
sqlbits
2026
FABRIC FINOPS
Cost Optimization for
Microsoft Fabric
Alexander Arvidsson | @arcticdba.se
alexander@arcticdba.se


EURO
200
EURO
50

Quick **Capacity Units** overview

Cost **drivers** in Microsoft Fabric

Practical ways to **optimize** CU usage and spend


Erbach
Senden
Vöhringen
Illertissen
Weißenhorn
Krumbach (Schwaben)
Babenhausen
Bobingen
Schwabmünchen
Mindelheim
Türkheim
Buchloe
Landsberg a.L.
Ochsenhausen
Boos
Heimertingen
Memmingen
Ungerhsn.
Sontheim
Bad Wörishofen
Ottobeuren
Markt Rettenbach
Kaufbeuren
Friesenried
Bad Wurzach
Aichstetten
Leutkirch im Allgäu
Bad Grönenbach
Allgäuer Tor
Untrasried
Obergünzburg
Markt-oberdorf
Altenstadt
Dietmannsried
Kraftisried
Kempten (Allg.)
Betzigau
Isny i.Allgäu
Wangen im Allgäu
Buchenberg
Durach
Görisried
Lengenwang
Roßhaupten
Nesselwang
Pfronten
Füssen
Immenstadt i.Allg.
Sonthofen
Oberstdorf
Linderhof
Mittenwald
Wallgau
Achenkirch
Großer Ahornboden
Karwendelhaus
Pürschhaus
Kißlegg
Ellwangen
Klosterkirche
Germaringen
Osterzell
Wessobrunn
Rott
Steingaden
Wieskirche
Halblech
Schwangau
Bad Kohlgrub
Saulgrub
Ettal
Farchant
Aitrach
Legau
Wiggensbach
Stötten a.Auerbg.
Auerberg
Lechbruck
Burggen
Wertach
Blaichach
Simmerberg
Oberstaufen
Sibratshofen
Heimenkirch


Alexander Arvidsson
(he/him)
Lead Data Transformation Architect@
Advania
Trainer | Speaker
advania
KNEE-DEEP
IN
TECH
MVP
Microsoft
Most Valuable
Professional

CAPACITY UNITS

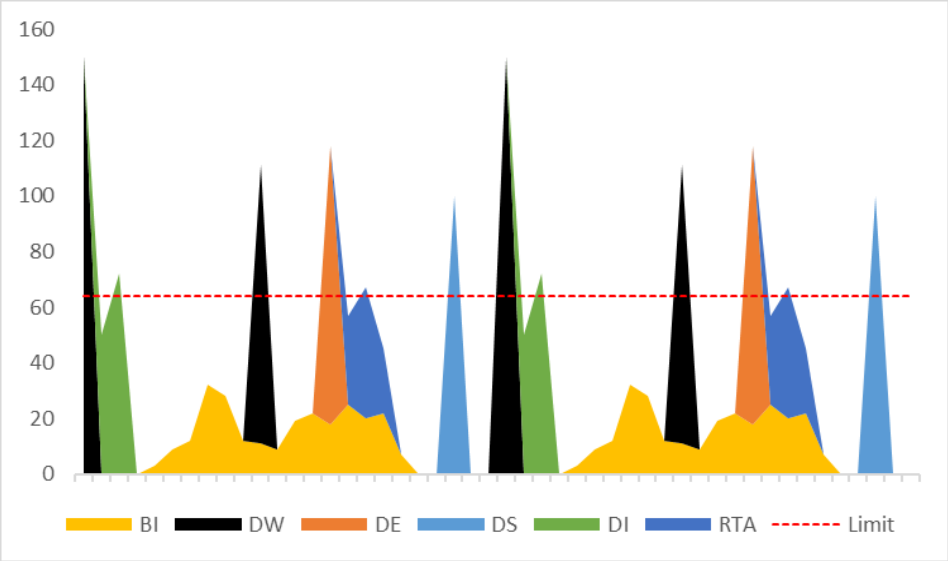

Fabric CU’s
are easy
to
understand

Bursting and smoothing | before and after

Looking at an example of a 64 CU capacity, running multiple

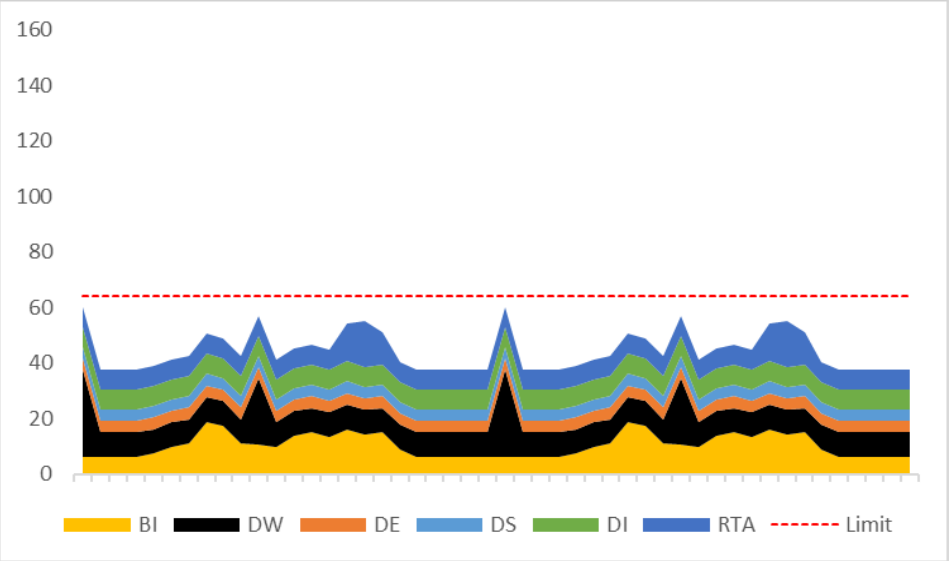
Before Smoothing

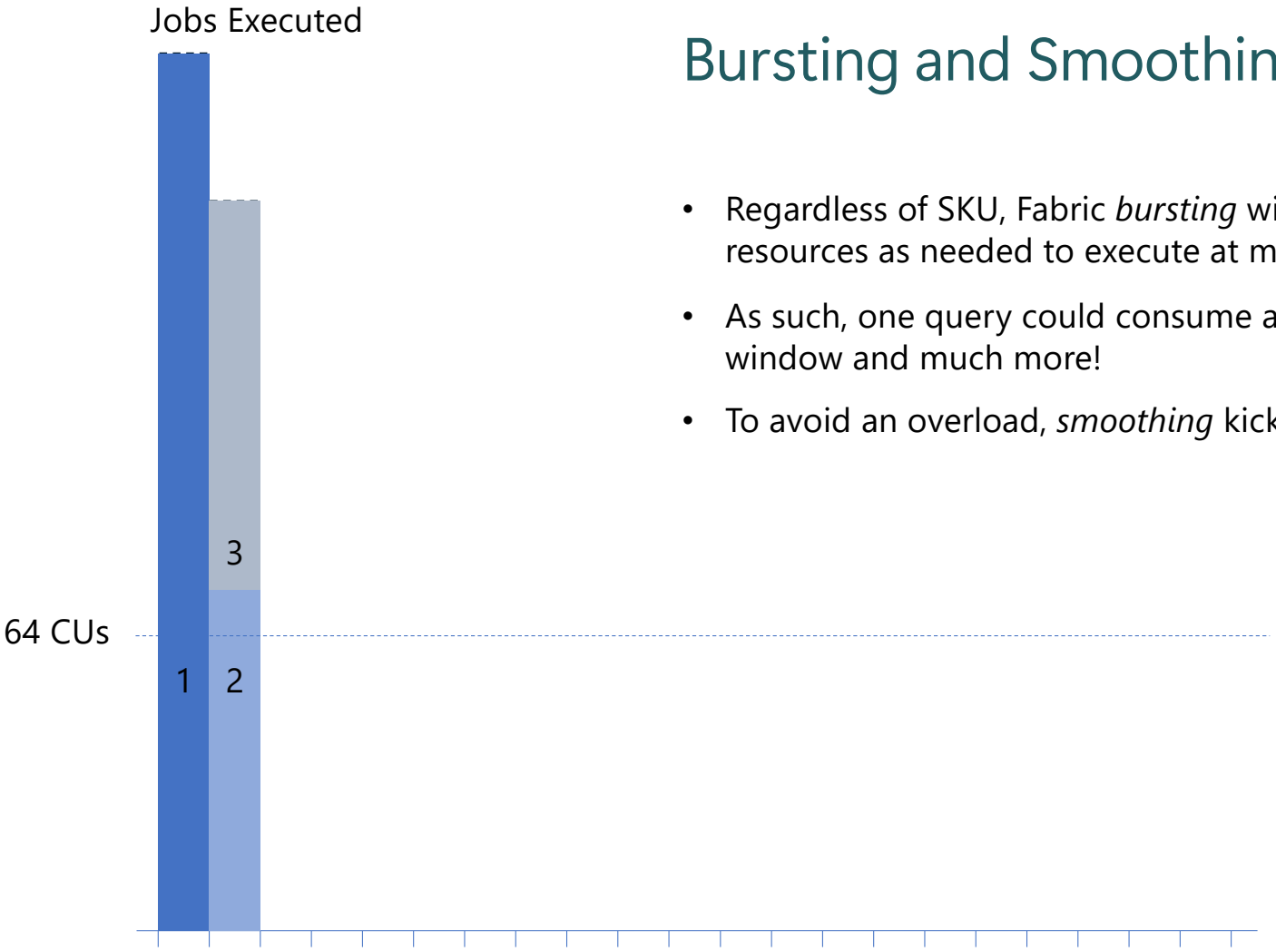
- Actual load as executed on the capacity before smoothing
- *Bursting* accelerates jobs execution by resource boosting
- The capacity could be overloaded 25% of the time
- Some of the overloads are more than 2x the limit
- There are periods of no/low usage



After Smoothing

- Shows the reported load (not runtime execution) against the capacity limits
- There is NO overload, and consumption is more stable
- The smoothing of usage fills in gaps

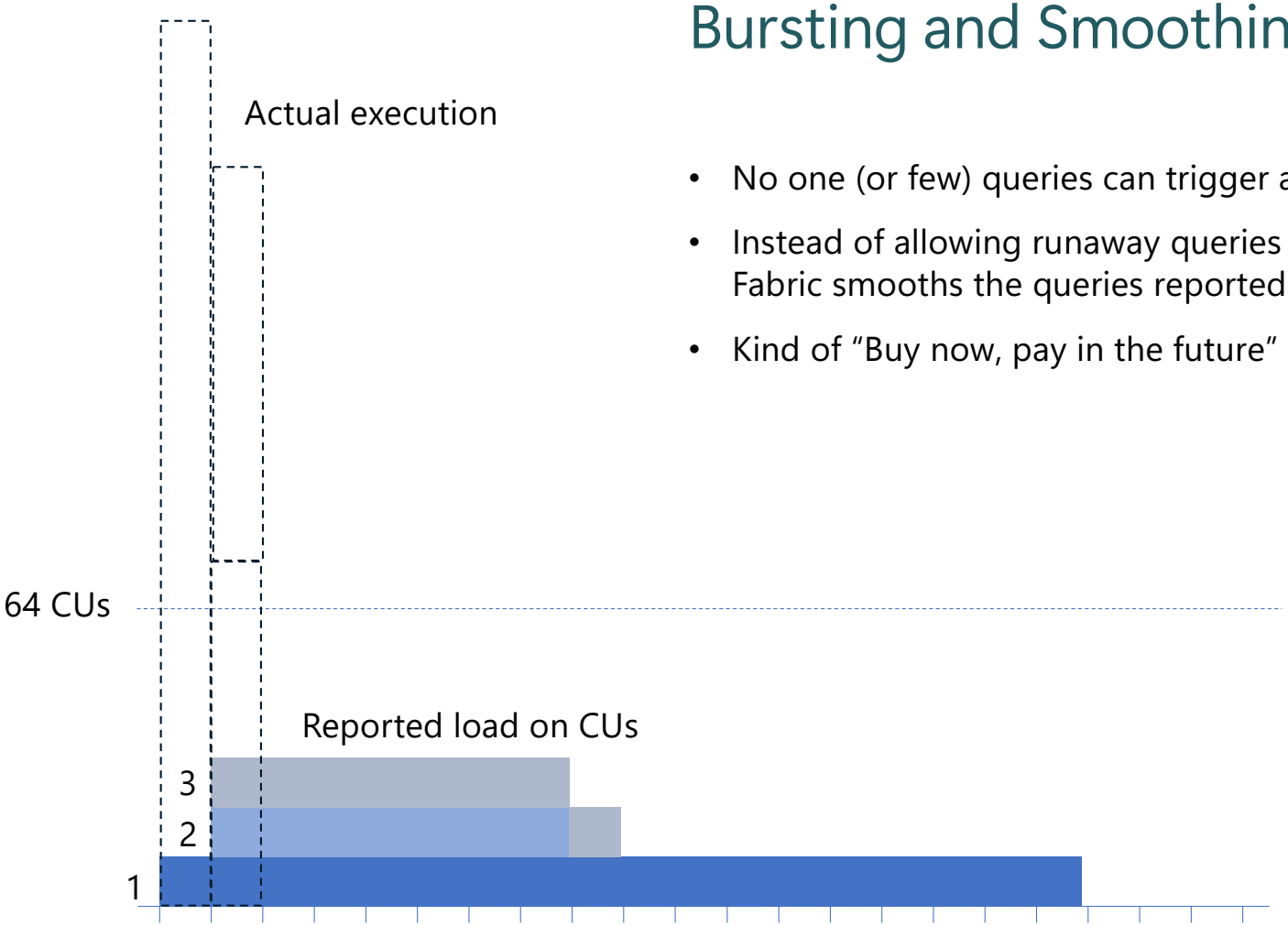




Bursting and Smoothing

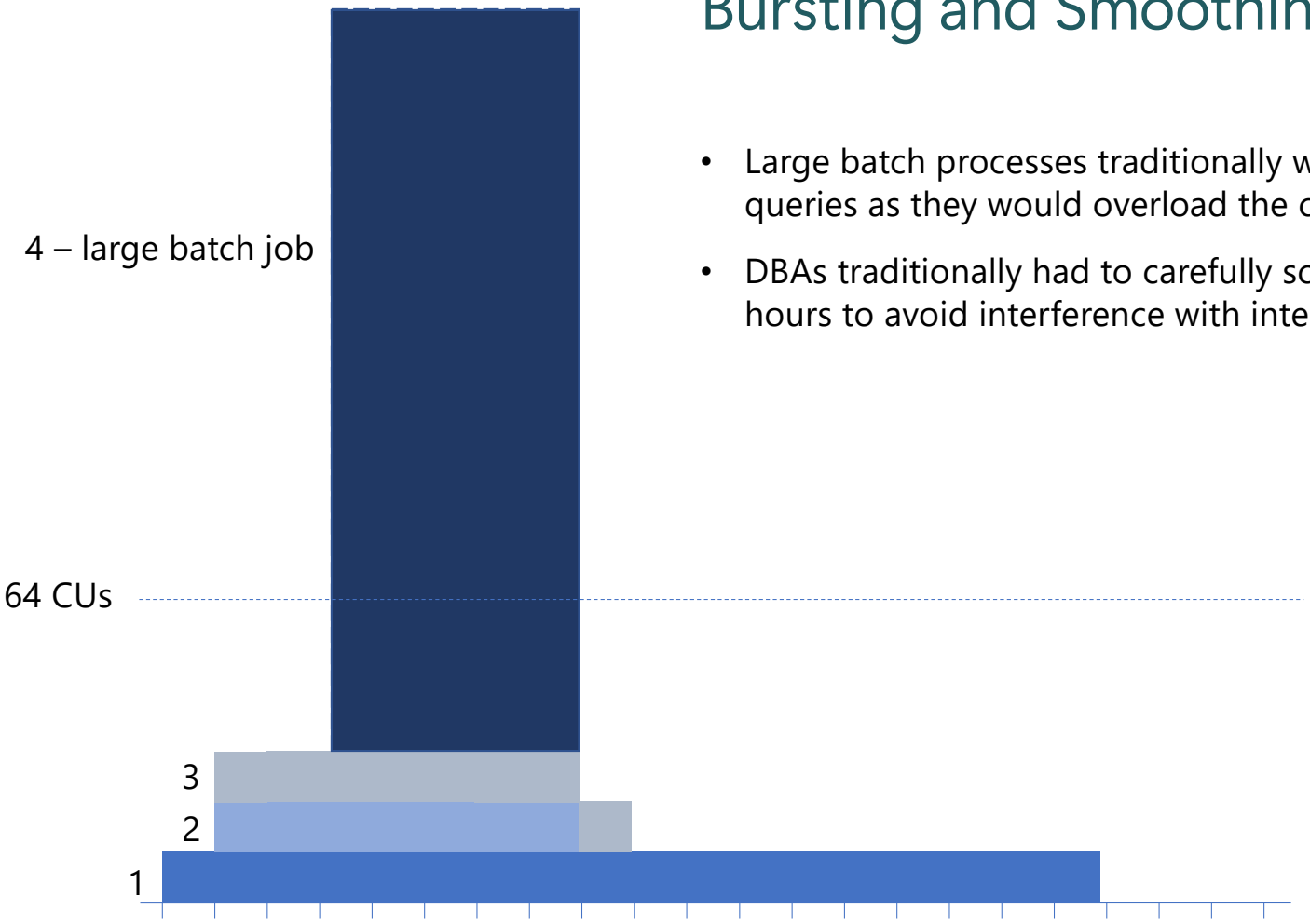
- Regardless of SKU, Fabric *bursting* will automatically allocate resources as needed to execute at maximum performance
- As such, one query could consume all the quota of a single time window and much more!
- To avoid an overload, *smoothing* kicks in

Bursting and Smoothing



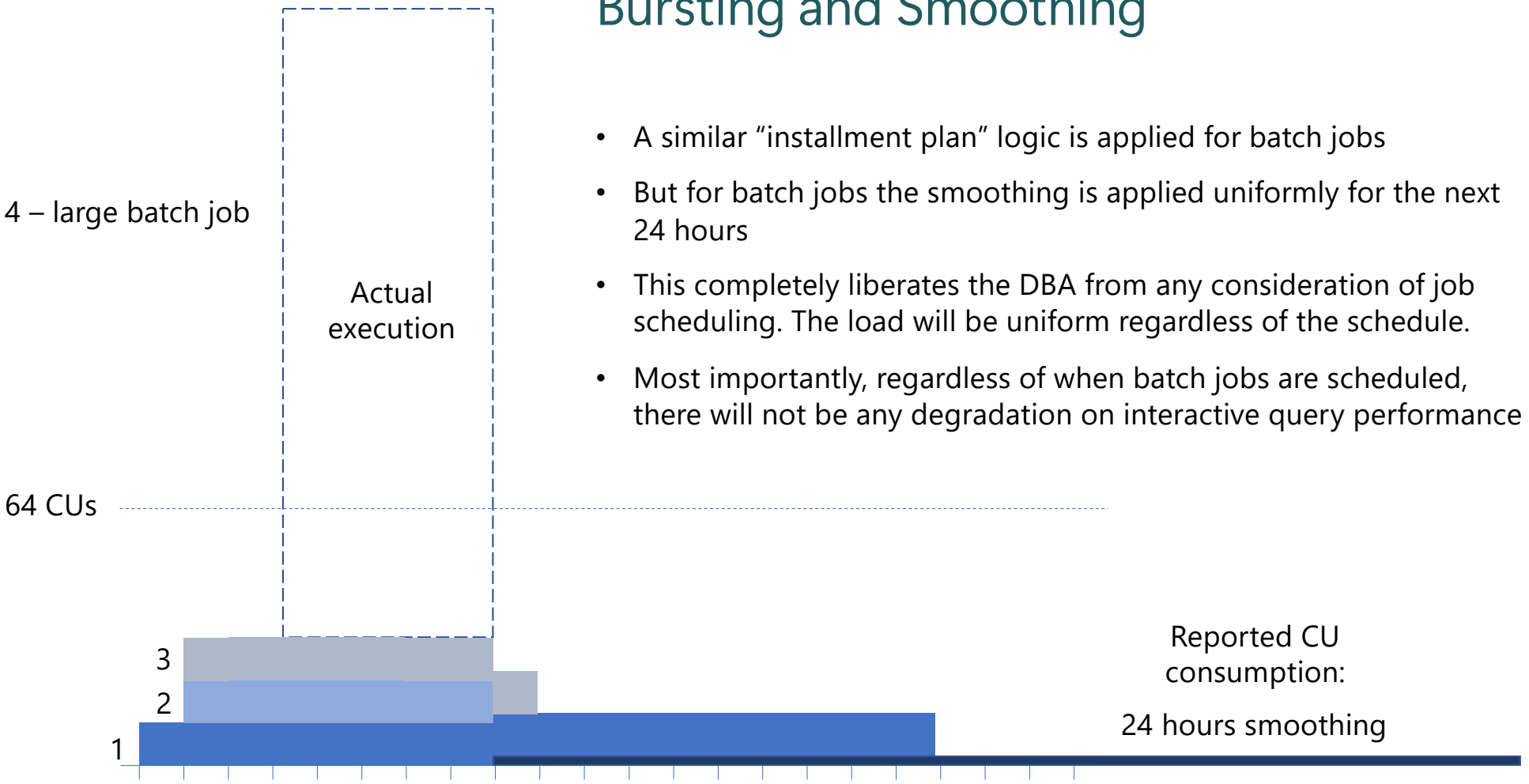
- No one (or few) queries can trigger an overload
- Instead of allowing runaway queries to create a local overload, Fabric smooths the queries reported usage to future time windows
- Kind of “Buy now, pay in the future” installment plan

Bursting and Smoothing



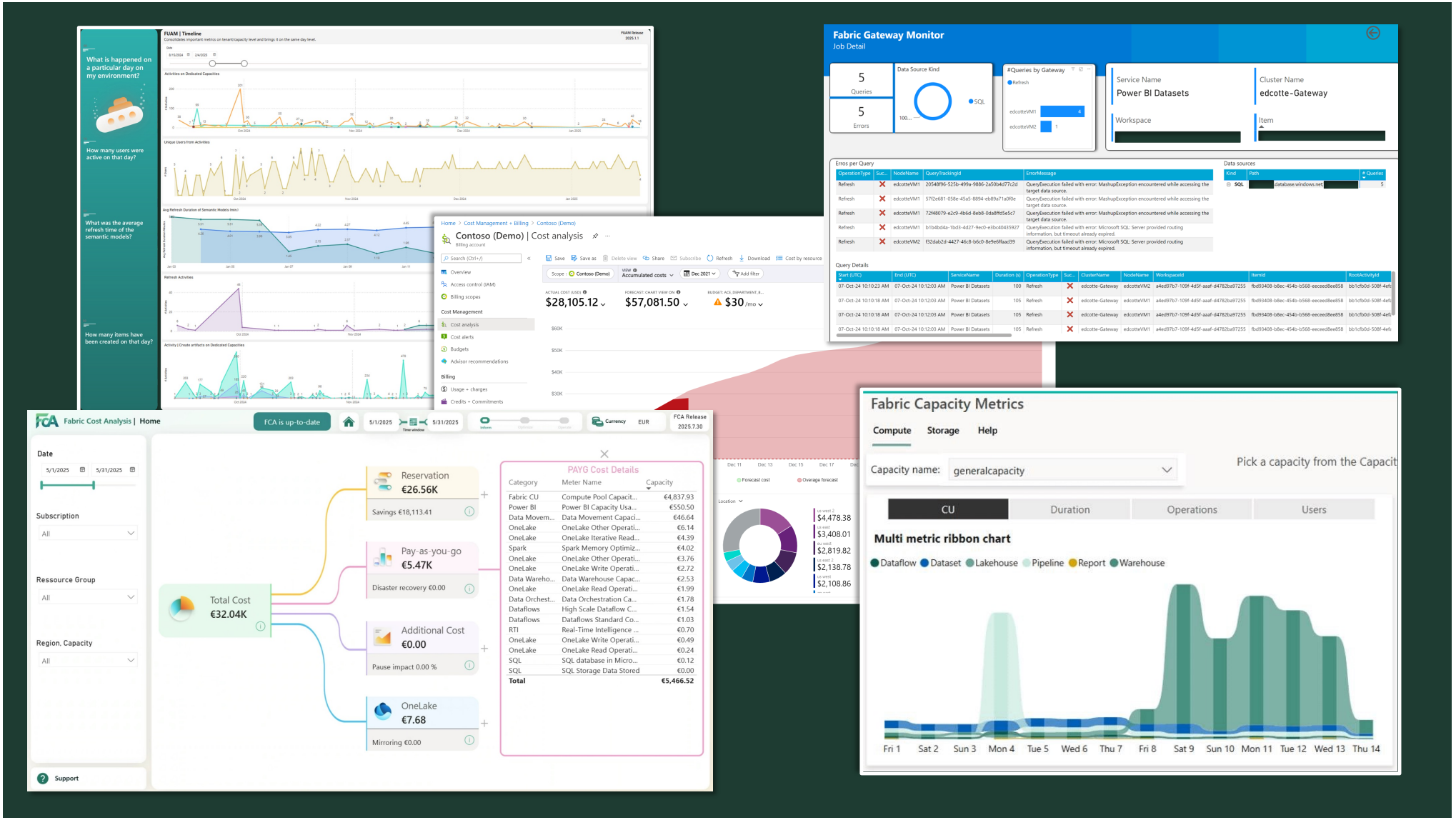
- Large batch processes traditionally were a threat to interactive queries as they would overload the compute resource
- DBAs traditionally had to carefully schedule these jobs to off-hours to avoid interference with interactive user experiences

Bursting and Smoothing



- A similar “installment plan” logic is applied for batch jobs
- But for batch jobs the smoothing is applied uniformly for the next 24 hours
- This completely liberates the DBA from any consideration of job scheduling. The load will be uniform regardless of the schedule.
- Most importantly, regardless of when batch jobs are scheduled, there will not be any degradation on interactive query performance

SKU	Capacity unit (CU)	Pay-as-you-go*	Reservation (~41% savings)
F2	2	**$262.80**/month	**$156.334**/month
F4	4	**$525.60**/month	**$312.667**/month
F8	8	**$1,051.20**/month	**$625.334**/month
F16	16	**$2,102.40**/month	**$1,250.667**/month
F32	32	**$4,204.80**/month	**$2,501.334**/month
F64	64	**$8,409.60**/month	**$5,002.667**/month
F128	128	**$16,819.20**/month	**$10,005.334**/month
F256	256	**$33,638.40**/month	**$20,010.667**/month
F512	512	**$67,276.80**/month	**$40,021.334**/month
F1024	1,024	**$134,553.60**/month	**$80,042.667**/month
F2048	2,048	**$269,107.20**/month	**$160,085.334**/month


FUAM | Timeline
What is happened on a particular day on my environment?
How many users were active on that day?
What was the average refresh time of the semantic models?
How many items have been created on that day?
Fabric Gateway Monitor
Job Detail
5
Queries
5
Errors
Data Source Kind
SQL
#Queries by Gateway
Refresh
edcotteVM1
edcotteVM2
Service Name
Power BI Datasets
Cluster Name
edcotte-Gateway
Workspace
Item
Erros per Query
Data sources
Query Details
Contoso (Demo) | Cost analysis
Billing account
$28,105.12
$57,081.50
$30 /mo
Fabric Cost Analysis | Home
FCA is up-to-date
5/1/2025
5/31/2025
Currency
EUR
FCA Release
2025.7.30
Date
Subscription
Ressource Group
Region, Capacity
Support
Reservation
€26.56K
Savings €18,113.41
Pay-as-you-go
€5.47K
Disaster recovery €0.00
Additional Cost
€0.00
Pause impact 0.00 %
OneLake
€7.68
Mirroring €0.00
Total Cost
€32.04K
PAYG Cost Details
Category
Meter Name
Capacity
Fabric CU
Compute Pool Capacit...
€4,837.93
Power BI
Power BI Capacity Usa...
€550.50
Data Movem...
Data Movement Capaci...
€46.64
OneLake
OneLake Other Operati...
€6.14
OneLake
OneLake Iterative Read...
€4.39
Spark
Spark Memory Optimiz...
€4.02
OneLake
OneLake Other Operati...
€3.76
OneLake
OneLake Write Operati...
€2.72
Data Wareho...
Data Warehouse Capac...
€2.53
OneLake
OneLake Read Operati...
€1.99
Data Orchest...
Data Orchestration Ca...
€1.78
Dataflows
High Scale Dataflow C...
€1.54
Dataflows
Dataflows Standard Co...
€1.03
RTI
Real-Time Intelligence ...
€0.70
OneLake
OneLake Write Operati...
€0.49
OneLake
OneLake Read Operati...
€0.24
SQL
SQL database in Micro...
€0.12
SQL
SQL Storage Data Stored
€0.00
Total
€5,466.52
Fabric Capacity Metrics
Compute
Storage
Help
Capacity name:
generalcapacity
Pick a capacity from the Capacit
CU
Duration
Operations
Users
Multi metric ribbon chart
Dataflow
Dataset
Lakehouse
Pipeline
Report
Warehouse
Fri 1
Sat 2
Sun 3
Mon 4
Tue 5
Wed 6
Thu 7
Fri 8
Sat 9
Sun 10
Mon 11
Tue 12
Wed 13
Thu 14

THE WORKLOAD MENU

Data Pipelines Engine Type	Charge Meters and Metric Units	Fabric Capacity Units (CU) consumption rate
Data movement	Based on Copy activity run duration in hours and the used intelligent optimization throughput resources	1.5 CU hours
Data orchestration	Incorporates orchestration activity runs and activity integration runtime charges	0.0056 CU hours for each non-copy activity run

Copy Patterns	Consumption Meters	Fabric Capacity Units (CU) consumption rate	Consumption reporting granularity
Full copy	Data movement	1.5 CU hours	Per Copy job item
Incremental copy	Data movement – incremental copy	3 CUs hours	Per Copy job item

Apache Airflow job size (Base)	Consumption Meters	Fabric CU consumption rate	Consumption reporting granularity
Small	DataWorkflow Small	5 CUs	Per Apache Airflow job item
Large	DataWorkflow Large	10 CUs	Per Apache Airflow job item

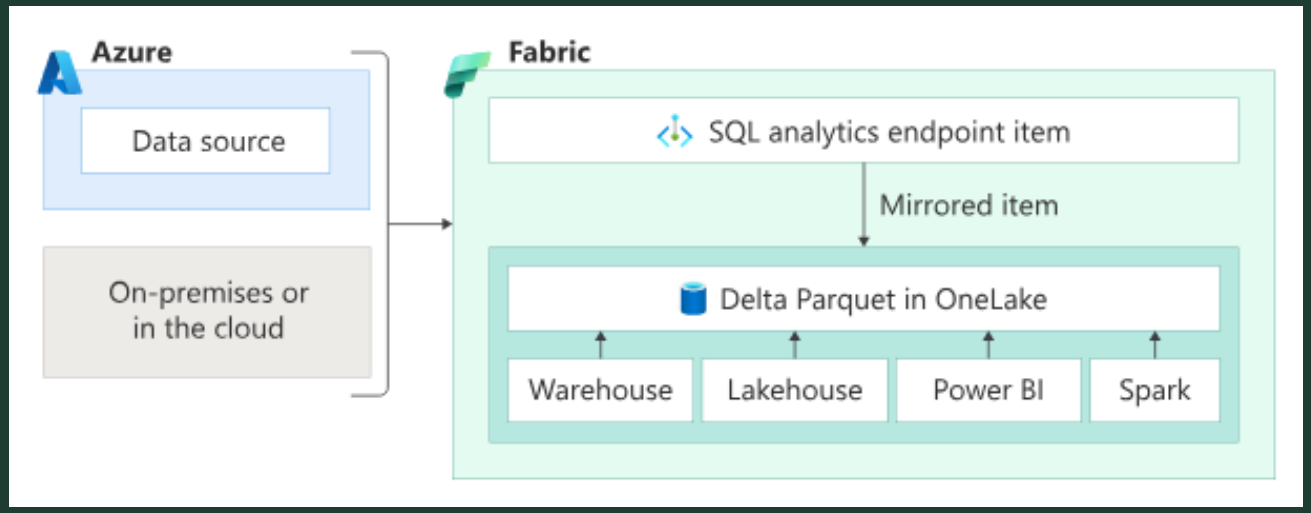
Dataflow Gen2 Engine Type	Consumption Meters	Fabric CU consumption rate	Consumption reporting granularity
Standard Compute (Dataflow Gen2 (CI/CD))	Based on each mashup engine query execution duration in seconds. Standard Compute has two tier pricing depending on the query duration.	- For every second up to 10 minutes, 12 CU - For every second beyond 10 minutes, 1.5 CU	Per Dataflow Gen2 item
Standard Compute (non CI/CD)	Based on each mashup engine query execution duration in seconds.	16 CU	Per Dataflow Gen2 item
High Scale Dataflows Compute	Based on Lakehouse/Warehouse SQL engine execution (with staging enabled) duration in seconds.	6 CU	Per workspace
Data movement	Based on Fast Copy run duration in hours and the used intelligent optimization throughput resources.	1.5 CU	Per Dataflow Gen2 item

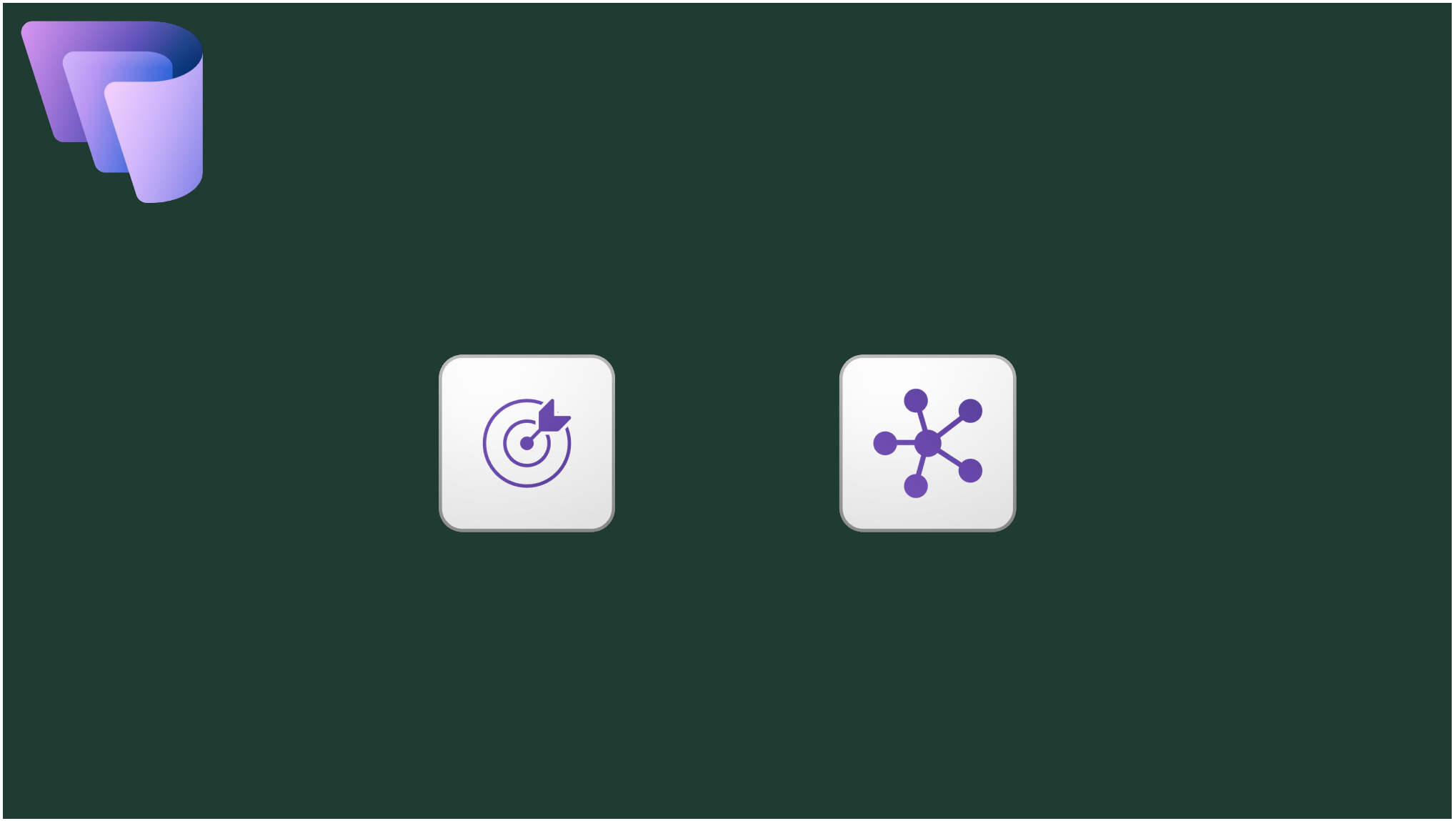

APACHE
Spark
TM

Operation name	CU (s)	Duration (s)	Users	Billing type
Warehouse Query	2,187.77	1,360.42	5	Billable
OneLake Compute	0.01	11,880.00	1	Billable
Total	**2,187.78**	**13,240.42**	**6**	

Operation name	CU (s)	Duration (s)	Users	Billing type
SQL Endpoint Query	4,086.87	1,503.81	6	Both
Total	**4,086.87**	**1,503.81**	**6**	

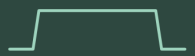
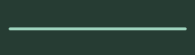
Operation name	CU (s)	Duration (s)	Users	Billing type
KustoUpTime	21,133,800.0000	860,460.0000	1	Billable
Total	**21,133,800.0000**	**860,460.0000**	**1**	


Azure
Data source
On-premises or in the cloud
Fabric
SQL analytics endpoint item
Mirrored item
Delta Parquet in OneLake
Warehouse
Lakehouse
Power BI
Spark



Operation	Description	Operation Unit of Measure	Capacity Units
OneLake BCDR Read via Redirect	OneLake BCDR Read via Redirect	Every 4 MB, per 10,000	104 CU seconds
OneLake BCDR Write via Redirect	OneLake BCDR Write via Redirect	Every 4 MB, per 10,000	3056 CU seconds
OneLake BCDR Iterative Read via Redirect	OneLake BCDR Iterative Read via Redirect	Per 10,000	1626 CU seconds
OneLake BCDR Iterative Write via Redirect	OneLake BCDR Iterative Write via Redirect	Per 100	2730 CU seconds
OneLake BCDR Other Operations Via Redirect	OneLake BCDR Other Operations Via Redirect	Per 10,000	104 CU seconds


FUAM
Take your tenant monitoring
to the next level.
Tenant Meta Data
Activity Logs
Capacity
Refreshables
and more...

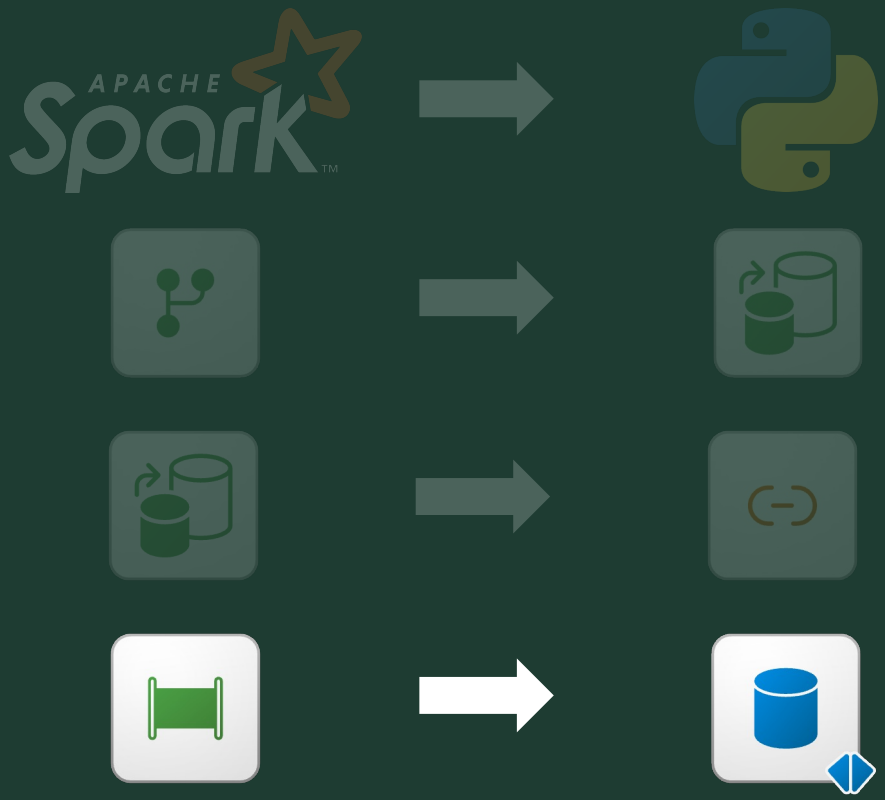
WORKLOAD	CU COST	PREDICTABILITY	SKILL REQ.		CONSUMPTION SHAPE
Dataflows Gen 2 Per minute of compute	High	Medium	Low		Sustained plateau
Pipelines & Copy Per activity / duration	Medium	High	Low		Activity spikes
Notebooks (PySpark) Per node-minute	High	Medium	High		Sustained plateau
Notebooks (Python) Per minute, no cluster	Low	Medium	High		Low and steady
Semantic model (Import) Per second, during refresh	Medium	High	Medium		Wide refresh burst
Semantic model (Direct Lake) Query-time only	Very Low	High	Medium		Near-zero, query bumps
SQL / Warehouse Per query · 1 vCore = 2 CU · 24h smoothing	Variable	Low	Medium		Irregular per-query
Fabric Database 2.611 CU/vCore · 5-min smoothing · always-on floor	Very High	High	Medium		Always-on + query spikes
Eventhouse / KQL Per query (efficient)	Low	Medium	Medium		Efficient per-query
Mirroring Continuous, near-zero	Very Low	High	Low		Continuous baseline


What gets
measured
SKU cost
Skills **availability**
Development **time**
Operational **burden**
Rework cost
Time to **value**
What **actually**
drives cost

	Copy job + notebook	Dataflows Gen 2
Who builds it	Senior data engineer	Business analyst
Hourly rate	$130	$80
Build time	40h	16h
Build cost	**$5,200**	**$1,280**
Year 1 CU cost	**$35**	**$141**
Year 1 total	**$5,235**	**$1,421**
Year 2 total	**$35**	**$141**


OPTIMIZATION

Compute substitutions



Spark configuration

Default cluster config

Starter pools

Node **count**

Spark **autoscale** billing

Storage optimization

Delta OPTMIZE & VACUUM

V-ordering

Partitioning **strategy**

Medallion storage **needs**

Semantic model architecture

Direct Lake **vs** Import

Aggregations

Refresh **interval**

Query efficiency

KQL **vs** SQL

DW **result set** caching

Incremental refresh

DAX measure **efficiency**

Smartness

Use Power BI Pro

BC/DR

Capacity **reservations**

Audit licenses

Find workload **cost** in CUs

Tie bursting events to **workloads**

Apply the right optimization – for **your** team

Reassess SKU headroom

Make the SKU decision **deliberately**

Upgrading is
sometimes right.

So is downgrading.

When did you last look at the Capacity Metrics app?

Did the numbers surprise you?

SUMMARY

The fixed price is real

What you get for it is up to you

CU cost is but one input

Skills, time, and risk are the others

Measure your consumption

You may not need the SKU you're paying for


Messenger
1
Instagram


THANK
YOU!
@arcticdba.se
Fabric FinOps - Cost Optimization for Fabric