Microsoft

Azure SQL Database Performance Tuning

Jes Borland
jes.borland@microsoft.com

Jes Borland

Premier Field Engineer
Microsoft
jes.borland@microsoft.com
@grrl_geek
LessThanDot.com

Tech Outbound Alaska 2018

- Aug 4-11, 2018
- Depart and return Seattle, WA
- The best training and mentoring you'll ever receive

- Topics include Performance Tuning, PowerShell, SQL on Linux, Security, Python, R, Data Science Fundamentals, Cosmos DB, and more!

Let's talk about

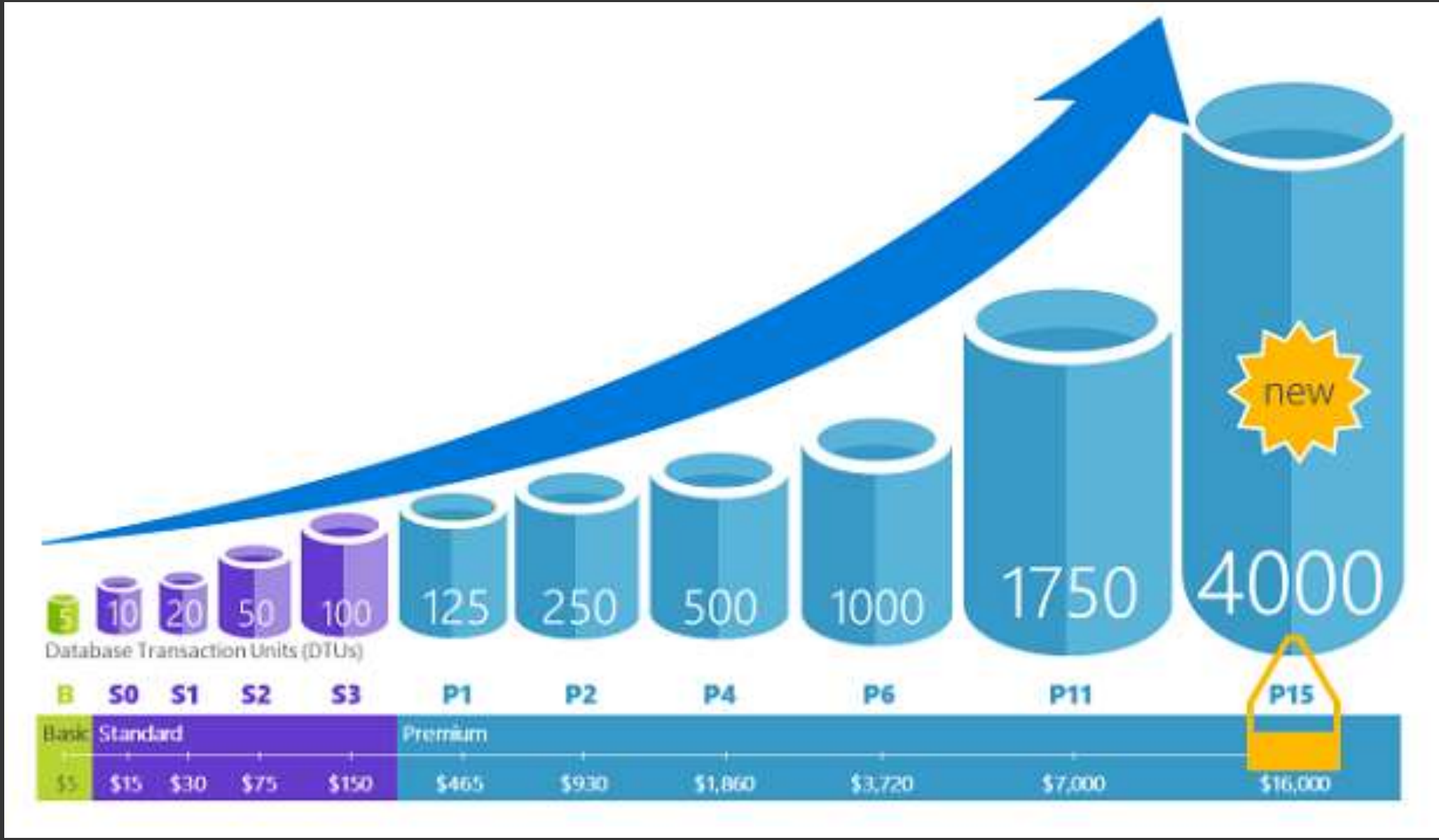
- Paying for performance
- Monitoring
- Indexes
- In-Memory OLTP
- Operational Analytics
- Scaling
- Query Performance Insight
- Automatic Tuning

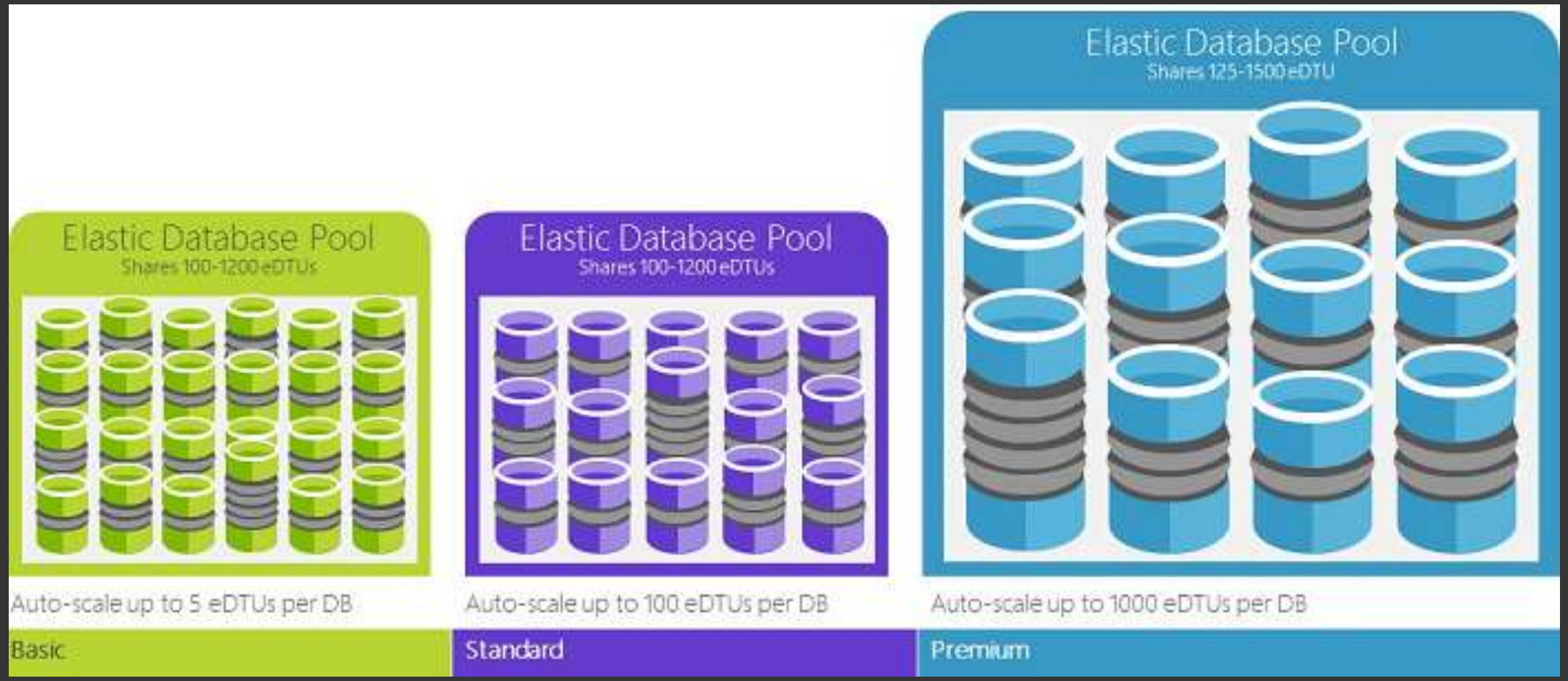
Paying for performance

How we measure (and pay for) performance in SQL Database

Database Throughput Units

- **“A blended measure of CPU, memory, I/O (data and transaction log I/O)”**
- **Guaranteed performance**
  - When workload exceeds one of those resources, throughput is throttled
- **Doubling DTUs by increasing tiers will double the resources available**


new
5
10
20
50
100
125
250
500
1000
1750
4000
Database Transaction Units (DTUs)
B
S0
S1
S2
S3
P1
P2
P4
P6
P11
P15
Basic
Standard
Premium
$5
$15
$30
$75
$150
$465
$930
$1,860
$3,720
$7,000
$16,000


Elastic Database Pool
Shares 100-1200 eDTUs
Auto-scale up to 5 eDTUs per DB
Basic
Elastic Database Pool
Shares 100-1200 eDTUs
Auto-scale up to 100 eDTUs per DB
Standard
Elastic Database Pool
Shares 125-1500 eDTU
Auto-scale up to 1000 eDTUs per DB
Premium

How many DTUs do I need?

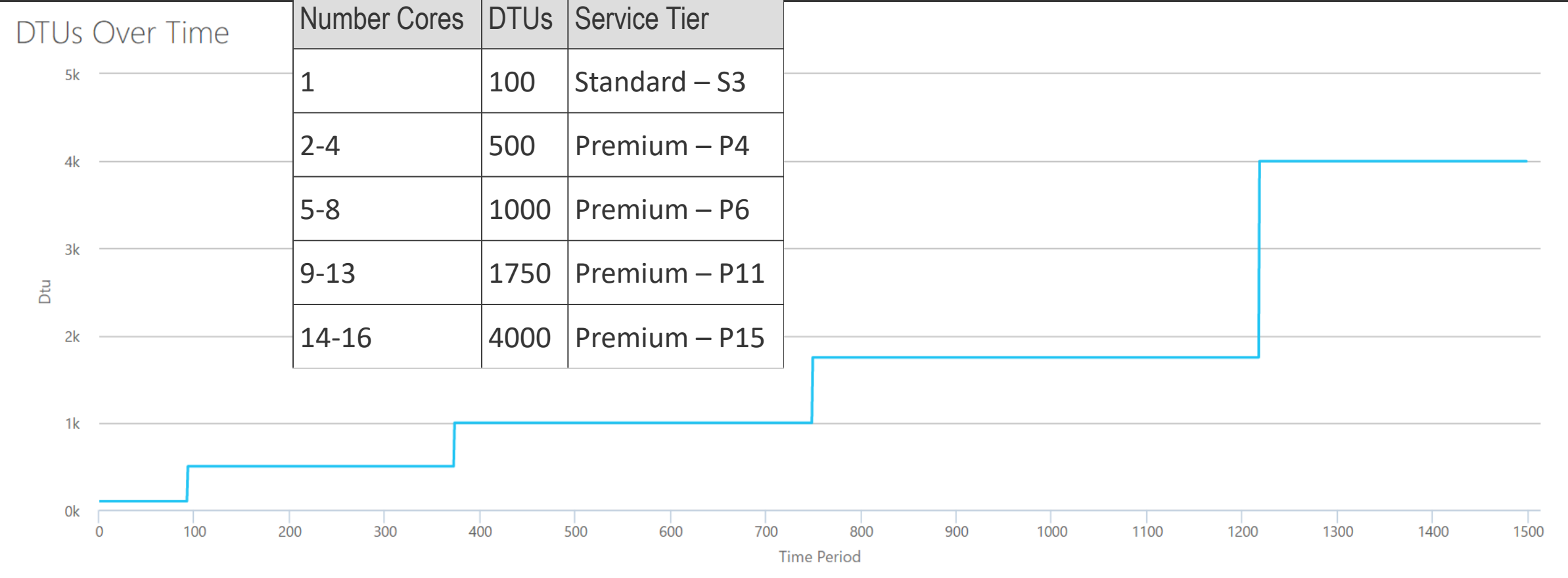
- **Migrating workloads**
  - DTU Calculator - http://dtucalculator.azurewebsites.net/
- **New workloads**
  - Start low, work up

Real life

- What the heck is a DTU? – Andy Mallon https://sqlperformance.com/2017/03/azure/what-the-heck-is-a-dtu
- Input synthetic loads into the DTU Calculator

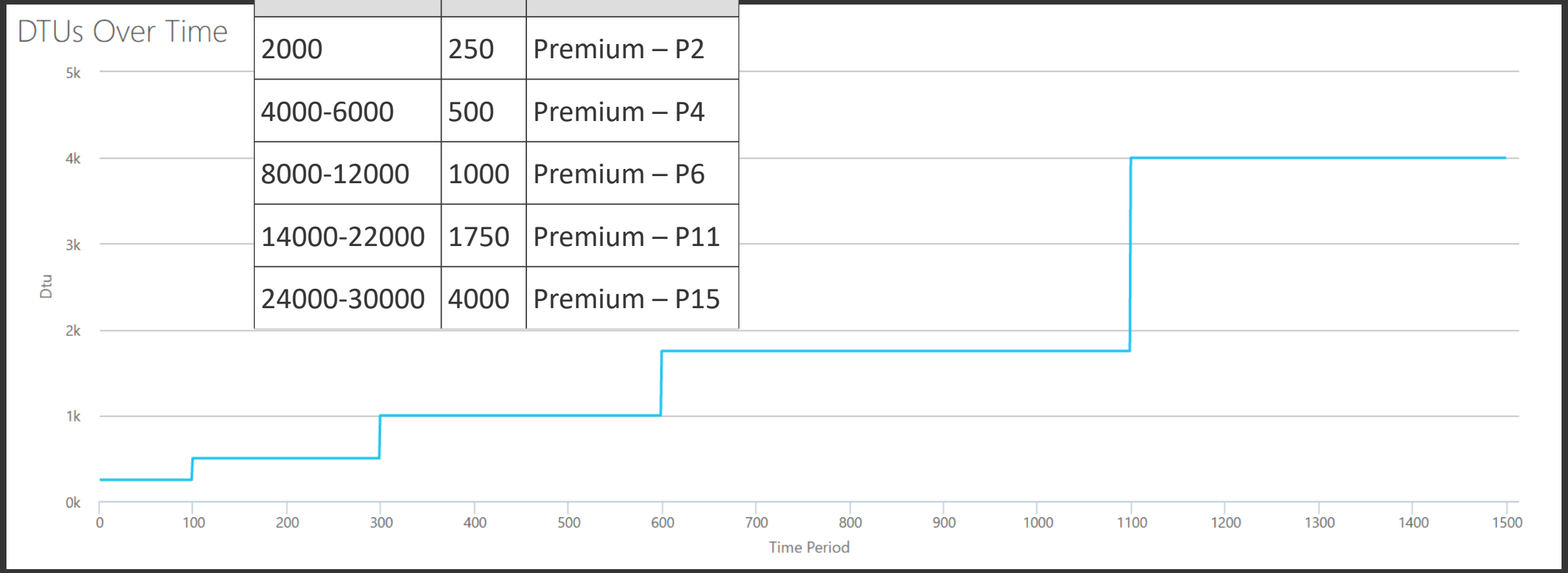
CPU

Number Cores	DTUs	Service Tier
1	100	Standard – S3
2-4	500	Premium – P4
5-8	1000	Premium – P6
9-13	1750	Premium – P11
14-16	4000	Premium – P15



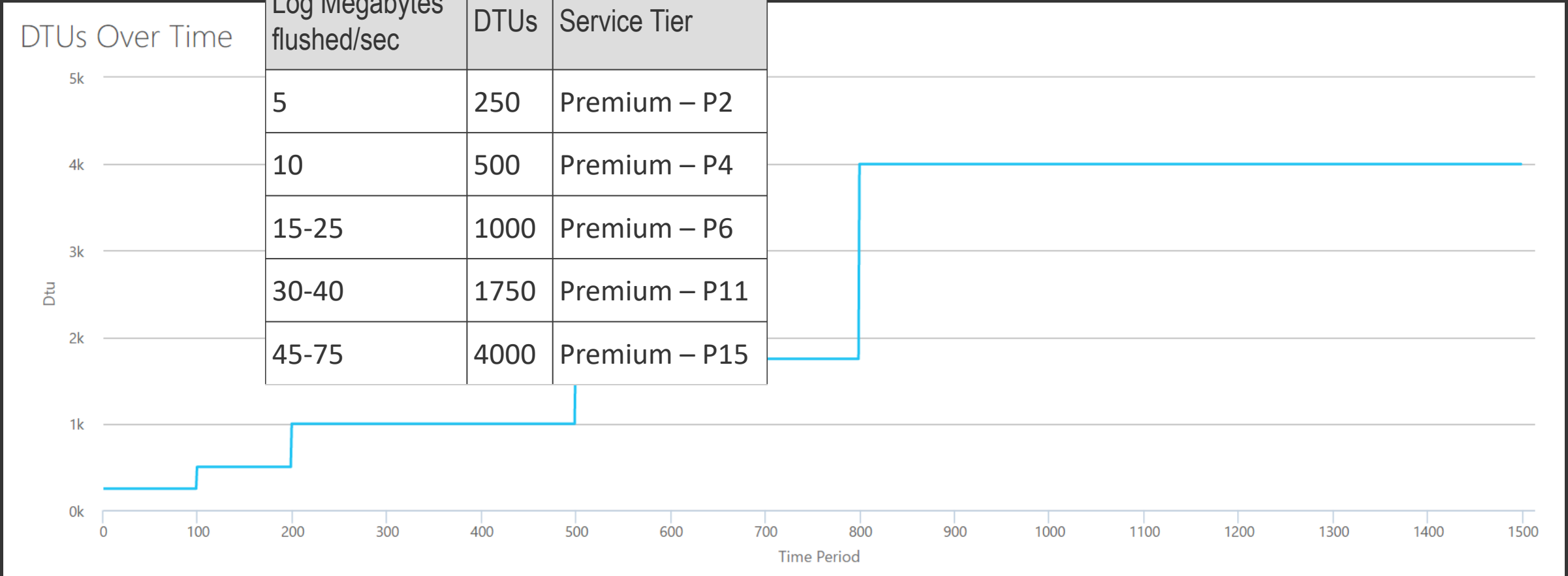
Reads

Reads/sec	DTUs	Service Tier
2000	250	Premium – P2
4000-6000	500	Premium – P4
8000-12000	1000	Premium – P6
14000-22000	1750	Premium – P11
24000-30000	4000	Premium – P15

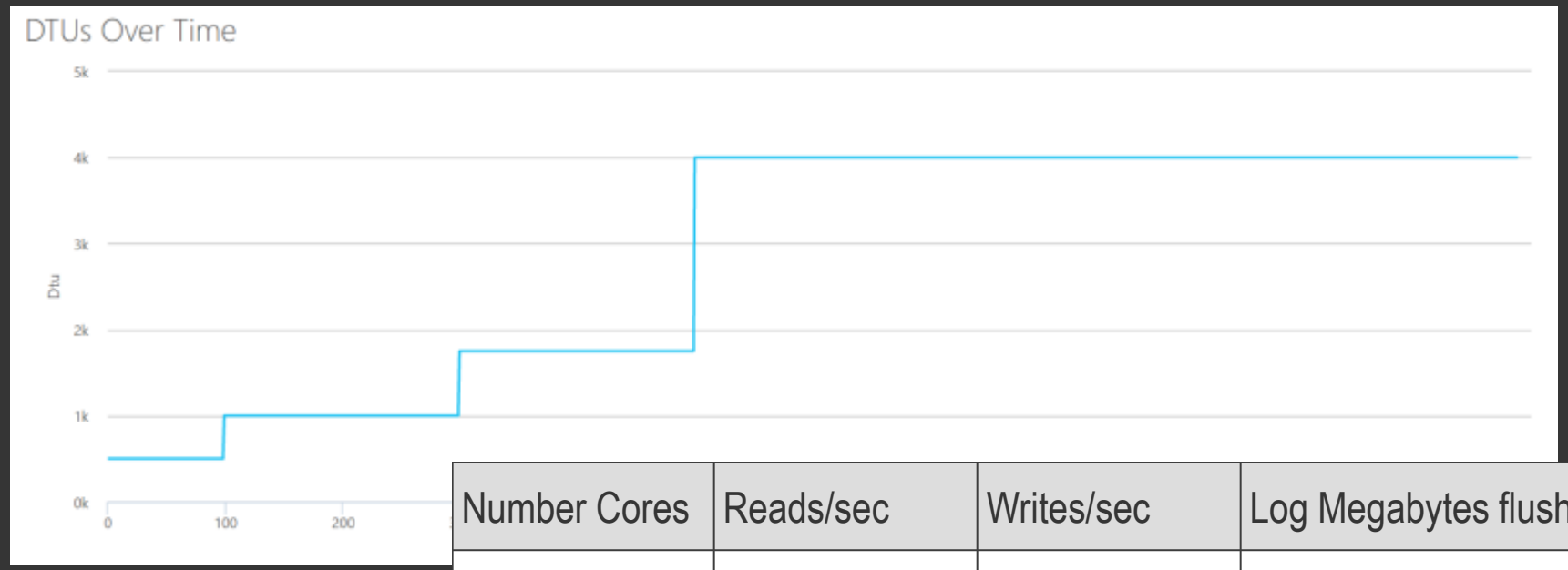


Log Bytes Flushed

Log Megabytes flushed/sec	DTUs	Service Tier
5	250	Premium – P2
10	500	Premium – P4
15-25	1000	Premium – P6
30-40	1750	Premium – P11
45-75	4000	Premium – P15



Combining



Number Cores	Reads/sec	Writes/sec	Log Megabytes flushed/sec	DTUs	Service Tier
1	2000	2000	5	500	Premium – P4
2-3	4000-6000	4000-6000	10	1000	Premium – P6
4-5	8000-10000	8000-10000	15-25	1750	Premium – P11
6-13	12000-24000	12000-24000	30-40	4000	Premium – P15

How expensive are my queries?

- Plan cache and execution plans are the same as SQL Server!
- SentryOne Plan Explorer works for SQL DB, too!
- You can track query performance and history using Query Store

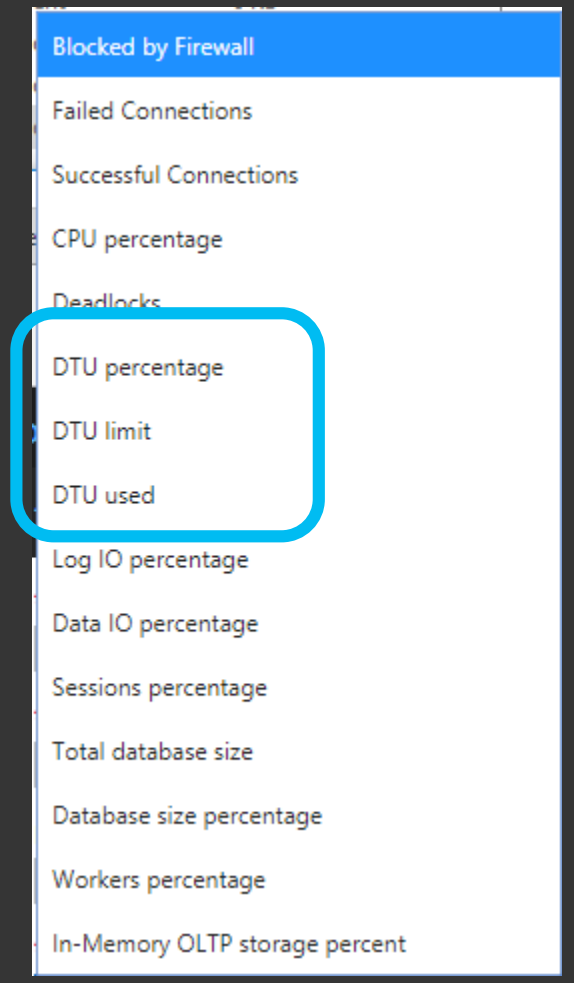
Monitoring your DTUs

DMVs

- **sys.dm_db_resource_stats**
  - Per database
  - Captures data every 15 seconds
  - Stored for one hour
  - Shows percentage used of allowed DTU limits for current tier
- **sys.resource_stats**
  - Stored in master database
  - Captures data every 5 minutes
  - Stored for 14 days
  - Shows percentage used of allowed DTU limits for current tier

Alert rules

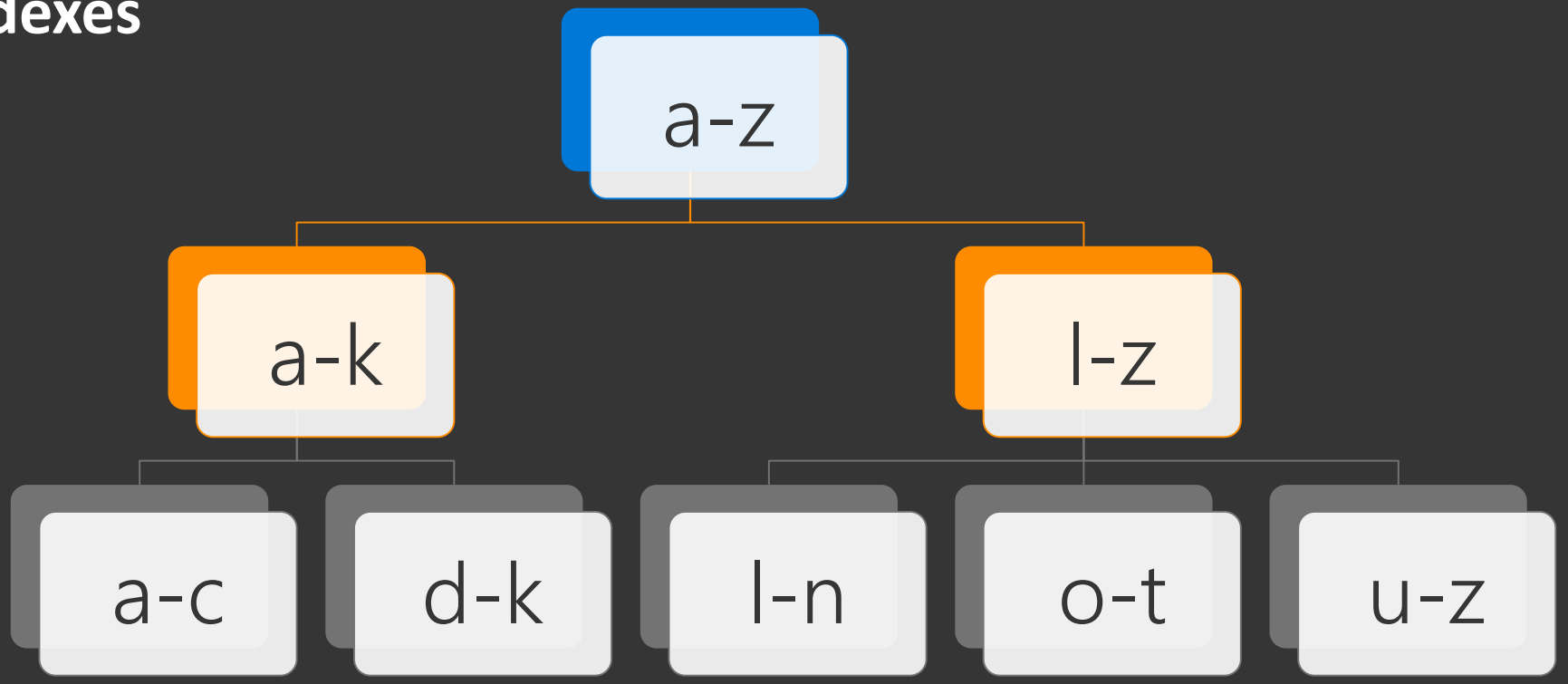
- **Monitor your databases - when set conditions are met, an email will be sent**
- **Portal**
  - ☐Choose a metric
  - ☐Set a condition - less than, equal to, greater than
  - ☐Set a threshold
  - ☐Pick a period of time
- **PowerShell**
  - ☐http://www.mikefal.net/2016/08/23/creating-alerts-for-azure-sql-database-with-powershell/



Indexes

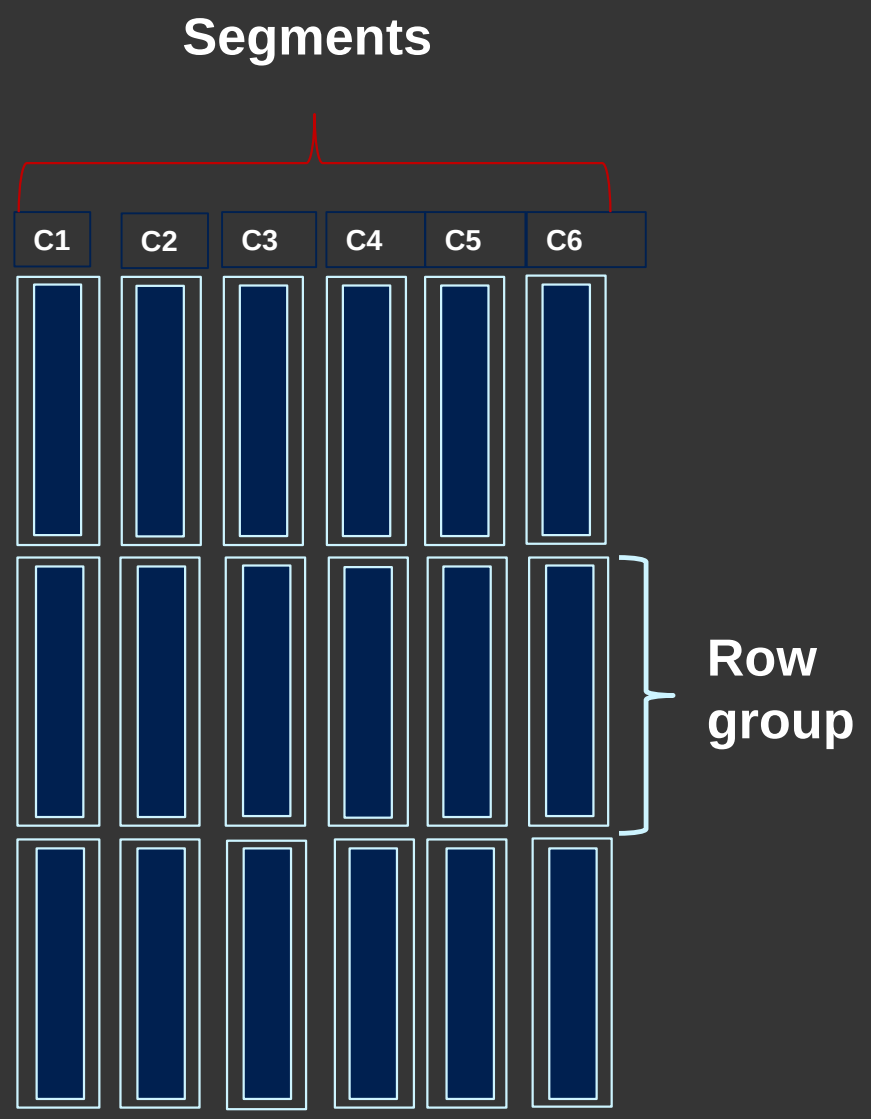
Rowstore indexes

- **Clustered indexes**
- **Nonclustered indexes**



Columnstore indexes

- **Available in Standard S3+ and Premium**
- **Not just an index, but a new way to store data**
- **Up to 10x data compression**
- **Up to 10x query performance in data warehouse scenarios**
  - Best for analytic queries searching large amounts of data



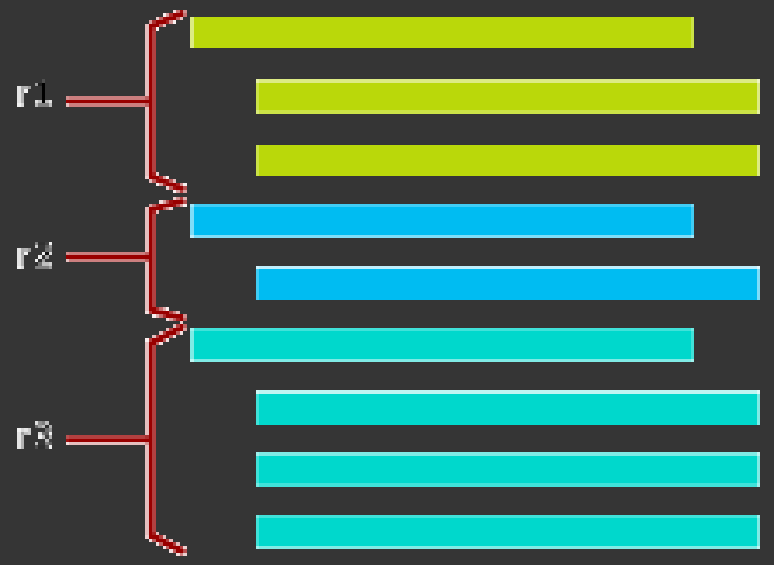
Columnstore flavors

- **Clustered columnstore index**
  - It is the data!
  - Can have nonclustered rowstore indexes built on it
- **Nonclustered columnstore index**
  - Built on top of a rowstore table

In-Memory OLTP

Memory is faster than disk

- **Premium tier only**
  - 1 GB storage for every 125 DTUs or eDTUs
- **Memory-optimized tables**
  - Schema or schema + data
  - No locking
  - Great for high-write workloads
- **Natively compiled T-SQL modules**
  - Designed to work with memory-optimized tables for best performance

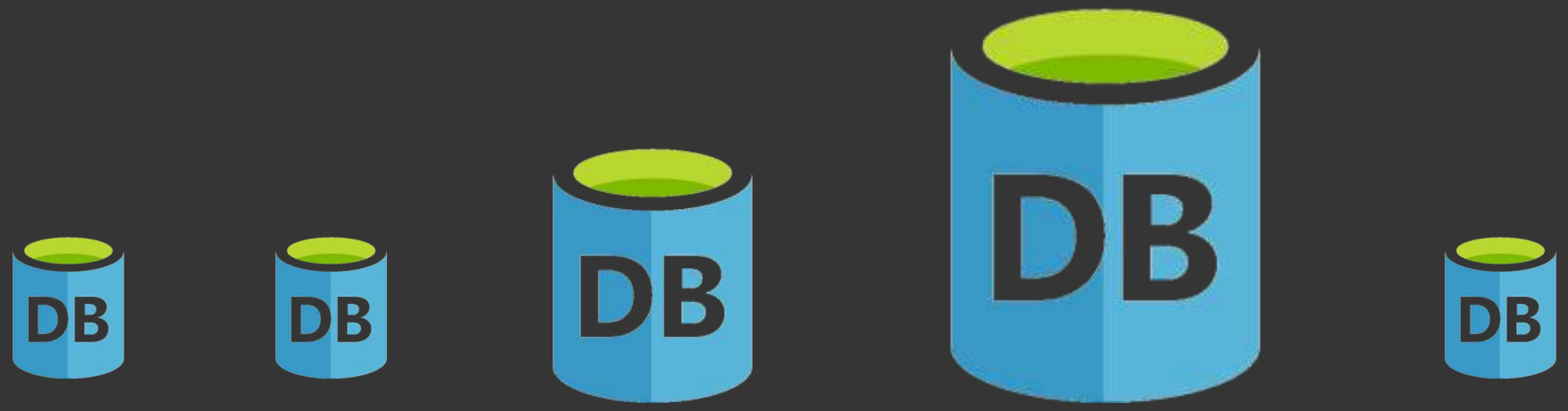


Operational Analytics

What is operational analytics?

- "An updateable columnstore index on a rowstore table or an in-memory table"
- Run analytic queries – quickly! – against your transactional database

Scaling


DB
DB
DB
DB
DB

How scaling works

- **Changing the service tier and/or performance level of a database creates a replica of the original database at the new performance level, and then switches connections over to the replica**
- **No data is lost during this process**
  - During the brief moment when we switch over to the replica, connections to the database are disabled, so some transactions in flight may be rolled back
- **The length of time for the switch over varies, but is generally under 4 seconds and is less than 30 seconds 99% of the time**
  - If there are large numbers of transactions in flight at the moment connections are disabled, the length of time for the switch over may be longer

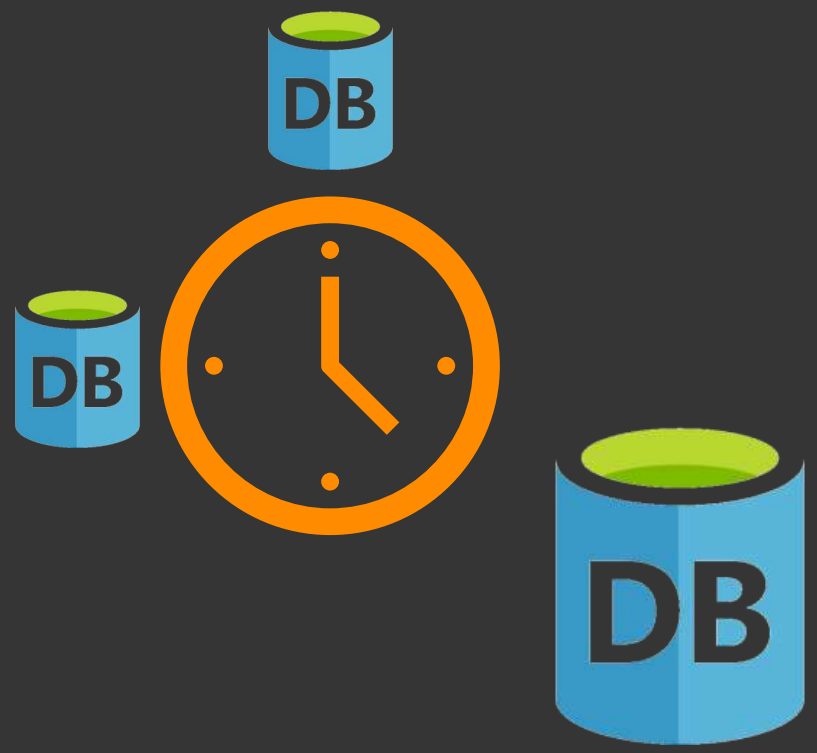
Scaling limitations

- **Scaling up**
  - If you upgrade to a higher tier or level, maximum database size doesn't change unless you specify it
  - If upgrading a database with DR (geo-replication) enabled, the recommendation is to upgrade the secondaries first
- **Scaling down**
  - The database size must not be larger than the maximum database size for the lower tier or level
  - When you downgrade to a lower tier, your DR options (such as backup retention period) may change, columnstore and in-memory may not be available

How to scale

- **Portal**
- **PowerShell**

- **Manually**
- **Scheduled**



Query Performance Insight

Query Performance Insight

- **A real-time view of how queries are affecting your database**
- **Requires Query Store be enabled (it is by default)**
- **View top resource-consuming queries**
  - CPU, data IO, log IO, duration, execution count
- **View long-running queries**
- **Get Performance Recommendations for a query**

Performance recommendations provided

- **Create index**
- **Drop index**
- **Parameterize queries**
  - Enables forced parameterization on the database
- **Fix schema issues**

Recommendations

ACTION	RECOMMENDATION DESCRIPTION		IMPACT
CREATE INDEX	Table: Indexed columns:	[test_table_0.430709] [index_1],[index_2],[index_3]	HIGH IMPACT
CREATE INDEX	Table: Indexed columns:	[test_table_0.914675] [index_1],[index_2],[index_3]	HIGH IMPACT
DROP INDEX (PREVIEW)	Index name: Reason:	IR_[test_schema]_[test_table_0.112348]_CD2E5085881888FC9A4 Duplicate index	HIGH IMPACT
DROP INDEX (PREVIEW)	Index name: Reason:	IR_[test_schema]_[test_table_0.950691]_9A67D9E88A31B315D14 Duplicate index	HIGH IMPACT
FIX SCHEMA ISSUES (PREVIEW)	Error code: Error message:	208 Invalid object name 'dbo.Companies'.	HIGH IMPACT

Real life

- **Using QPI to identify high data usage in Elastic Pool – Jim Donahoe - http://sqlflipflopsdba.com/2017/10/01/using-qpi-to-identify-high-data-usage-in-elastic-pool/**
- **Customer wanted to move from Premium to Standard Elastic Pool**
  - Only in preview!
- **Performance tune existing databases!**
- **Found the top 5 resource-consuming databases**
- **Identified highest data and log I/O queries**
- **Tuning top 3 queries reduced I/O by 10 billion reads**
  - Yes, BILLION!

Query Performance Insight

Demo

Automatic Tuning

Automatic Tuning

- **Executed queries are monitored for improvements; improvements are applied and measured**
- **Automatic index management**
  - Creates useful indexes
  - Drops duplicate or unused indexes
  - If improvement isn't significant, actions are reverted
- **Automatic plan choice correction**
  - If plan regression is detected, the database will switch to the last known good plan for that query

Azure SQL Database built-in intelligence automatically tunes your databases to optimize performance. Click here to learn more about automatic tuning.

Inherit from:

Server | Azure defaults | Don't inherit

The database is inheriting automatic tuning configuration from the server. You can set the configuration to be inherited by going to: Server tuning settings

The database is inheriting settings from the server, but the server is in the unspecified state. Please specify the automatic tuning state on the server.

Configure the automatic tuning options

OPTION	DESIRED STATE	CURRENT STATE
FORCE PLAN	ON OFF INHERIT	OFF Inherited from server
CREATE INDEX	ON OFF INHERIT	ON Inherited from server
DROP INDEX	ON OFF INHERIT	ON Inherited from server

Summary

Azure SQL Database

- You pay for performance!
- Monitor your database for DTU usage because…you pay for performance!
- Use rowstore & columnstore indexes to optimize query processing
- Use In-Memory OLTP if a workload would benefit from no locks
- Scale your database up or down as needed
- Use Query Performance Insight to identify expensive queries
- Enable Automatic Tuning to fine-tune indexes and execution plans

Jes Borland

Premier Field Engineer
Microsoft
jes.borland@microsoft.com
@grrl_geek
LessThanDot.com