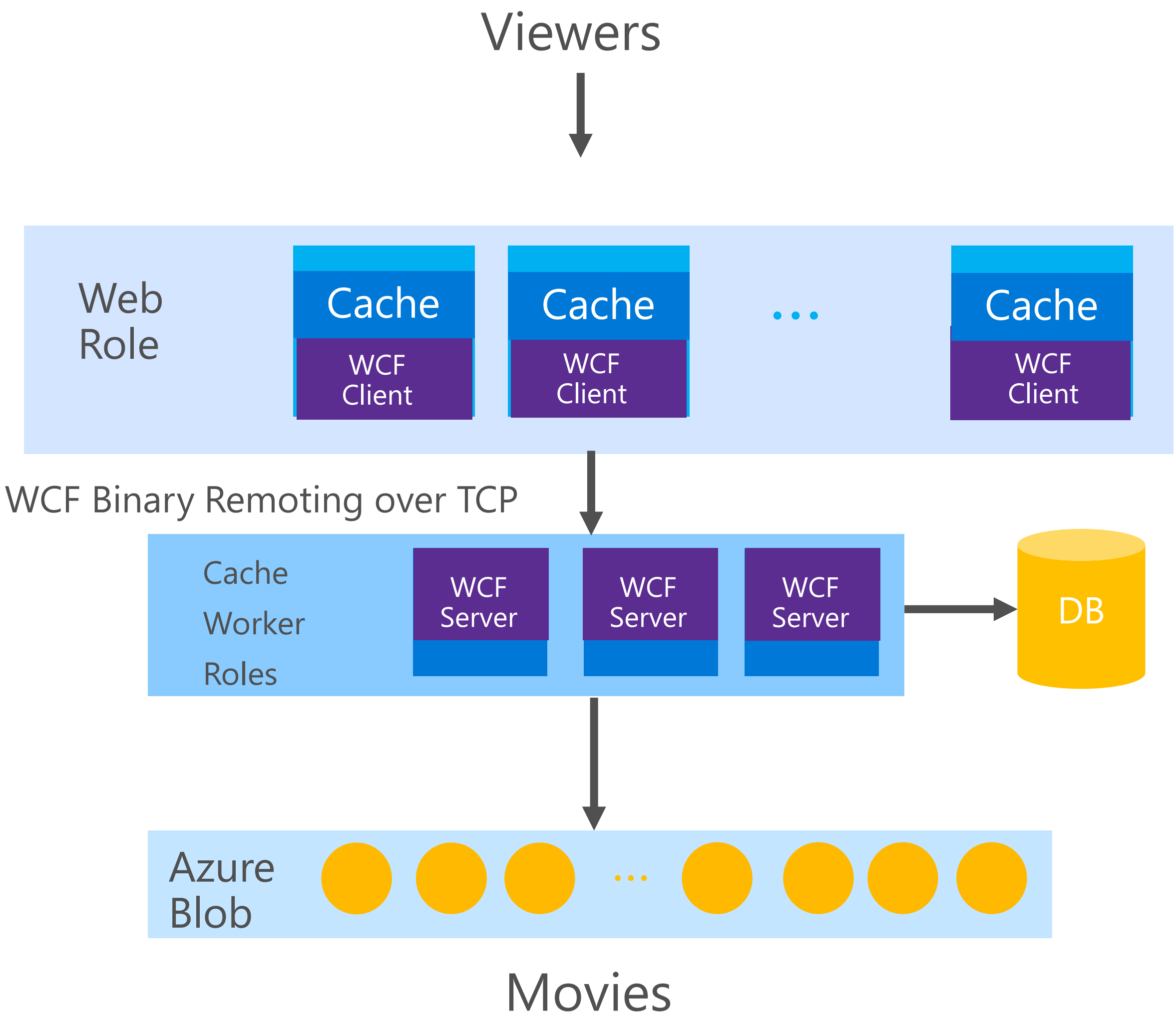
Cache Me If You Can

- Startup (now large on-demand movie streaming company) started with pure PaaS streaming service
- Built custom caching tier Worker Role
  - Caches movie metadata
  - If remote cache query > 2s, query database directly



An Extra Cache Goes A Long Way

- Problem: if cache role rebooted or updated, Web Role would overwhelm database
- Solution: add a local cache to the Web Role



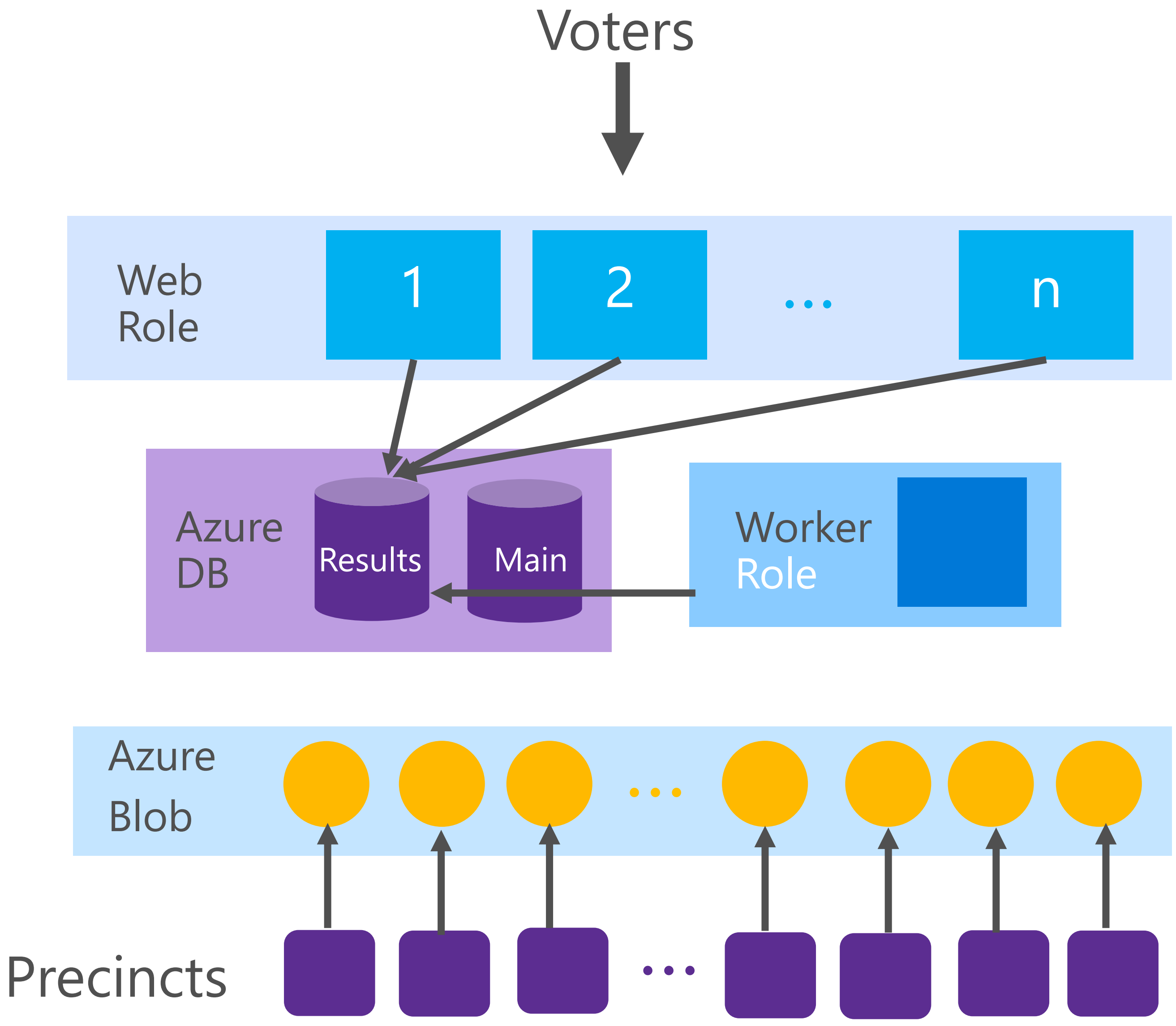
Election Tracking

“Vote early, vote often”

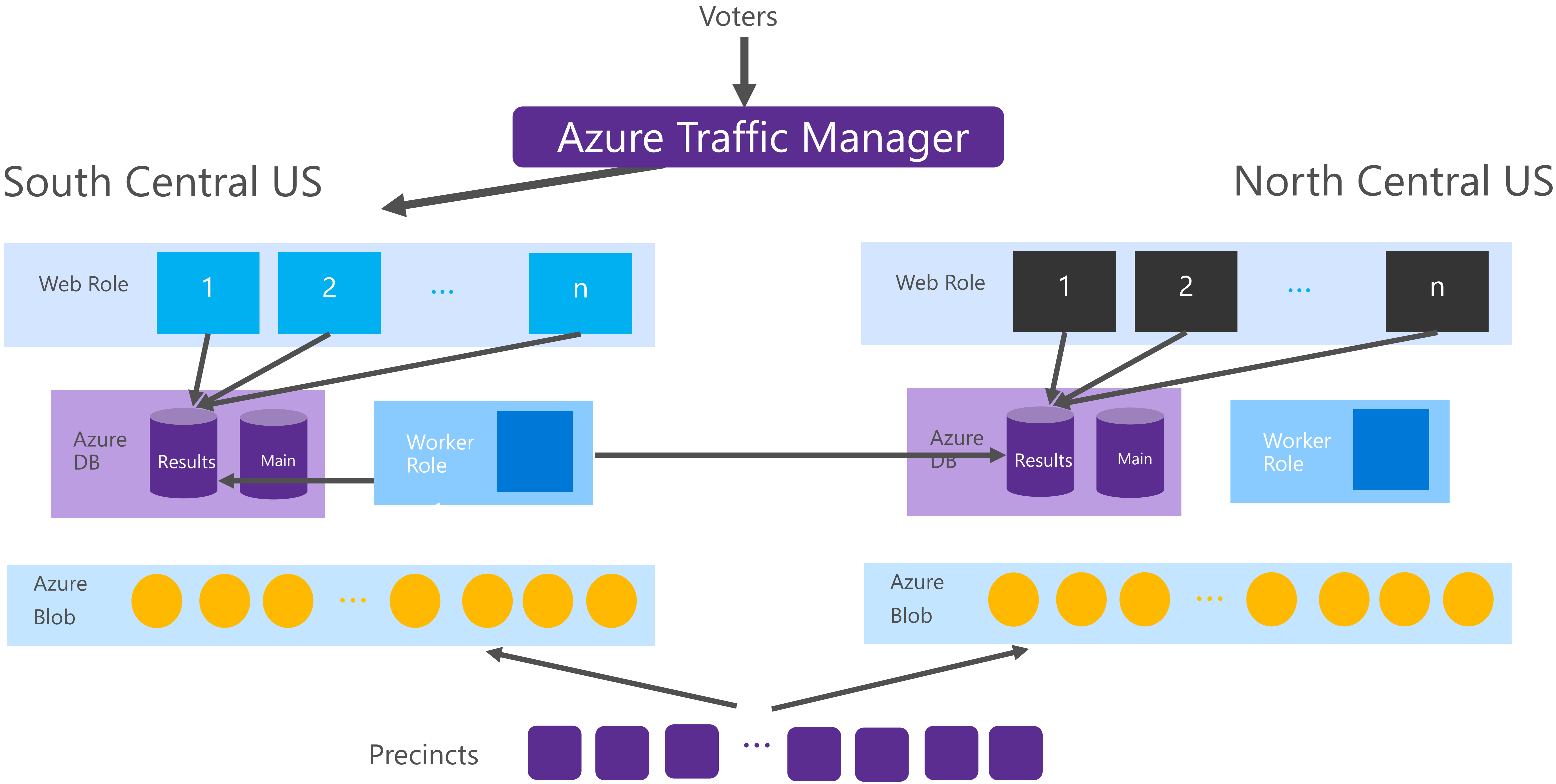
Who’s Winning?

- Customer created service for reporting live tally of US Presidential, State and local elections
- Served a major state’s September election results successfully
- November election was coming – was the architecture going to handle the load?

Election Results Architecture



Disaster-Proof Deployment



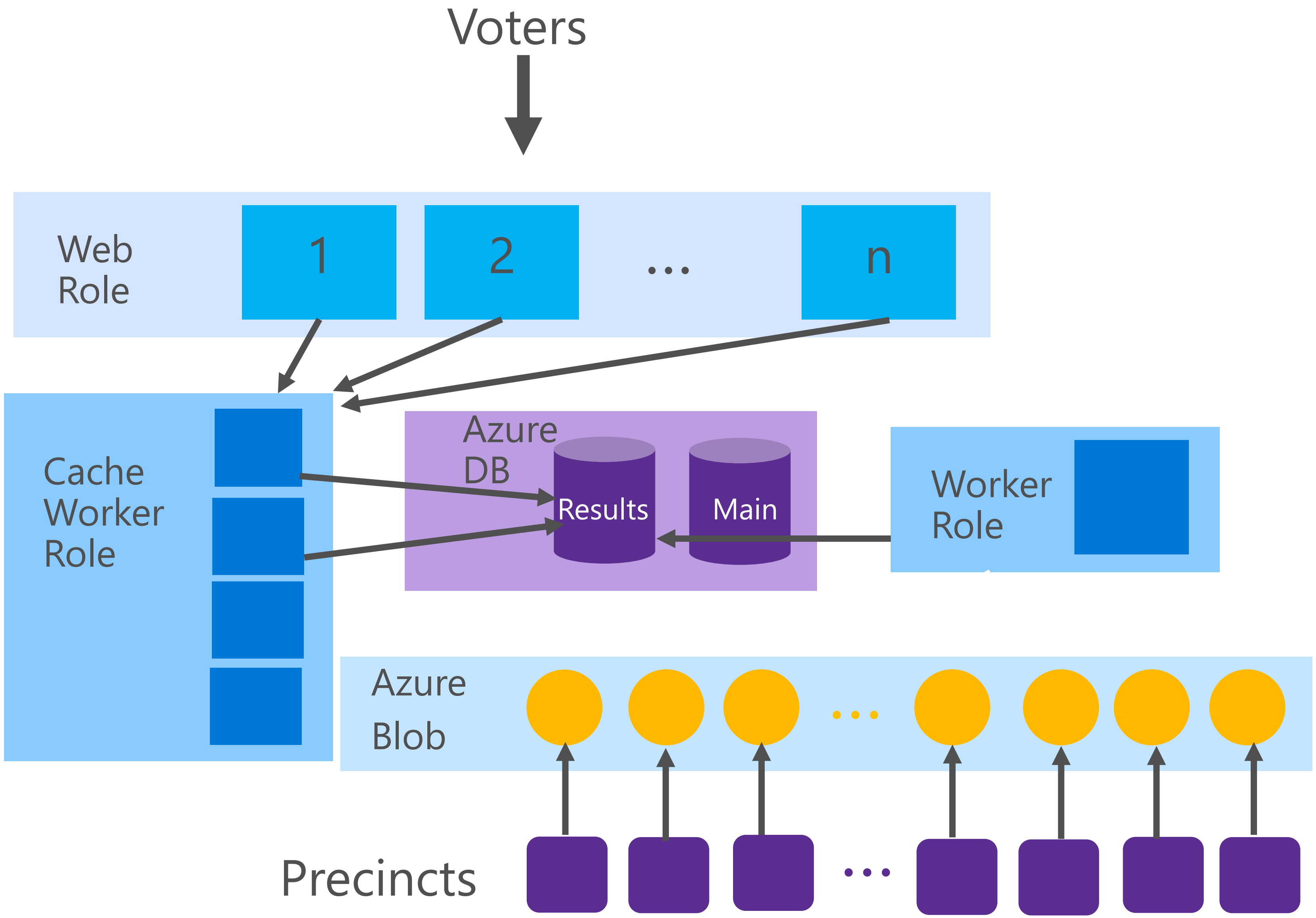
Let Me Check That For You

- Each web request results in about 10 SQL queries
- Load estimate for November election:

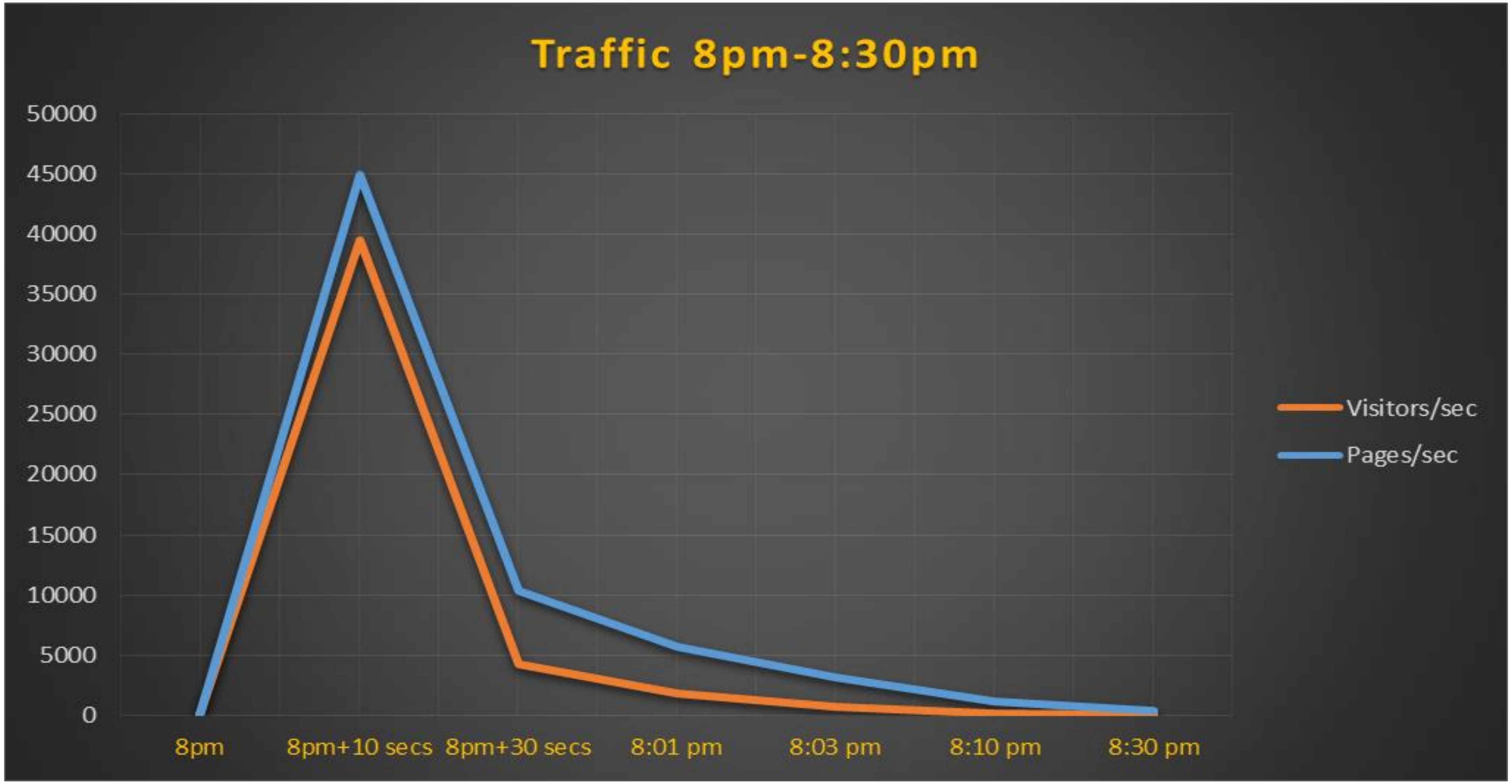
Expected Load				
Scenarios	Expected Page Views	Time Window (hrs)	Page View/sec	10X/pvs DB Calls/sec
Average	10,000,000	4	694	6,944
Peak Hour	6,000,000	1	1,667	16,667

- Problem is that Azure DB scales to 5000 connections, 180 concurrent requests, 1000 requests per second
- Solution: put a cache between the front-end and database with 40,000 requests/s per instance

Election Results Architecture



How'd I Do?


Traffic 8pm-8:30pm
50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
8pm
8pm+10 secs
8pm+30 secs
8:01 pm
8:03 pm
8:10 pm
8:30 pm
Visitors/sec
Pages/sec

Whew, That Was a Good Call!

- How the application would have performed without cache:

Actual: if direct SQL DB Calls (~10X)					
Time	Actual Page Views	Time Window (sec)	Page View/sec	Possible 10X DB Calls/sec	Est Capacity DIFF DB Calls/sec
8pm+10 secs	448,932	10	44,893	448,932	(447,932)
8pm+30 secs	206,925	20	10,346	103,463	(102,463)
8:01 pm	171,231	30	5,708	57,077	(56,077)
8:03 pm	378,350	120	3,153	31,529	(30,529)
8:10 pm	494,423	420	1,177	11,772	(10,772)
8:30 pm	416,379	1200	347	3,470	(2,470)

- With cache:

Actual: using Azure Cache (10X)					
Time	Actual Page Views	Time Window (sec)	Page View/sec	Actual 10X Cache Calls/sec	Est Capacity DIFF Cache Calls/sec
8pm+10 secs	448,932	10	44,893	448,932	(288,932)
8pm+30 secs	206,925	20	10,346	103,463	56,538
8:01 pm	171,231	30	5,708	57,077	102,923
8:03 pm	378,350	120	3,153	31,529	128,471
8:10 pm	494,423	420	1,177	11,772	148,228
8:30 pm	416,379	1200	347	3,470	156,530

Demo: Caching

Migration to PaaS

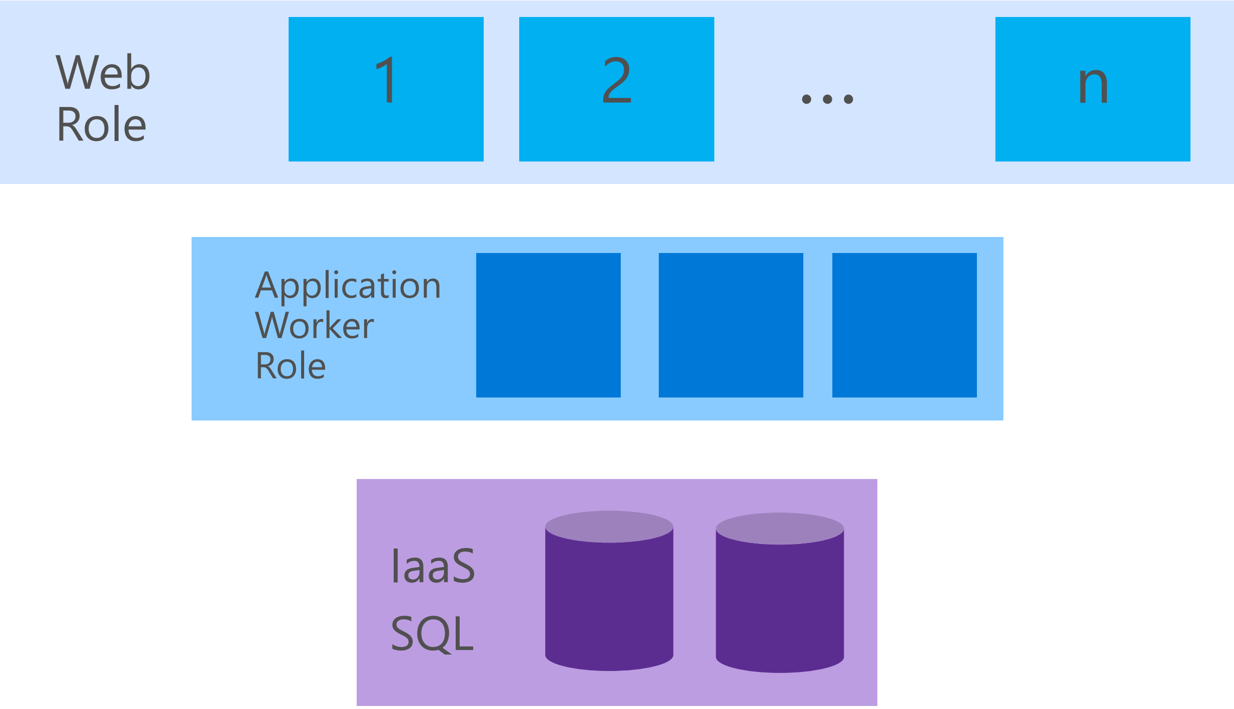
“I’m tired of updating your OS”

To The Cloud!

- Customer: software development company that produces solutions for the design, construction and operation of building, plant, civil, and geospatial infrastructure
- Goal: move existing multi-tier application to the cloud
  - .NET, SQL
  - Leverage PaaS where possible
  - Match performance of existing deployment

Let's Start Here

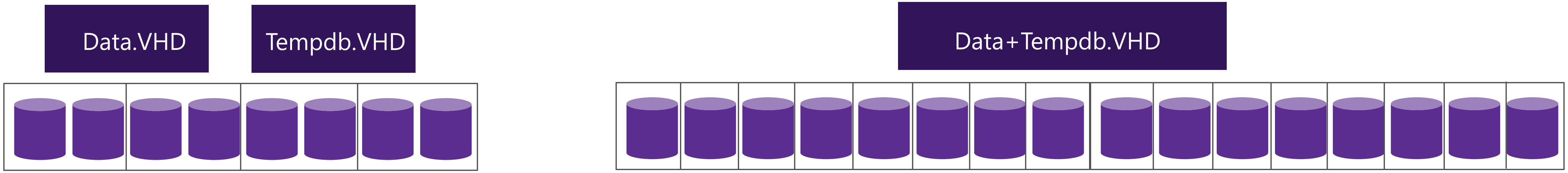
- PaaS design:
  - ASP.NET moved to Web Role
  - Application logic moved to Worker Role
  - SQL moved to IaaS SQL mirror
- Scale testing results were…disappointing:



	200 User Test	
	On-Prem App	Azure
Total Counts For Test 1 (batch):	33,174	10,135
Total Counts For Test 2 (interactive):	114,497	39,341

Let's Try This Instead…

- On-prem architecture had combined web/app server
  - Inter-role communication introduced extra cross-server communication
  - Optimization: collapse Azure Web and Worker roles
- Storage not optimized for IaaS SQL usage



Initial Configuration

Data.VHD

Tempdb.VHD

1844 IOPS
15.26 MB/sec

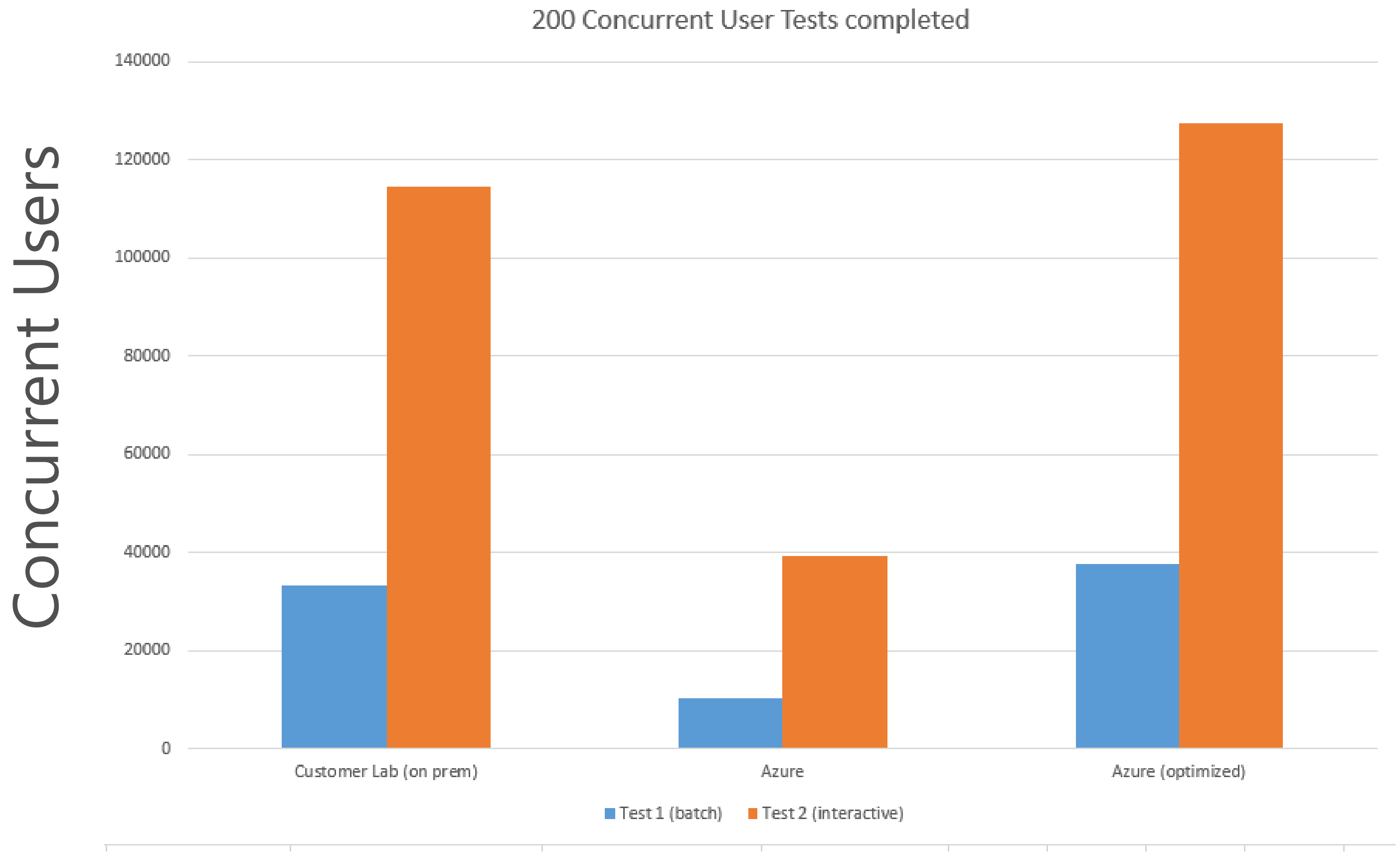
Optimized Configuration

Data+Tempdb.VHD

5379 IOPs
336.15 MB/sec

That Did the Trick

- Performance tests of optimized configuration showed great performance:

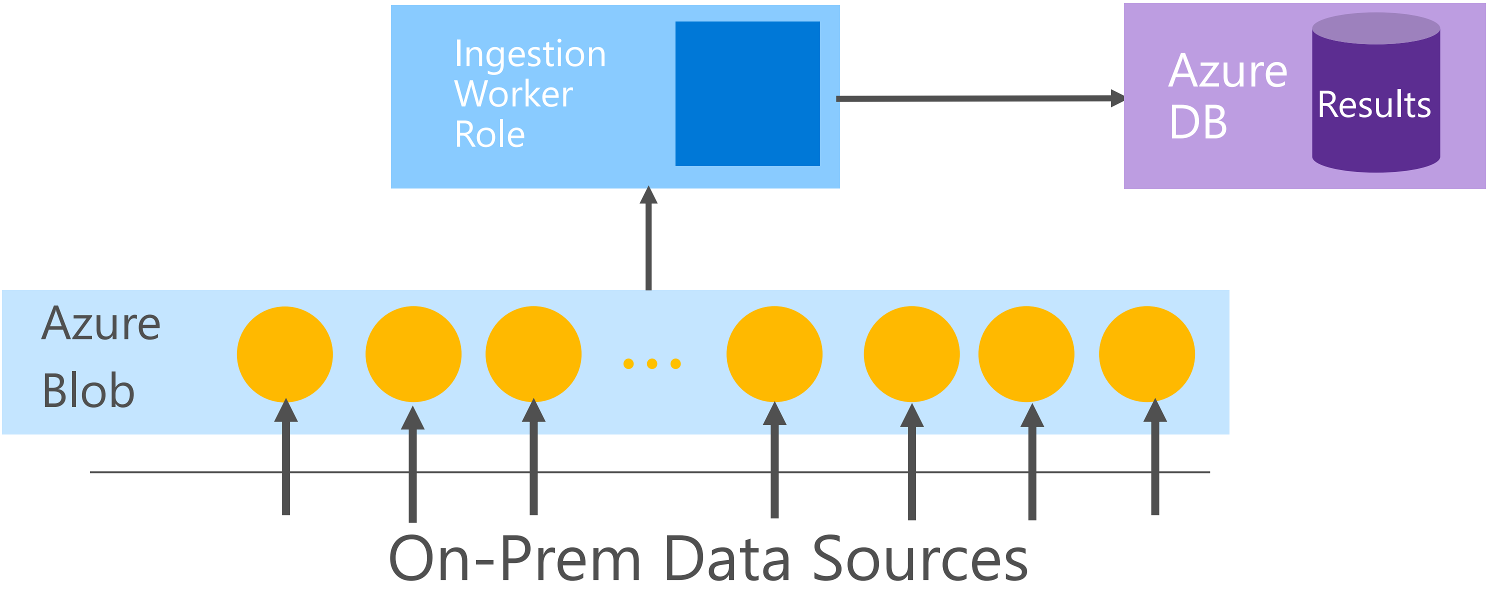


Data Upload to Azure DB

“Chew on this…”

What Does the Data Say?

- Leading software company for advertising monetization created hybrid PaaS Azure solution
- Architecture includes:
  - Daily activity and transaction history .csv file upload to Azure storage
  - Import to Azure DB
  - Trailing 7-day aggregate view for analytics and trending with HDInsight and prediction with CloudML

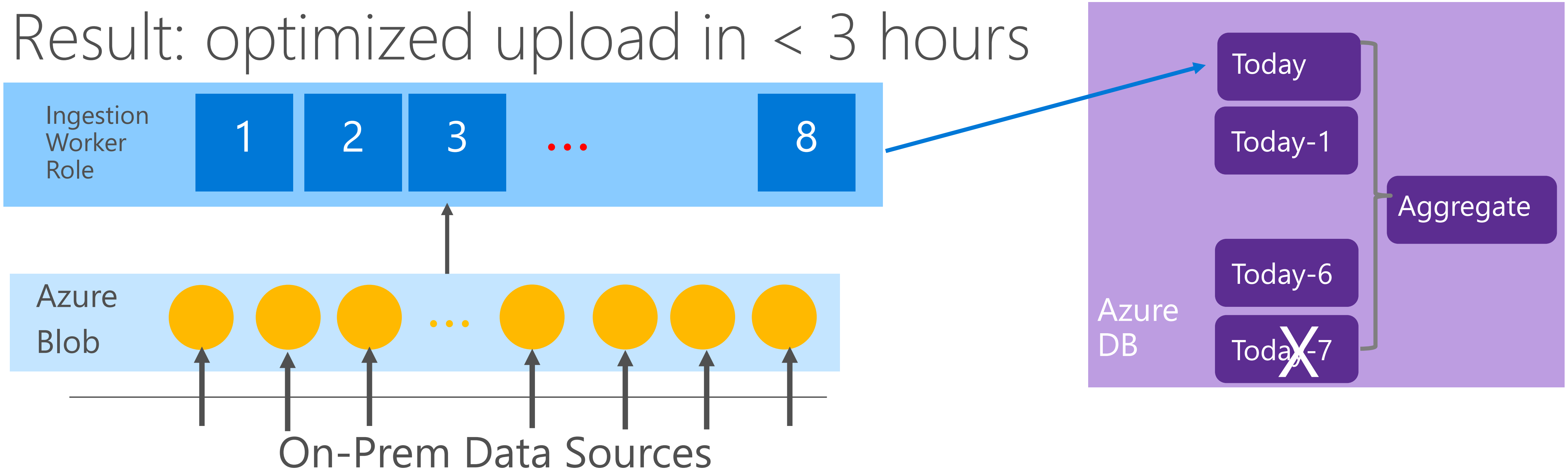


Any Day Now…

- Source data was over 100 CSV files between 10MB and 1.4GB
  - Average total ingest was around 40 GB
  - Customer wrote custom ETL process using SqlBulkCopy
- Problem: ingest took around 37 hours
- Realization: Azure DB is a scale-out architecture

Keeping Up in Three Parts

- 1: Move to Azure DB Premium
- 2: Parallelize data upload by scaling the worker role to eight streams
- 3: Create one table/day, view aggregates week of data
- Result: optimized upload in < 3 hours

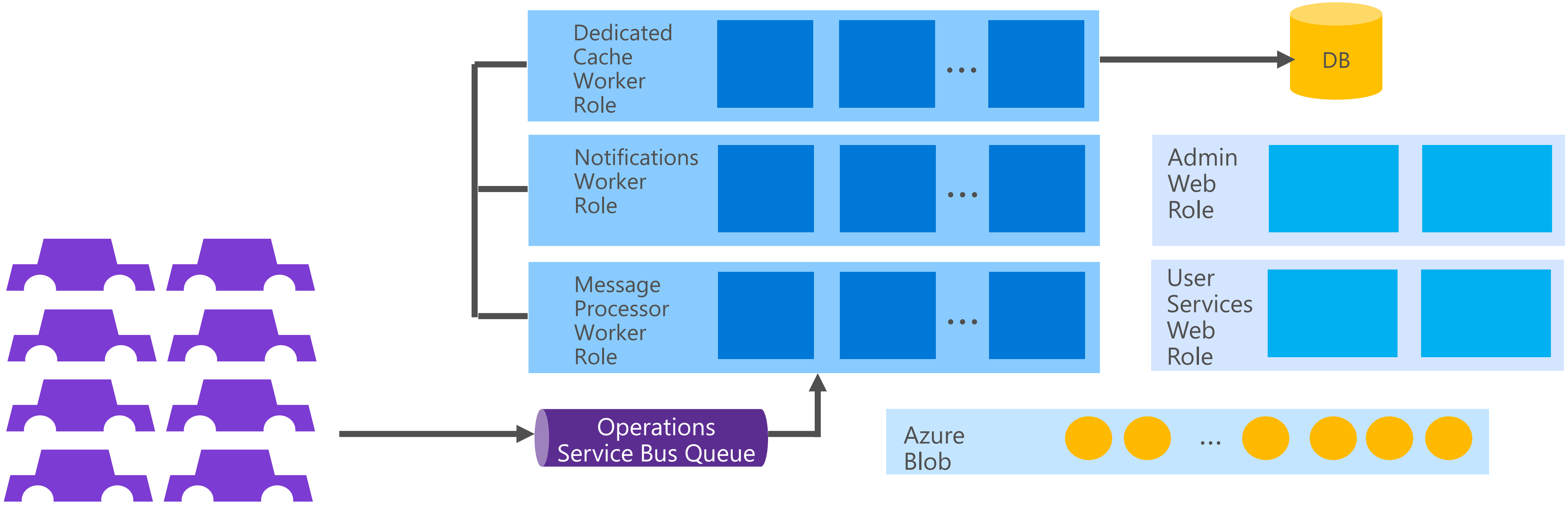


Connected Cars

“Calling all cars”

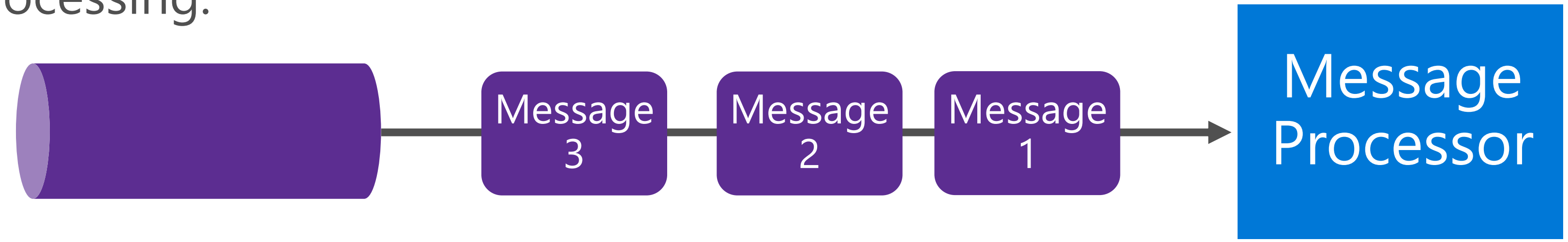
Connecting Azure with the World

- Large connected car services company created new service on Azure
- Goal: leverage Azure PaaS services

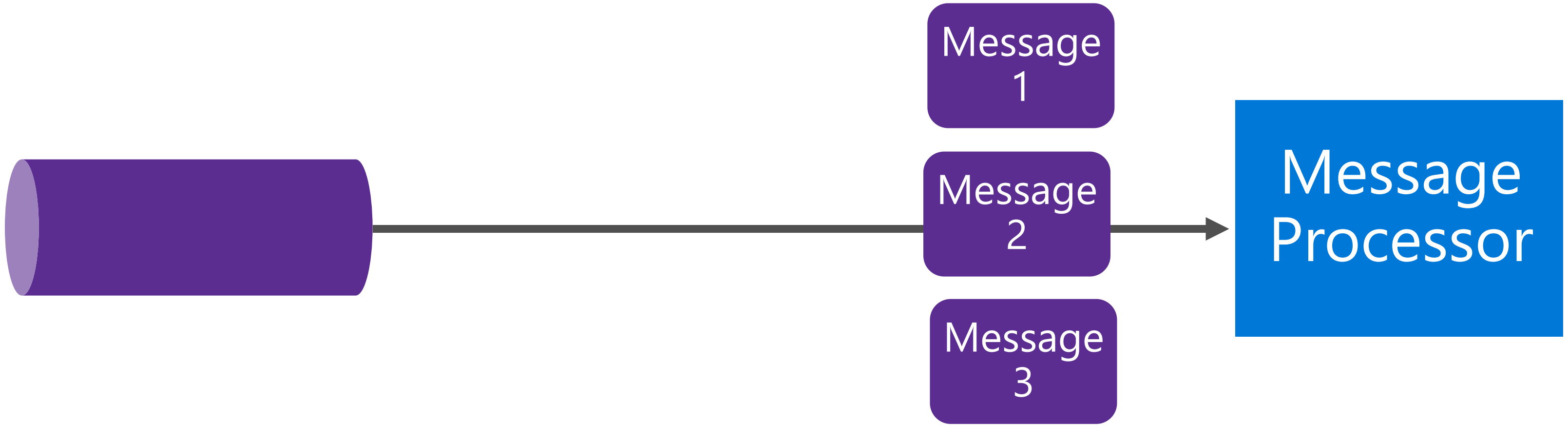

Dedicated Cache Worker Role
…
DB
Notifications Worker Role
…
Admin Web Role
Message Processor Worker Role
…
User Services Web Role
Operations Service Bus Queue
Azure Blob
…

One By One

- Performance measurements showed message process far less than 10,000/s required
- Problem 1:
  - Synchronous message processing:

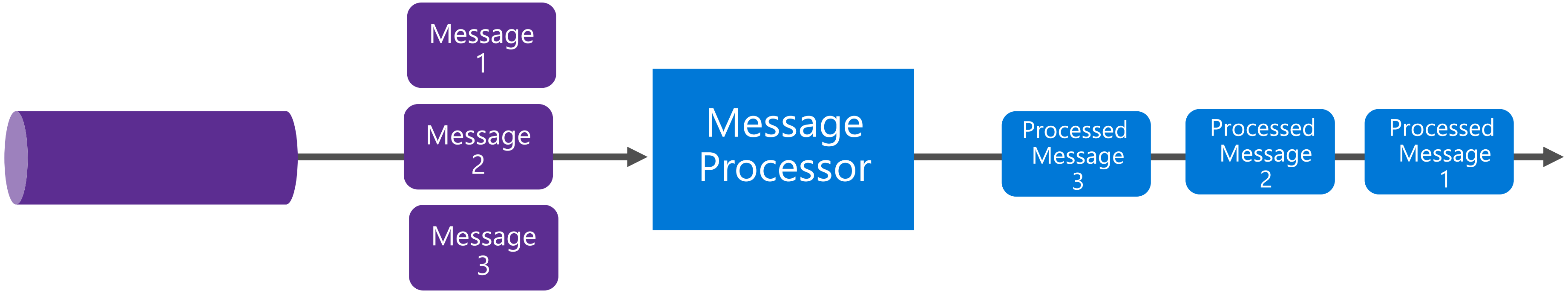


  - Fix: asynchronous (batch) receive:

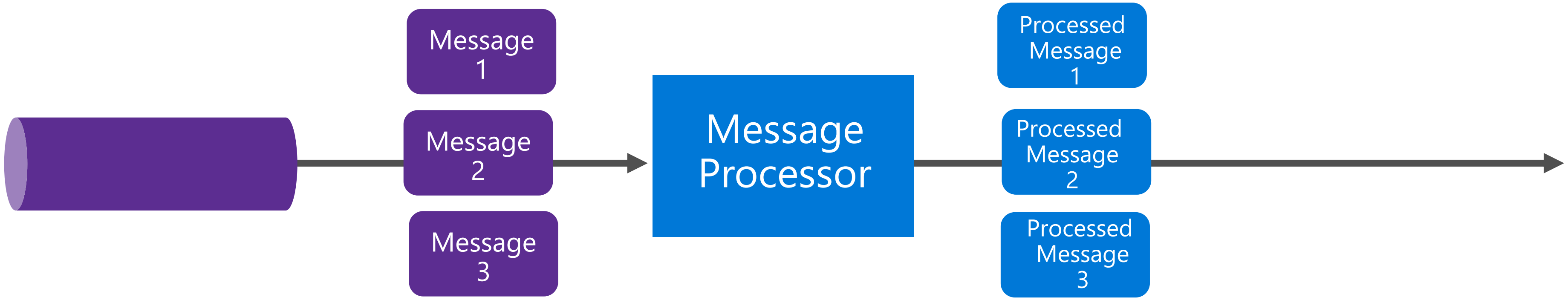


Remove One Bottleneck, Find Another

- Problem 2:
  - Processing one by one:



  - Fix: concurrent processing



http://msdn.microsoft.com/en-us/library/azure/hh528527.aspx

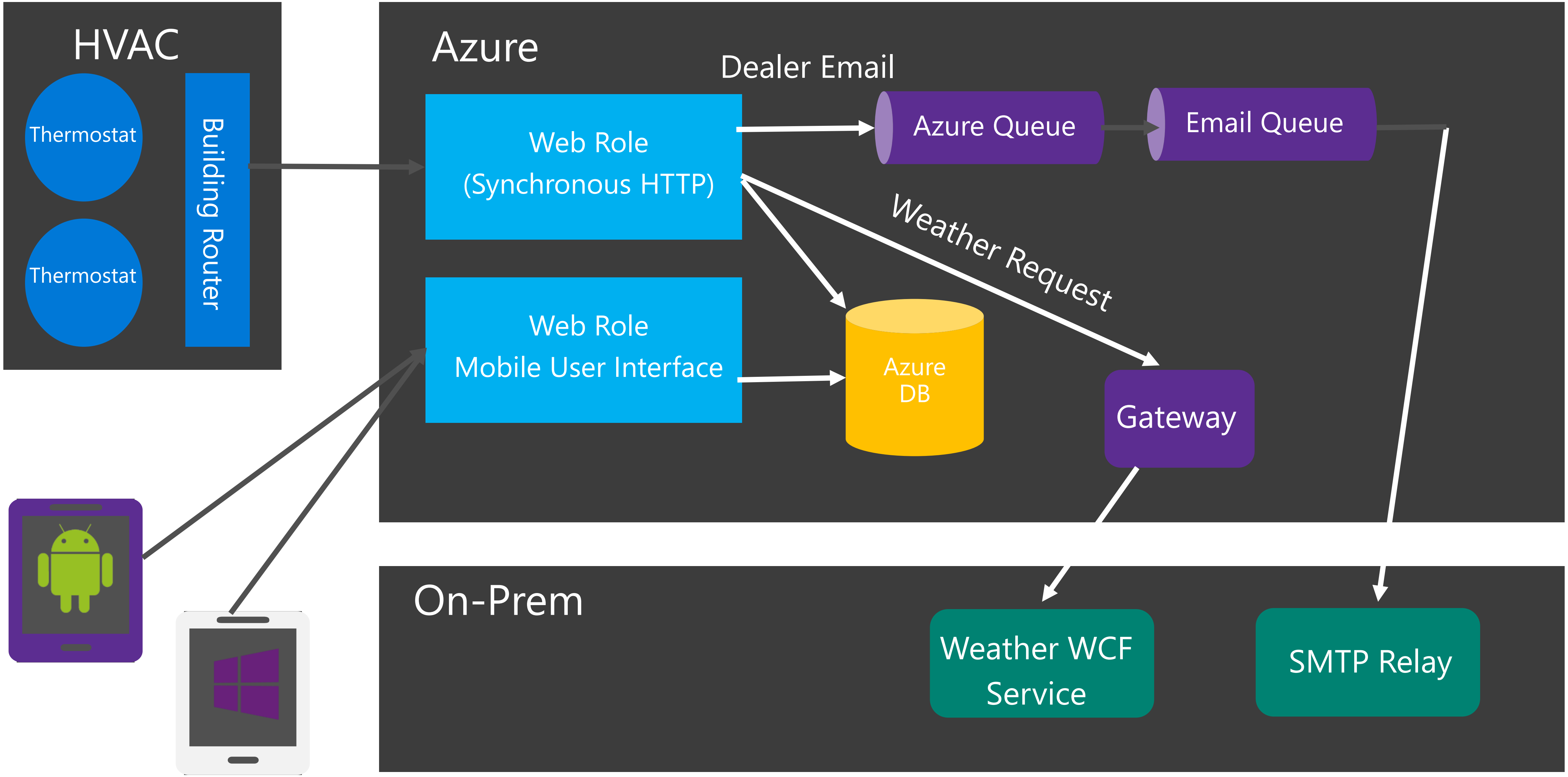
Smart Thermostats

“Let me make you comfortable”

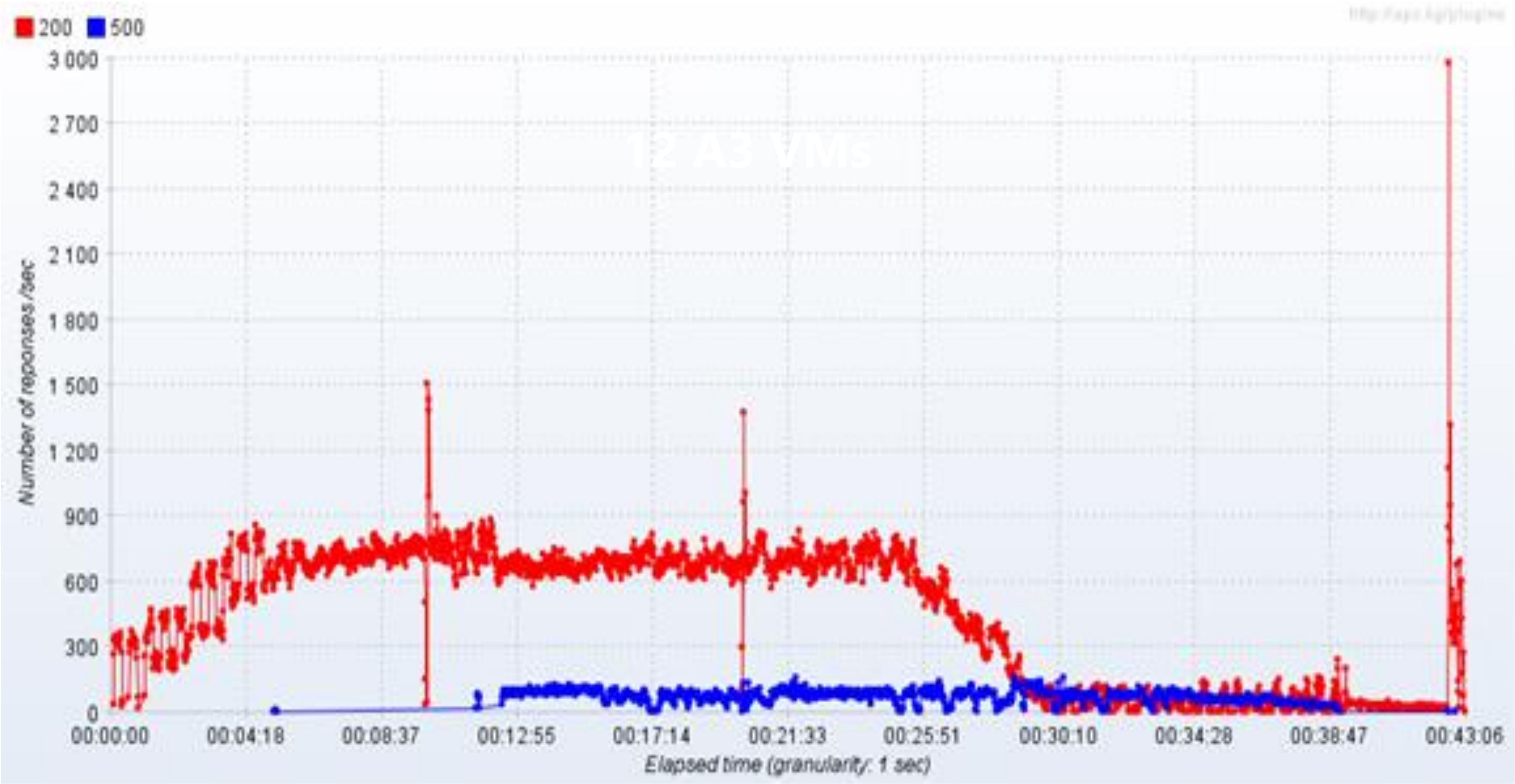
Your Temperature is Too High

- Leading HVAC company created new temperature management service on Azure PaaS
  - Thermostats report temperature to cloud service
  - Cloud service serves as control point for devices and schedules to remotely set target temperatures
- Initial product release failed to scale past more than 35,000 connected thermostats
  - Target was 100,000
  - Stretch goal was 150,000
- Azure CAT team called in…

Architecture



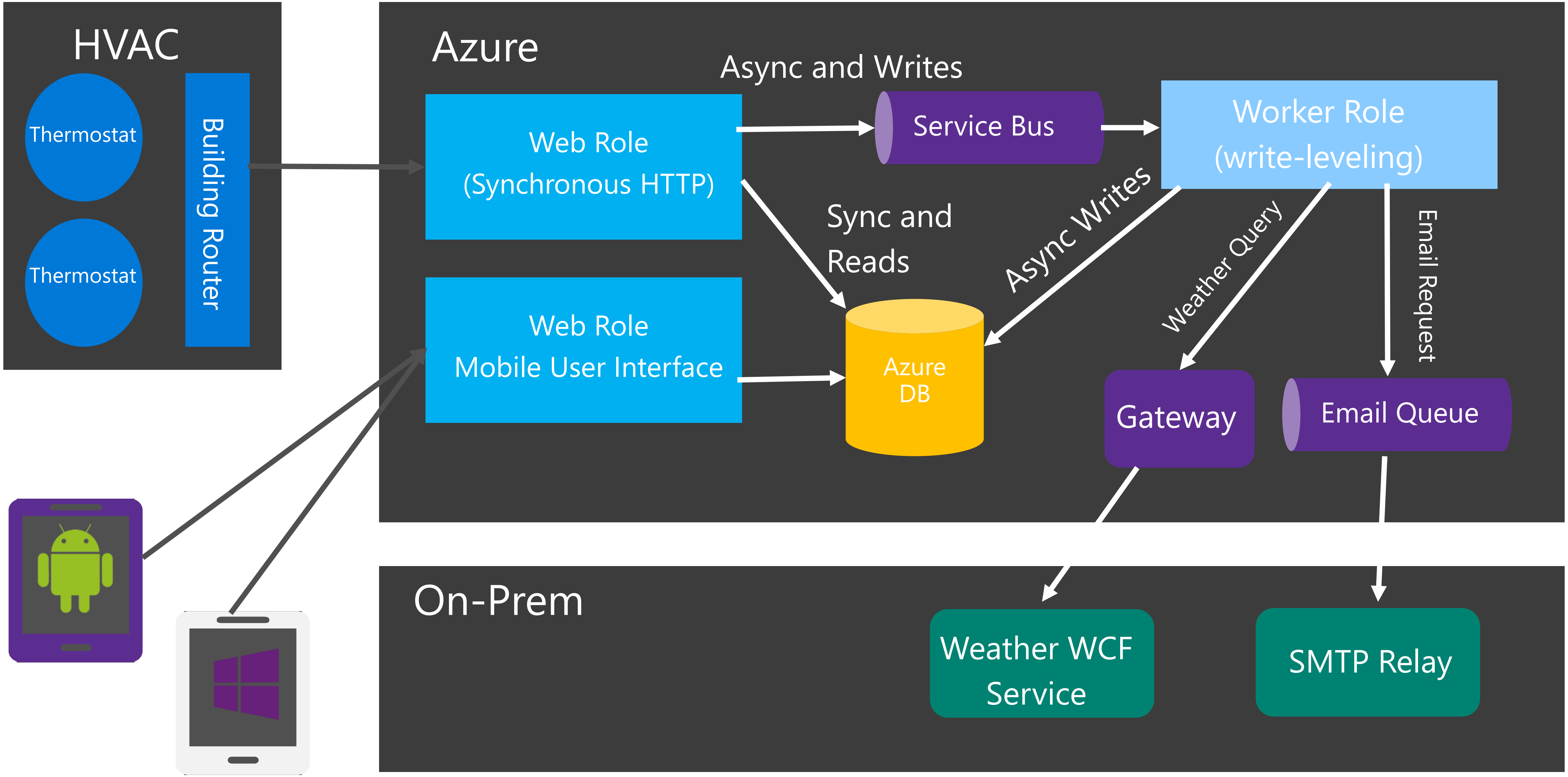
Initial Performance Results



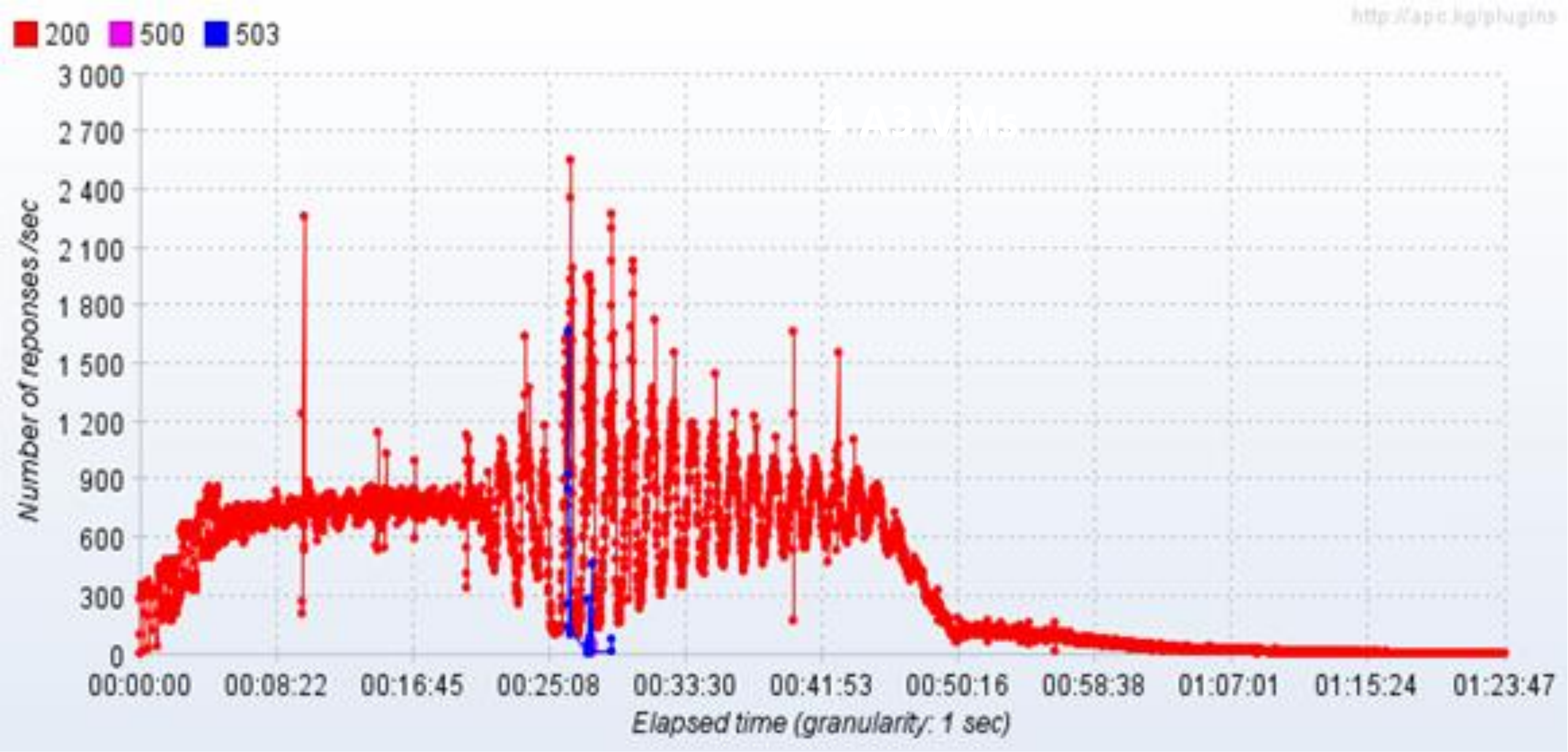
Don’t Wait Up For Me

- Biggest issue: synchronous HTTP handler
  - Changed so only interactive queries synchronous
- Single row updates to DB
  - Changed to multi-row batch updates
- Single low-end Azure DB
  - Moved hot tables to premium DB
- XML parameters preserved to DB
  - Convert XML to Table Valued Parameters
- Single Azure Queue
  - Switch to multi-partition Service Bus queues

Updated Architecture



Much Better!


http://apc.kg/plugins
🟥 200
🟪 500
🟦 503
4 A3 VMs
Number of reponses /sec
3 000
2 700
2 400
2 100
1 800
1 500
1 200
900
600
300
0
00:00:00
00:08:22
00:16:45
00:25:08
00:33:30
00:41:53
00:50:16
00:58:38
01:07:01
01:15:24
01:23:47
Elapsed time (granularity: 1 sec)

Smart Card Service

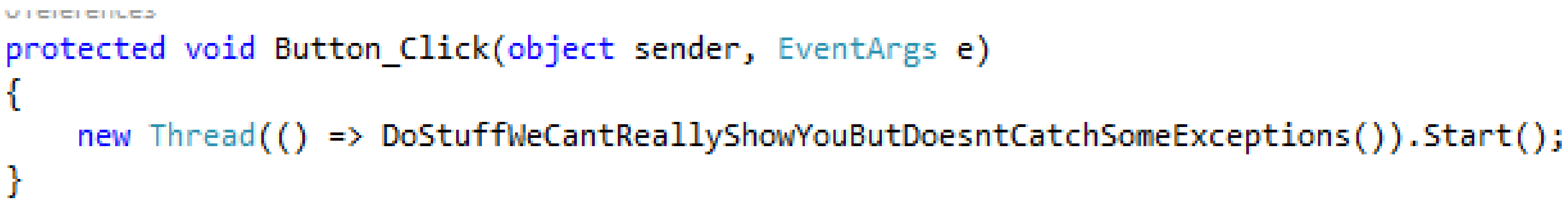
“Not as smart as it appears”

Azure Smart Card Provider

- Leading smart card authentication company created Azure service for eCommerce
- Traditional SOA with Web Role, Cache Role, and Azure DB
- Problem: Web Role randomly crashed

Some Concurrency is Good

- Analysis localized issues to Web Role posting jobs to local threads:



```
protected void Button_Click(object sender, EventArgs e)
{
    new Thread(() => DoStuffWeCantReallyShowYouButDoesntCatchSomeExceptions()).Start();
}
```

- Two problems:
  - If backends go down or slow down, threads pile up and exhaust server resources
  - An exception on the thread takes down the process
- Resolution: moved worker logic to Worker Role and connected to Web Role via Azure Storage queue

Demo:
Isolating Background Jobs

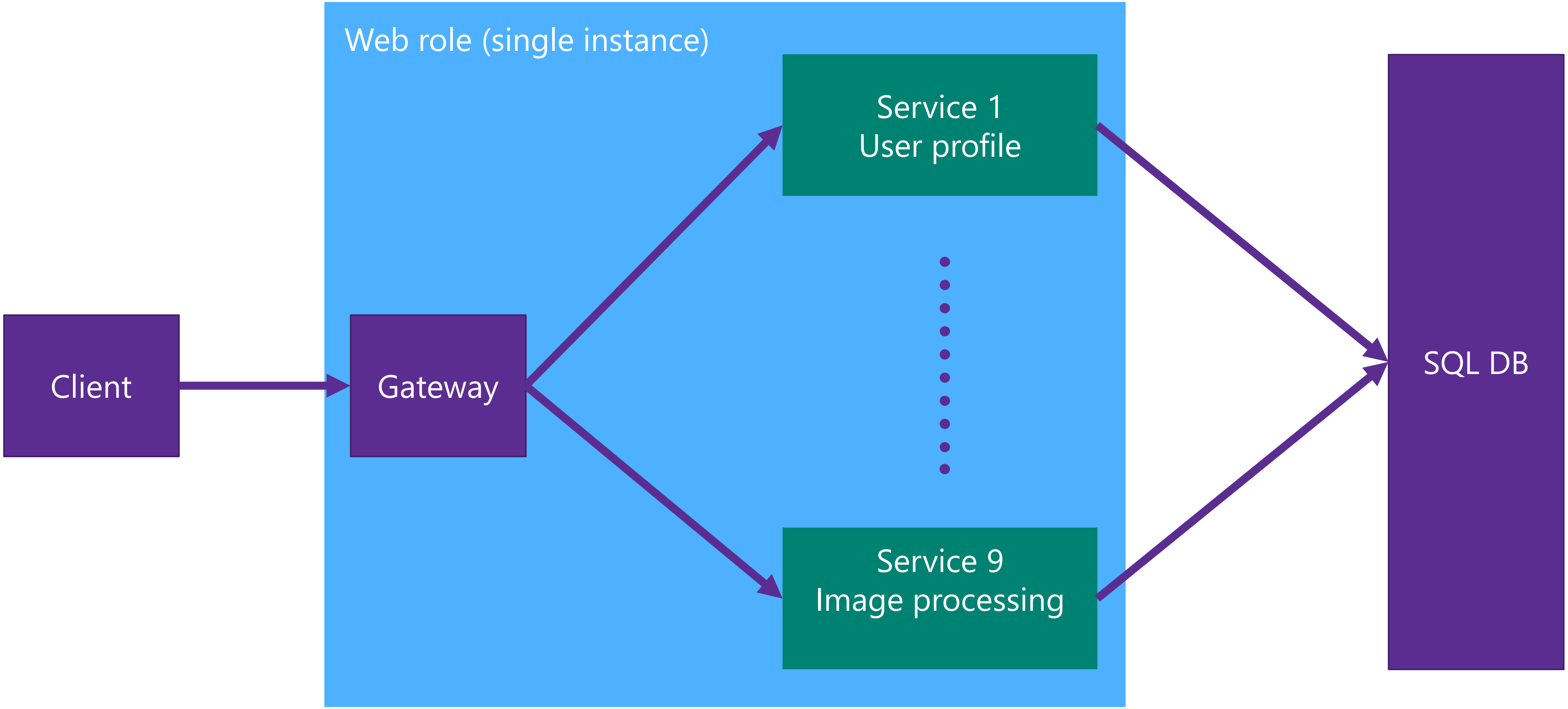
Photo Sharing Service

“Share your life with friends and family”

50 is less than 7,000

- Cloud storage with image processing capability
  - Unlimited storage with thumbnails and image correction
- First release had a limit of 50 request per sec
  - Target was 7,000 rps
- Monolithic architecture with a few major issues
  - Need distributed system principles in place
- Azure onboarding team called in…

Monolithic, Synchronous and Monolingual



Web role (single instance)

Client

Gateway

Service 1
User profile

Service 9
Image processing

SQL DB

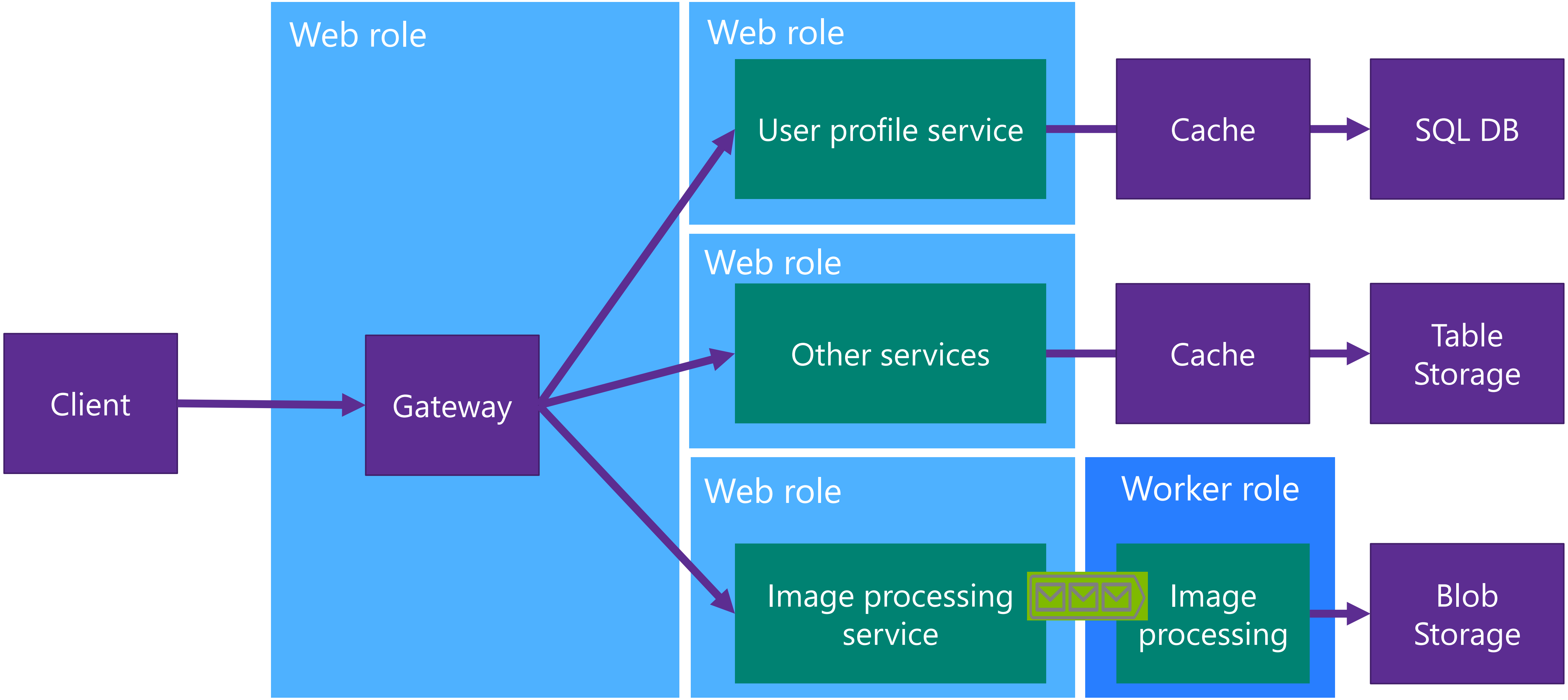
Refactoring to gain x150 performance

- Monolithic architecture
  - Decomposed workload, Moved CPU intensive tasks to worker role
- Synchronous I/O calls across architecture
  - Changed to async calls
- SQL DB for every data type
  - Optimized the storage for each entity
- Lack of caching
  - Added caching between app and data tier
- Instantiating objects per every call
  - Changed to singleton or object pooling

Distributed, Asynchronous and Polyglot

Decompose workloads into different roles

Optimal storage per each data type



Async calls all around

Use queue and worker for CPU intensive task

Demo:
Sync vs Async

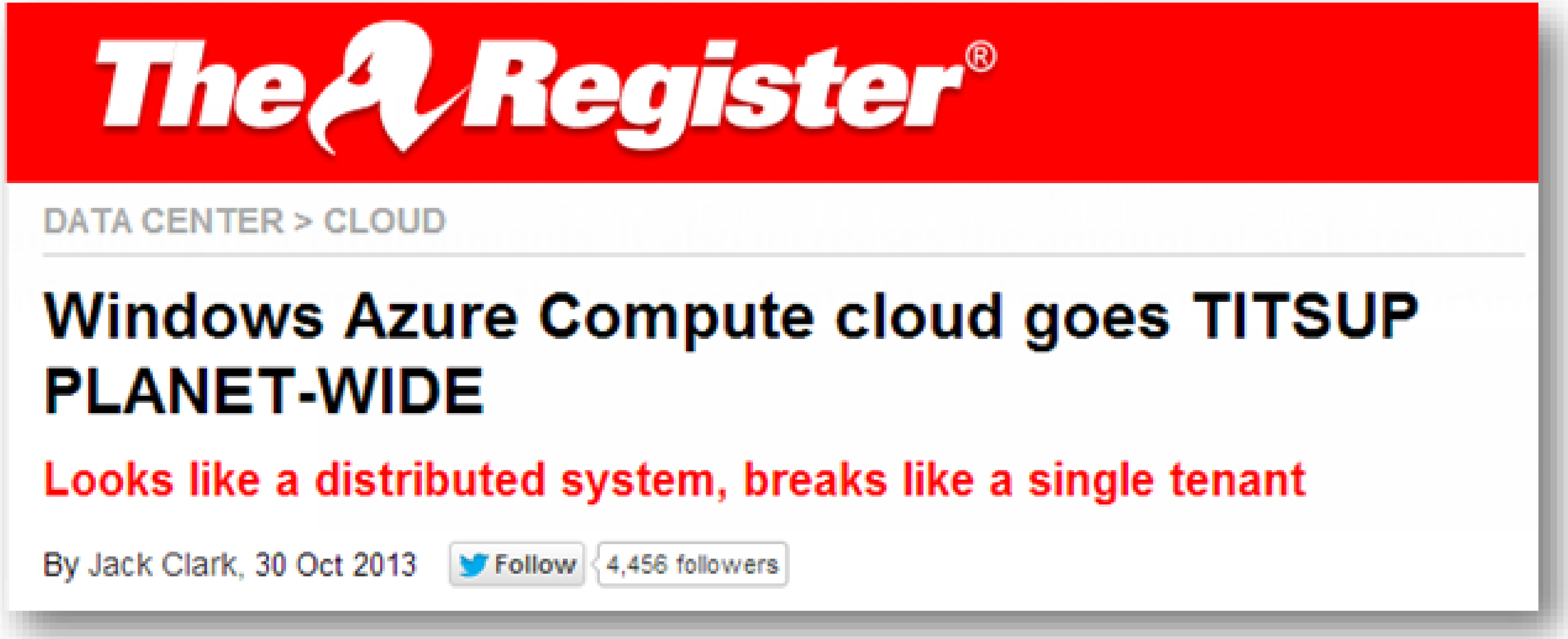
Azure Lessons

VIP Swap

“I like your VIP better than mine”

Really? Isn't that a Bit Much?

- Users started complaining that after a VIP swap that they could not perform operations on their cloud services
  - Was not detected by monitoring systems
  - Affected only a small number of customers



The Register®

DATA CENTER > CLOUD

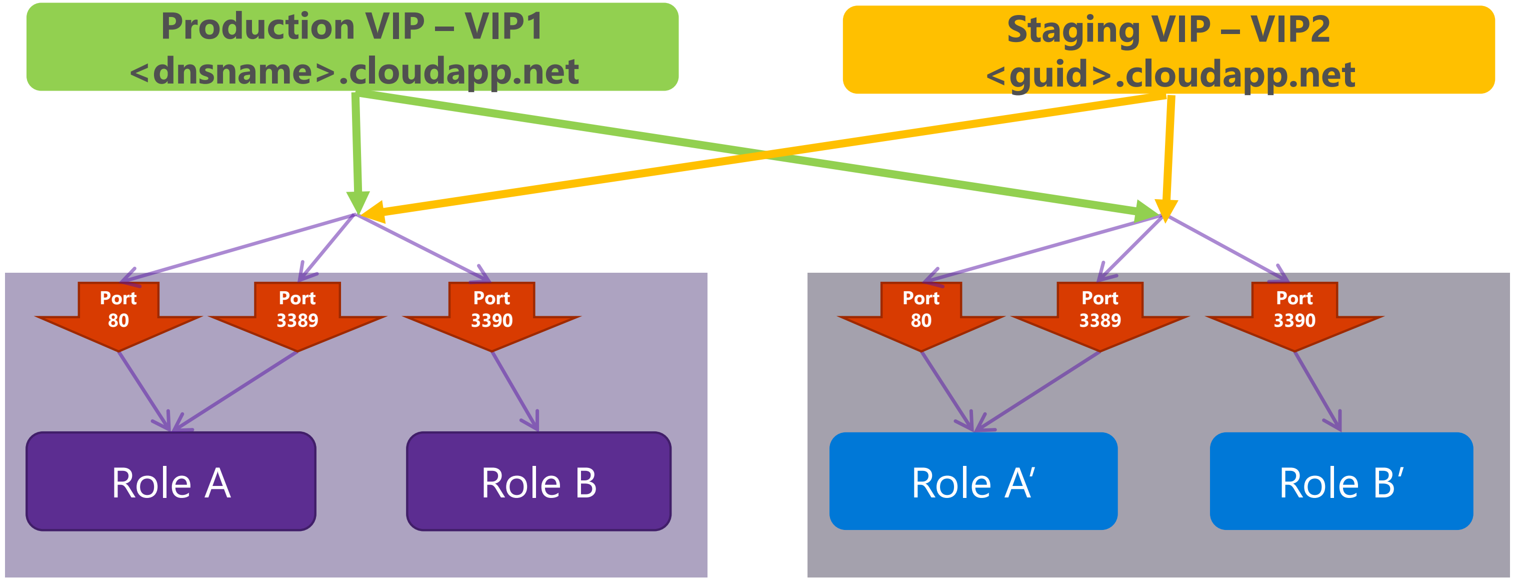
Windows Azure Compute cloud goes TITSUP PLANET-WIDE

Looks like a distributed system, breaks like a single tenant

By Jack Clark, 30 Oct 2013

What’s a VIP Swap?

- You can deploy two versions of a cloud service:
  - Production: has the DNS name and IP address of the cloud service you publish
  - Stage: has a temporary DNS name and IP address
- To promote the Stage version to Production, you “VIP Swap”



VIP Swap Internals

- RDFE uses storage table rows to cache the state of cloud service deployments
  - Includes state of role instances and deployment slots
  - Row is updated by mutating operations like VIP Swap
  - It’s also updated by RDFE cache updating status of roles
- Multiple roles updated via table conditional update (opportunistic concurrency)

Slot	VIP	Role A	Role B
Stage	168.124.33.22	Healthy	Healthy
Production	168.133.1.22	Healthy	Healthy

The VIP Swap Bug

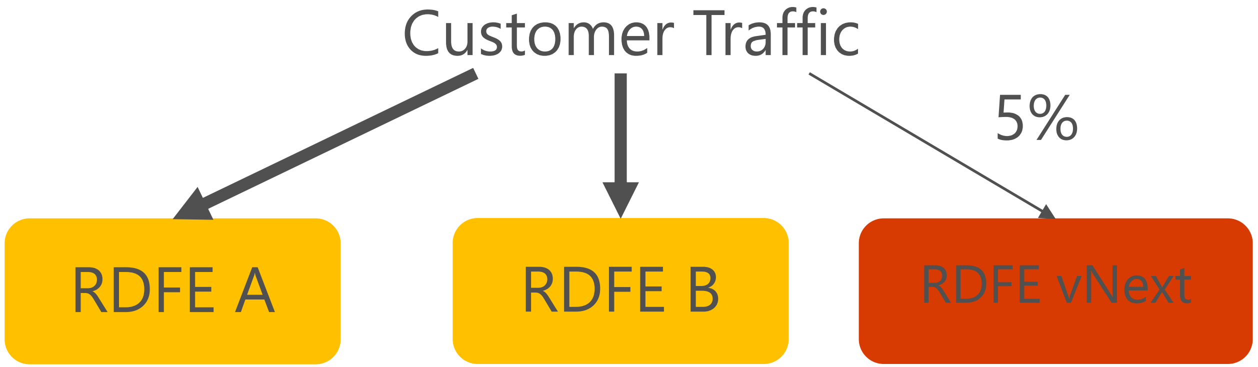
- Bug in RDFE update caused race condition
  - Change would be overwritten, causing inconsistent state

Slot	VIP	Role A	Role B
Stage	168.124.33.22	Healthy	Healthy
Stage	168.124.33.22	Healthy	Healthy

  - RDFE does not allow update operations when it detects inconsistency
- Race condition meant error rate was only marginally higher than normal and went undetected

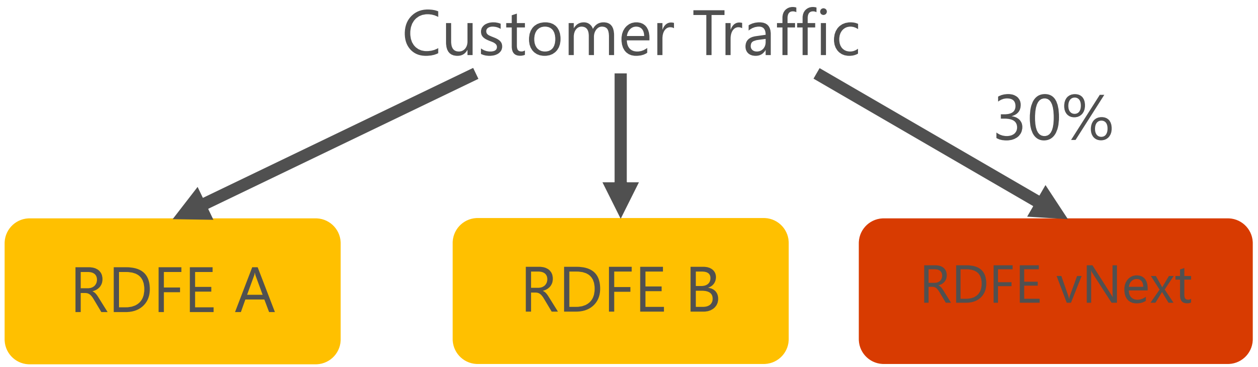
VIP Swap Learnings

- Root cause: developer claimed “unintuitive behavior of ADO.NET”
- Rule: direct a slice of traffic to an updated version
  - Increase traffic gradually
  - Set alerts based on difference in failure rates of two versions



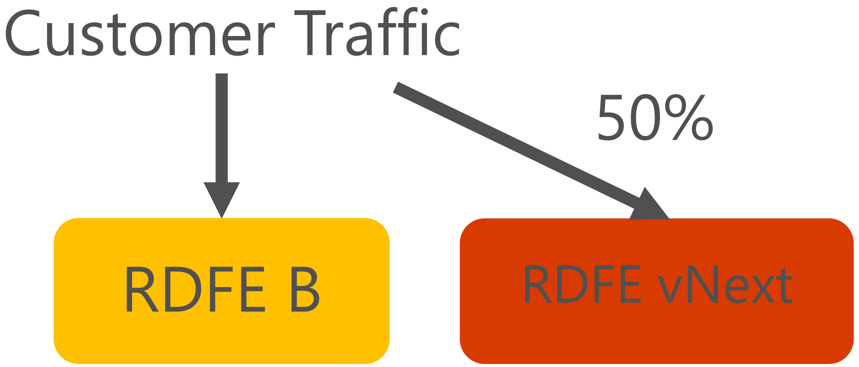
VIP Swap Learnings

- Root cause: developer claimed “unintuitive behavior of ADO.NET”
- Rule: direct a slice of traffic to an updated version for several days
  - Increase traffic gradually
  - Set alerts based on difference in failure rates of two versions



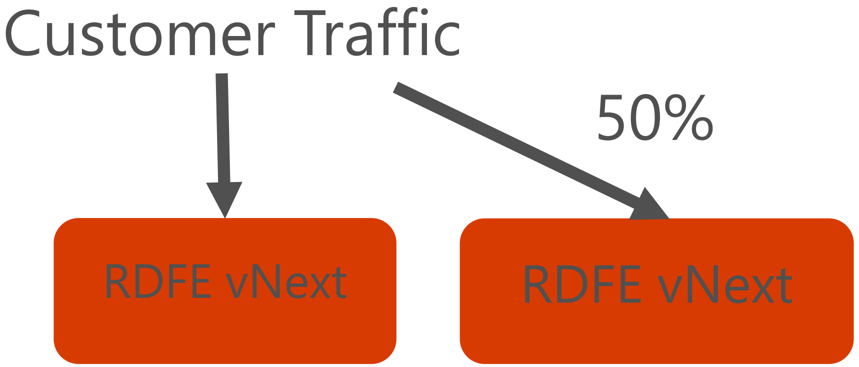
VIP Swap Learnings

- Root cause: developer claimed "unintuitive behavior of ADO.NET"
- Rule: direct a slice of traffic to an updated version for several days
  - Increase traffic gradually
  - Set alerts based on difference in failure rates of two versions



VIP Swap Learnings

- Root cause: developer claimed "unintuitive behavior of ADO.NET"
- Rule: direct a slice of traffic to an updated version for several days
  - Increase traffic gradually
  - Set alerts based on difference in failure rates of two versions



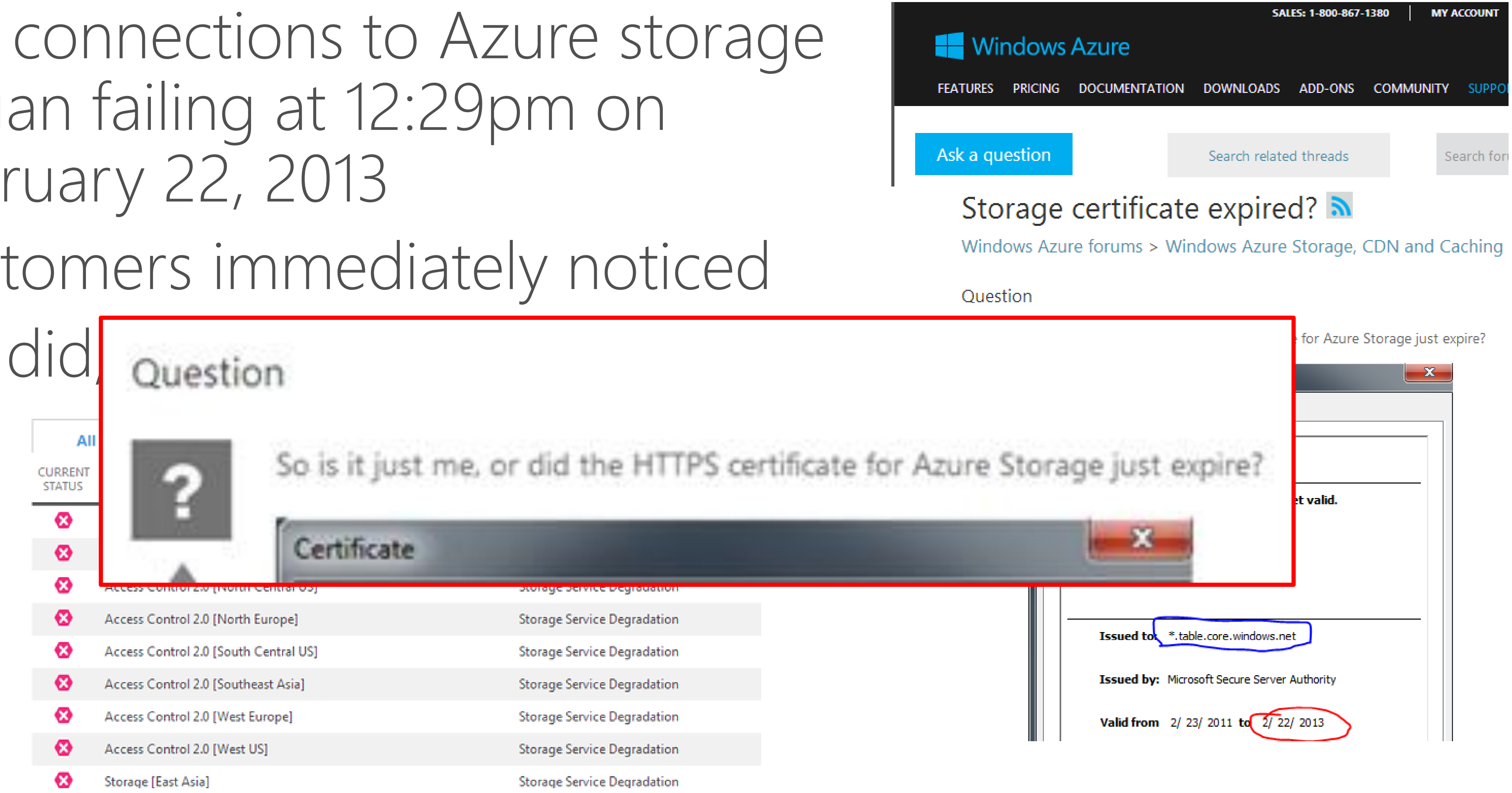
Storage Certificate Expiration

“Sorry I’m late, the alarm clock never rang”

http://blogs.msdn.com/b/windowsazure/archive/2013/03/01/details-of-the-february-22nd-2013-windows-azure-storage-disruption.aspx

It’s Not You, It’s Me

- SSL connections to Azure storage began failing at 12:29pm on February 22, 2013
- Customers immediately noticed
- We did



Storage certificate expired?

Windows Azure forums > Windows Azure Storage, CDN and Caching

Question

for Azure Storage just expire?

Question

So is it just me, or did the HTTPS certificate for Azure Storage just expire?

Certificate

t valid.

Issued to: *.table.core.windows.net

Issued by: Microsoft Secure Server Authority

Valid from 2/ 23/ 2011 to 2/ 22/ 2013

All

CURRENT STATUS		
✖		
✖		
✖	Access Control 2.0 [North Central US]	Storage Service Degradation
✖	Access Control 2.0 [North Europe]	Storage Service Degradation
✖	Access Control 2.0 [South Central US]	Storage Service Degradation
✖	Access Control 2.0 [Southeast Asia]	Storage Service Degradation
✖	Access Control 2.0 [West Europe]	Storage Service Degradation
✖	Access Control 2.0 [West US]	Storage Service Degradation
✖	Storage [East Asia]	Storage Service Degradation

We Updated It, We Promise!

- Certificates are managed by the “Secret Store”
  - Once a week an automated system scans the store
  - An alert is fired for certs within 180 days of expiration
  - Team obtains new cert and updates Secret Store
- That process was followed
- The breakdown:
  - On January 7, the storage team updated the three certs in question
  - Failed to flag that a storage deployment had a date deadline
  - Deployment was delayed behind other higher-priority update

Be Certain About Your Certs

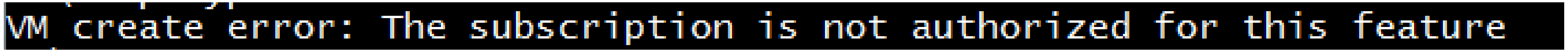
- The real breakdown was not monitoring production:
  - We now scan all service endpoints, internal and external, on a weekly basis
  - At 90 days until expiration, shows up on VP reports
- Rule: service development requires thinking through the entire life-cycle of the software
- We are working on “managed service identities” to fully automate non-PKI certs

Log As If That's All You Have

- A little more detail can go a long way…
- Error log not reporting a name made correlation difficult:

```
System.Reflection.TargetInvocationException: Exception has been thrown by the target of an
invocation. ---> Microsoft.ServiceModel.Web.WebProtocolException: Server Error: The service
name is unknown (NotFound)
```

- Error message in test environment indicating a beta feature was missing was ambiguous:



```
VM create error: The subscription is not authorized for this feature
```

- Intermittent failures because of header incompatibility in test environment made troubleshooting painful:

```
HTTP Status Code: 400. Service Management Error Code:
MissingOrIncorrectVersionHeader. Message: The versioning header is not specified or was specified incorrectly.
```

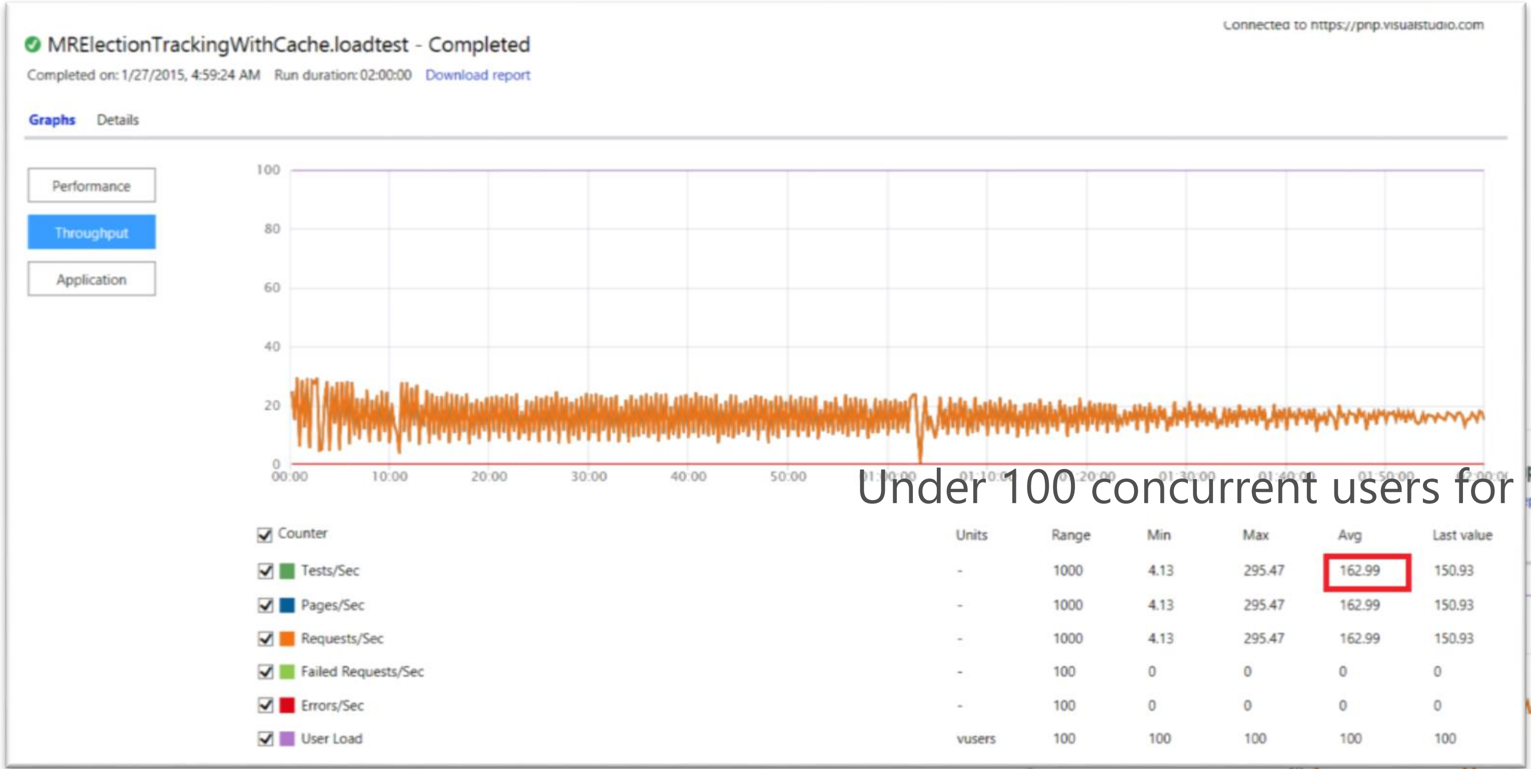
Lessons From Scale

- Cache aggressively to hide latency
- Async with queues when possible
- Process in batches to minimize round trips
- Partition data and compute to scale out
- Roll out with monitored slices
- Log excessively

Election Tracking Demo - results

Throughput – NoCache vs. Cache

270% more throughput with cache



Connected to https://pnp.visualstudio.com

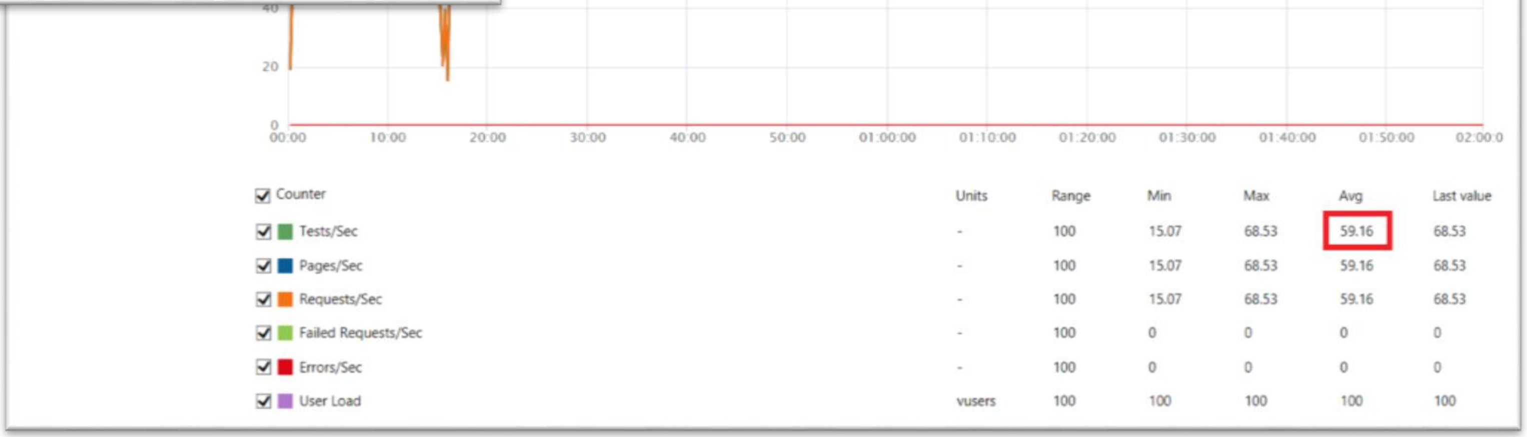
MRElectionTrackingWithCache.loadtest - Completed

Completed on: 1/27/2015, 4:59:24 AM Run duration: 02:00:00 Download report

Graphs Details

☑ Counter	Units	Range	Min	Max	Avg	Last value
☑ 🟩 Tests/Sec	-	1000	4.13	295.47	162.99	150.93
☑ 🟦 Pages/Sec	-	1000	4.13	295.47	162.99	150.93
☑ 🟧 Requests/Sec	-	1000	4.13	295.47	162.99	150.93
☑ 🟩 Failed Requests/Sec	-	100	0	0	0	0
☑ 🟥 Errors/Sec	-	100	0	0	0	0
☑ 🟪 User Load	vusers	100	100	100	100	100

Under 100 concurrent users for 1 hour



pleted

port

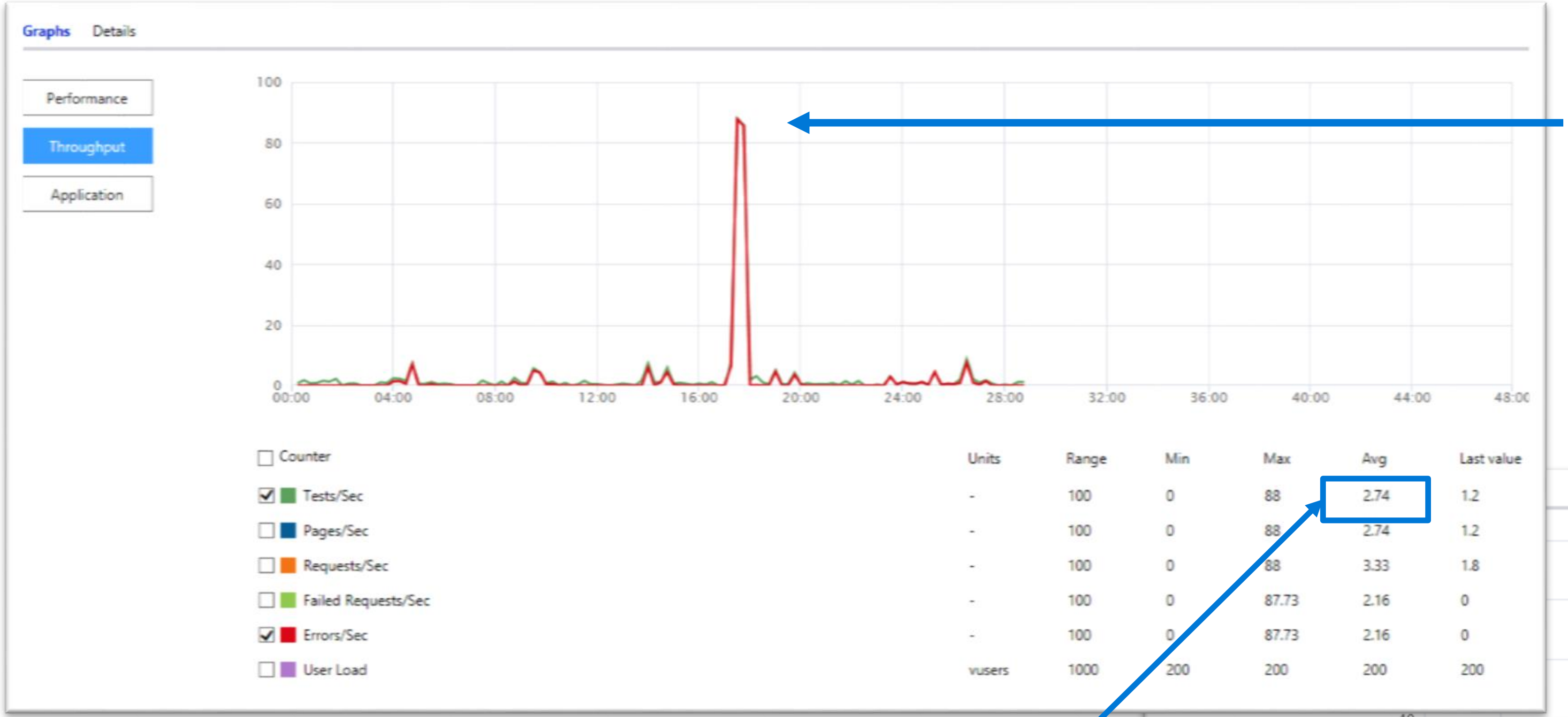
Connected to https://pnp.visualstudio.com

☑ Counter	Units	Range	Min	Max	Avg	Last value
☑ 🟩 Tests/Sec	-	100	15.07	68.53	59.16	68.53
☑ 🟦 Pages/Sec	-	100	15.07	68.53	59.16	68.53
☑ 🟧 Requests/Sec	-	100	15.07	68.53	59.16	68.53
☑ 🟩 Failed Requests/Sec	-	100	0	0	0	0
☑ 🟥 Errors/Sec	-	100	0	0	0	0
☑ 🟪 User Load	vusers	100	100	100	100	100

Under 100 concurrent users for 1 hour

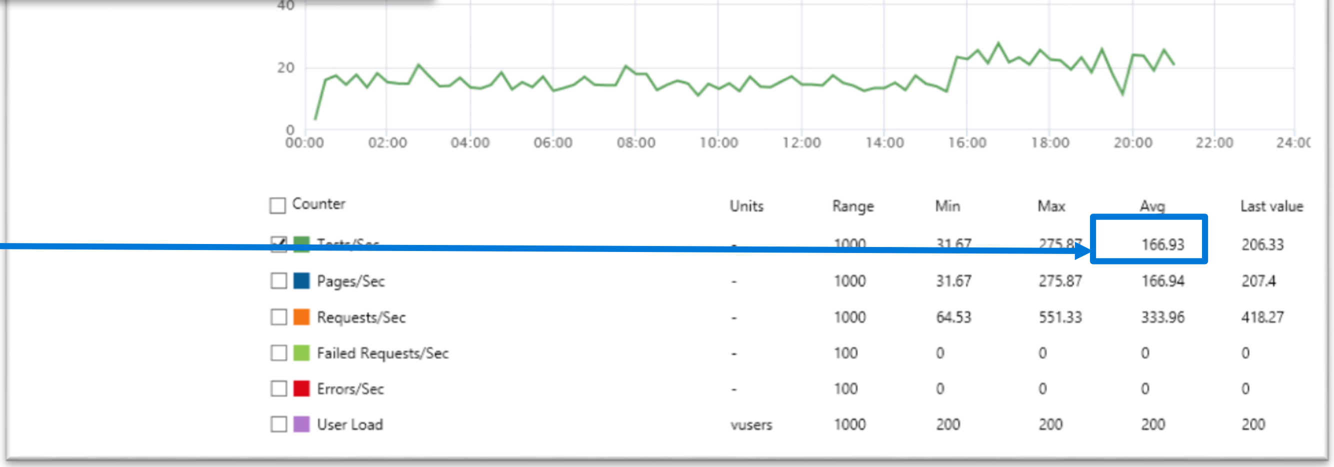
Smart Card Service Demo - results

Throughput - Monolithic vs. Distributed



□ Counter	Units	Range	Min	Max	Avg	Last value
☑ 🟩 Tests/Sec	-	100	0	88	2.74	1.2
□ 🟦 Pages/Sec	-	100	0	88	2.74	1.2
□ 🟧 Requests/Sec	-	100	0	88	3.33	1.8
□ 🟩 Failed Requests/Sec	-	100	0	87.73	2.16	0
☑ 🟥 Errors/Sec	-	100	0	87.73	2.16	0
□ 🟪 User Load	vusers	1000	200	200	200	200

Monolithic architecture causes resource starvation

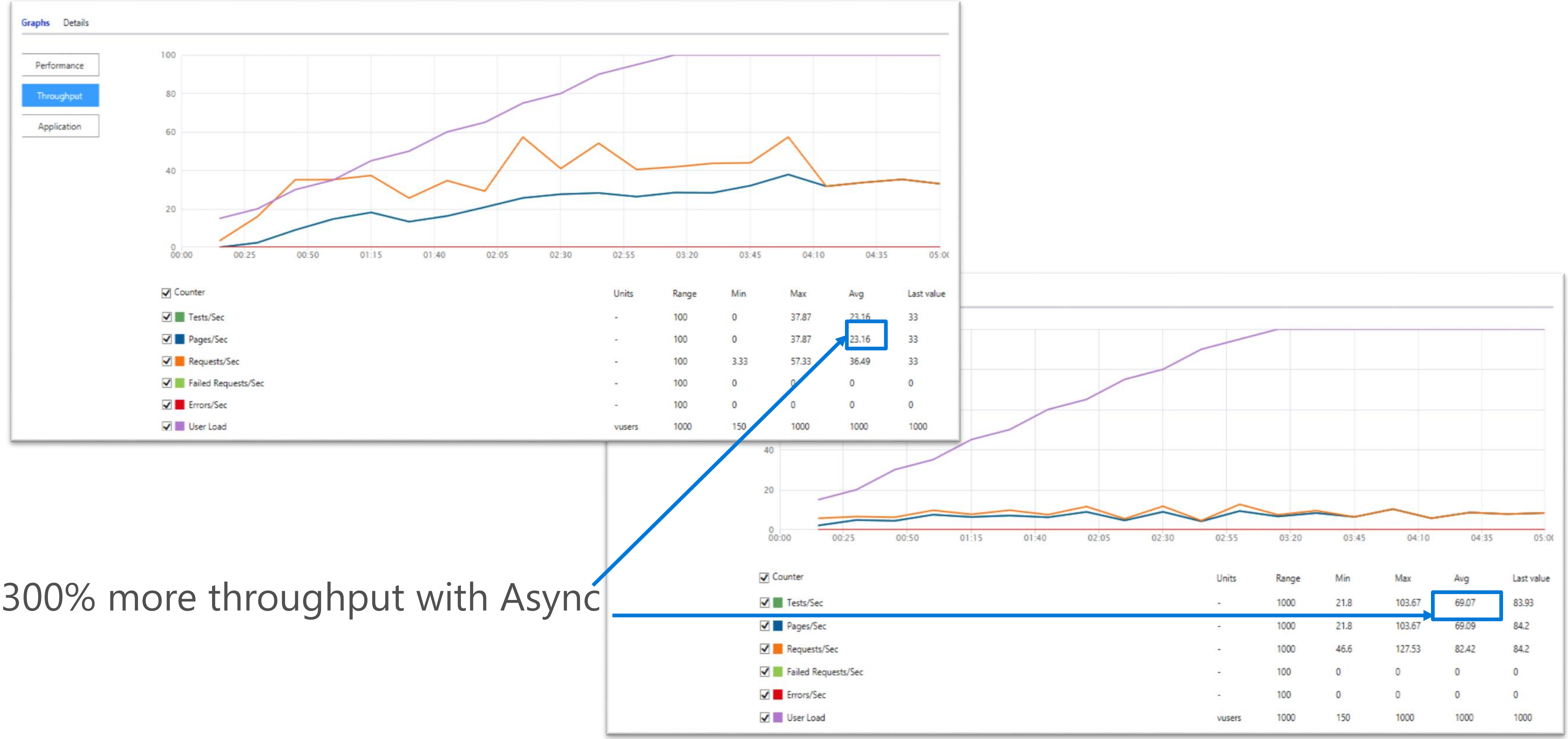


□ Counter	Units	Range	Min	Max	Avg	Last value
☑ 🟩 Tests/Sec	-	1000	31.67	275.87	166.93	206.33
□ 🟦 Pages/Sec	-	1000	31.67	275.87	166.94	207.4
□ 🟧 Requests/Sec	-	1000	64.53	551.33	333.96	418.27
□ 🟩 Failed Requests/Sec	-	100	0	0	0	0
□ 🟥 Errors/Sec	-	100	0	0	0	0
□ 🟪 User Load	vusers	1000	200	200	200	200

x60 higher throughput by isolating CPU intensive tasks

Photo Sharing Demo - results

Throughput – Sync vs. Async



☑ Counter	Units	Range	Min	Max	Avg	Last value
☑ 🟩 Tests/Sec	-	100	0	37.87	23.16	33
☑ 🟦 Pages/Sec	-	100	0	37.87	23.16	33
☑ 🟧 Requests/Sec	-	100	3.33	57.33	36.49	33
☑ 🟩 Failed Requests/Sec	-	100	0	0	0	0
☑ 🟥 Errors/Sec	-	100	0	0	0	0
☑ 🟪 User Load	vusers	1000	150	1000	1000	1000

☑ Counter	Units	Range	Min	Max	Avg	Last value
☑ 🟩 Tests/Sec	-	1000	21.8	103.67	69.07	83.93
☑ 🟦 Pages/Sec	-	1000	21.8	103.67	69.09	84.2
☑ 🟧 Requests/Sec	-	1000	46.6	127.53	82.42	84.2
☑ 🟩 Failed Requests/Sec	-	100	0	0	0	0
☑ 🟥 Errors/Sec	-	100	0	0	0	0
☑ 🟪 User Load	vusers	1000	150	1000	1000	1000

300% more throughput with Async