A CLOUD GURU

Serverless Leaderboards

Dale Salter
SOFTWARE DEVELOPER LEAD

A CLOUD GURU

Who am I?

1. **Serverless Enthusiast**
   Using λ since it was released 2015, AWS for 4 years

2. **ACG Platform Developer**
   Billing, {Course, Series} Catalog, Transcoder

@enepture

Just because you can does not mean you should

~ Start Demo ~

Leaderboards

Solution should be able to accommodate for

A CLOUD GURU

1 **Faceted Search**

Should be able to drill down and provide scores for, **time**, **location**, **cloud service**, **organisation**

2 **Current**

The results should be able to be **streamed** and there should be a little delay between earning the score and having it **appear on a leaderboard**

3 **Performant**

Leaderboard should be able to be retrieved in a reasonable amount of time, this is going to imply some pre-computation

4 **Compare against previous periods**

It would be ideal to track where you were in a given month on a board, perhaps you were within the top 100 on the previous day. This could provide encouragement

The challenge

A CLOUD GURU

1 **Completely elastic**

Should be able to scale up from one user to millions of users in a reasonable amount of time*

2 **Pay per use**

Resources which are not needed used should not be costing money

3 **Serverless Technologies**

Low maintenance overhead, maintaining large Kafka / SQL clusters is not something we are good at

4 **Highly Available / Fault Tolerant**

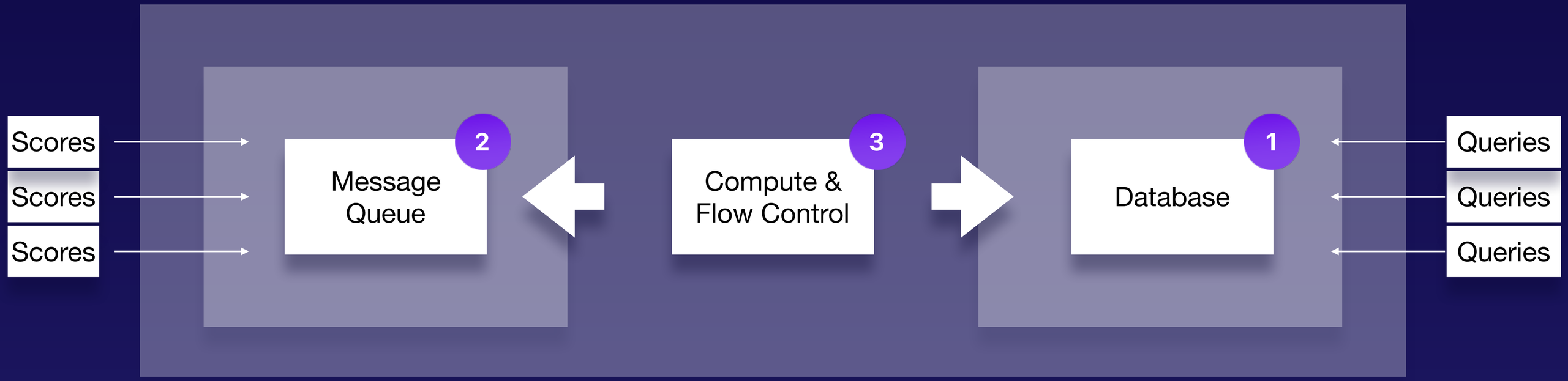
System should be able to continue working correctly without any performance degradation if an AZ goes down

Considerations

1. Application Maintained Materialised Views
2. Stream Based System
3. Pull Based System

Architecture

A CLOUD GURU



1 **Choice of Database**

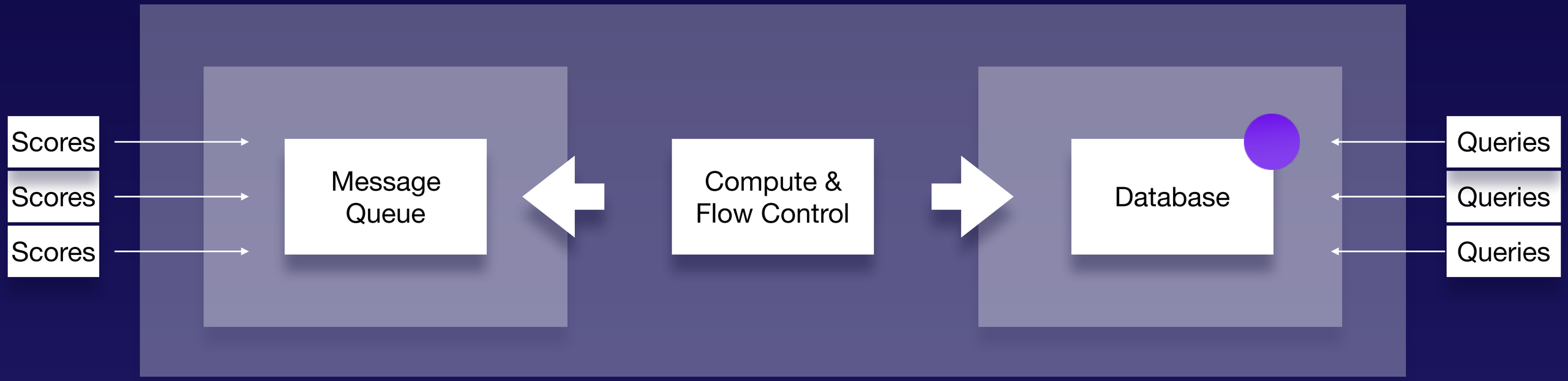
2 **Choice of Message Queue**

3 **Choice of Compute & Flow Control**

Database

Architecture

A CLOUD GURU



Why DynamoDB?

A CLOUD GURU

1. **Elastic**
   Scales up and down based on load, does not sit around idle. Does require turning to get high utilisation

2. **Serverless**
   Does not require any operations work, pay per use

3. **Predictable / Low latency**
   DynamoDB can typically perform inserts / reads in single digit latencies

4. **Upfront design**
   Limited querying capabilities means that it is important to design our key schema upfront

Partitioning Schema

A CLOUD GURU

1 Hot Partitions

These have to be considered when looking at read / write load into the database. Hard cardinality keys are a requirement

2 Query / Write Models

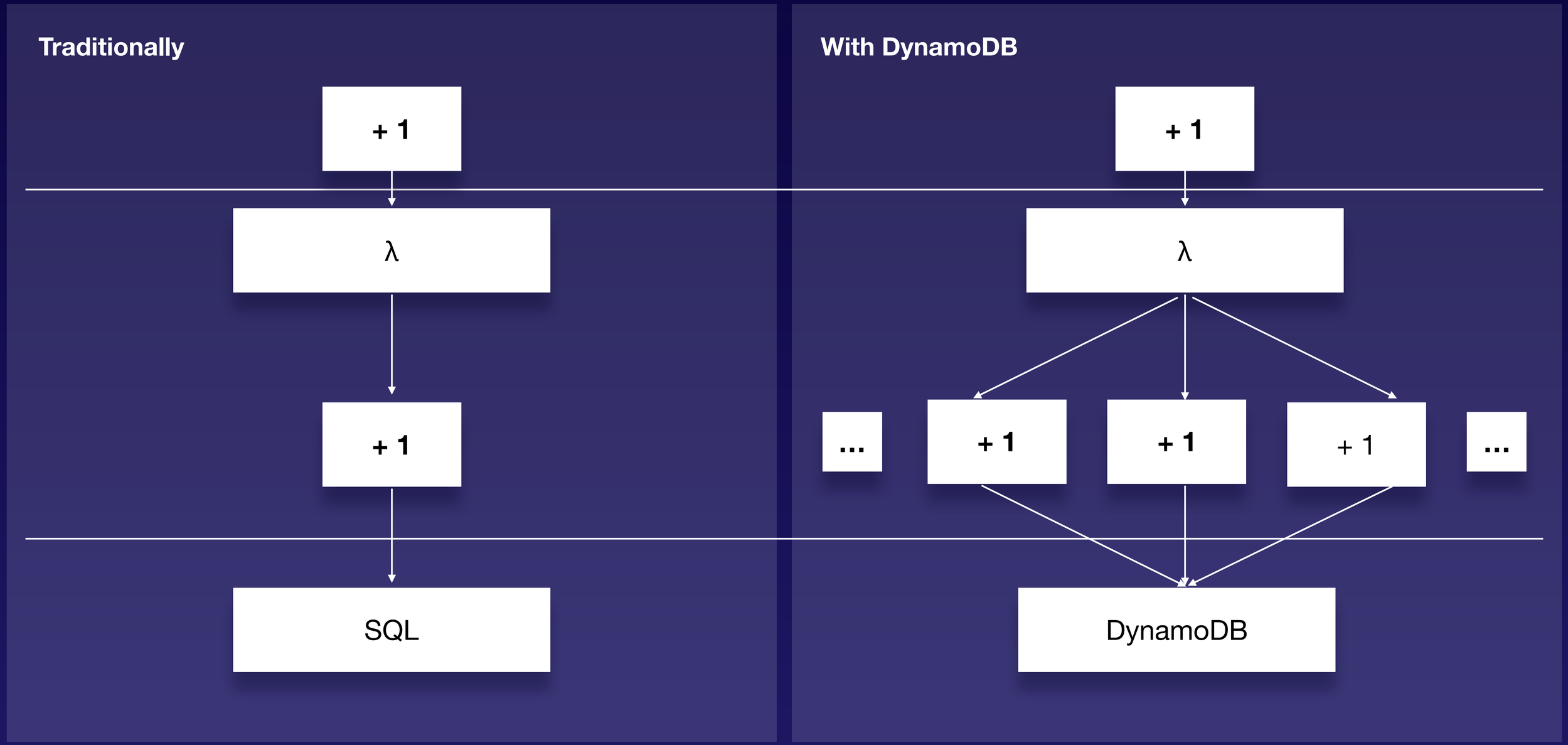
Base table has all information in it to update a score.

GSI contains all the information needed for querying the leaderboards

Can assign different throughputs on different parts of the table

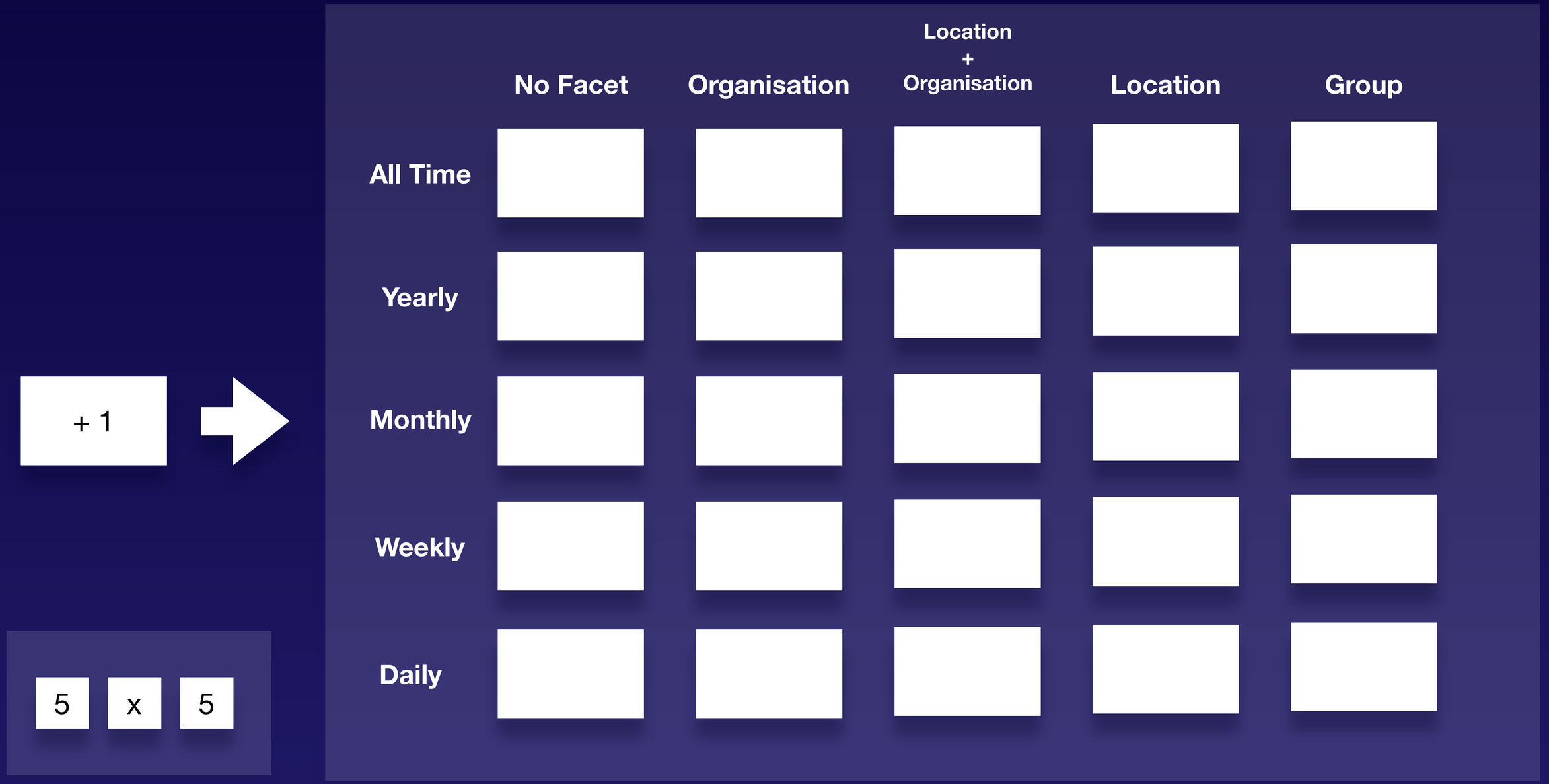
Materialised View Updates

A CLOUD GURU



Materialised View Updates

A CLOUD GURU



DynamoDB Basics - Partition Key + Sort Key

A CLOUD GURU

Partition Key	Sort Key	
Frank	7	3
Frank	6	3
Dale	3	2
Dale	3	2
Dale	3	2
Paul	2	1
Paul	1	1
Paul	1	1

Read / Write Model

A CLOUD GURU

Read / Write Model

A CLOUD GURU

Write Model

A CLOUD GURU

UserID (Partition)	Leaderboard (Sort)	Score
Dale	Yearly-2018	3
Dale	Yearly-2019	8
Dale	Monthly-2018/1	2
Dale	Monthly-2018/4	2
Dale	Monthly-2018/5	2
Dale	Monthly-2018/43	1
Dale	Monthly-2018/42	1
Dale	Monthly-2018/41	1

Read / Write Model

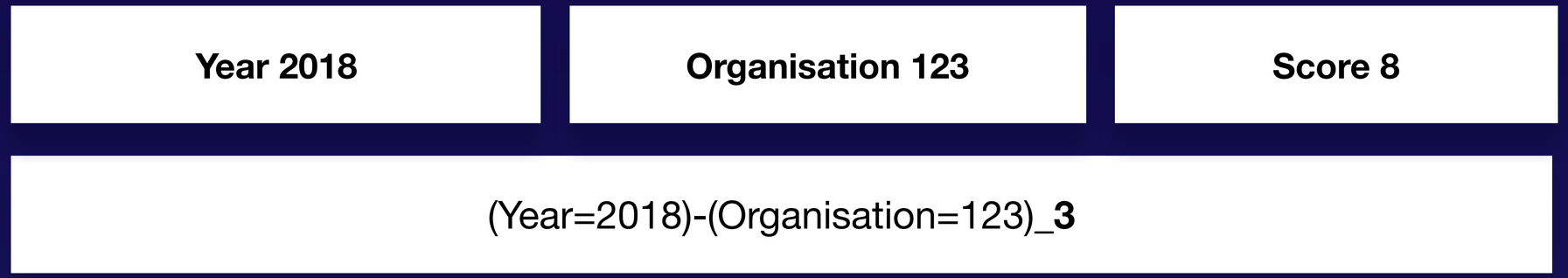
A CLOUD GURU

Query Model

A CLOUD GURU

Leaderboard (Partition)	Score (Sort)	
(Year=2018)-(org=123)	7	Bob
(Year=2018)-(org=123)	6	Frank
(Year=2018)-(org=123)	3	Fred
(Year=2018)-(org=123)	3	Dale
(Year=2018)-(org=123)	3	Sam
(Year=2018)-(org=123)	2	Paul
(Year=2018)-(org=123)	1	John
(Year=2018)-(org=123)	1	James

Read Model - Partitioning Scheme



Query Model - Partition split based off of score

A CLOUD GURU

Leaderboard (Partition)	Score (Sort)	
(Year=2018)-(org=123)_3	7	Bob
(Year=2018)-(org=123)_3	6	Frank
(Year=2018)-(org=123)_2	3	Fred
(Year=2018)-(org=123)_2	3	Dale
(Year=2018)-(org=123)_2	3	Sam
(Year=2018)-(org=123)_1	2	Paul
(Year=2018)-(org=123)_1	1	John
(Year=2018)-(org=123)_1	1	James

Write / Query Model

A CLOUD GURU

UserID (Partition)	Leaderboard (Sort)	
Dale	Year-2018	7
Dale	Year-2018	6
Dale	**Month-2018/1**	**6**
Dale	Month-2018/4	2
Dale	Month-2018/5	1
Dale	Week-2018/43	4
Dale	Week-2018/42	7
Dale	Week-2018/41	8

Leaderboard (Partition)	Score (Sort)	UserID
Month-2018/1_3	7	Frank
Month-2018/1_3	**6**	**Dale**
Month-2018/1_3	5	Sam
Month-2018/1_2	3	John
Month-2018/1_1	2	Fred
Month-2018/1_1	2	Jo
Month-2018/1_1	2	James
Month-2018/1_1	1	Jeff

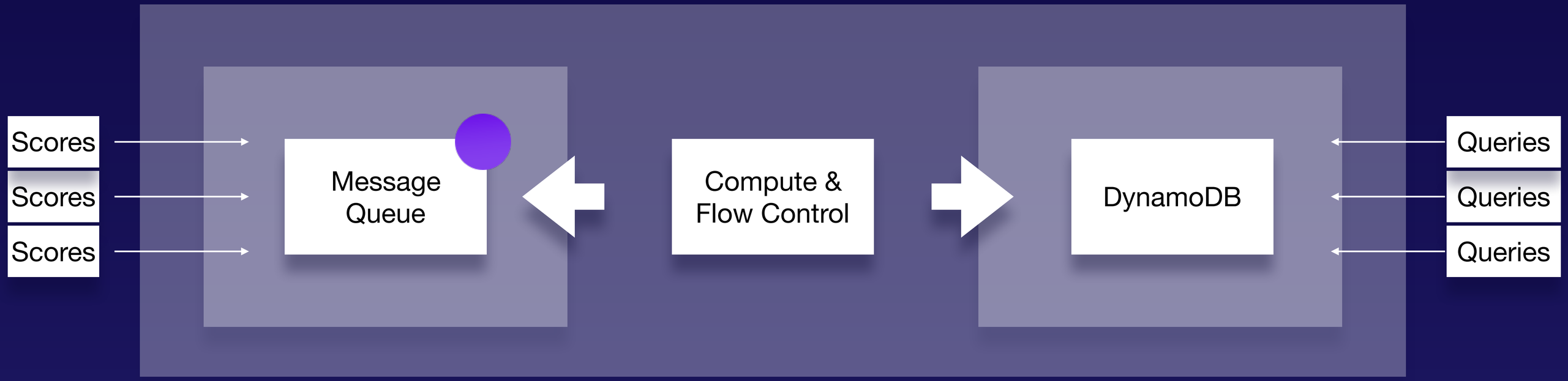
Read / Write Model

A CLOUD GURU

Message Queue

Architecture

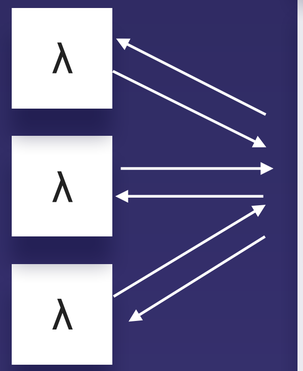
A CLOUD GURU



Optimistic Concurrency Problem?

Optimistic Concurrency

A CLOUD GURU



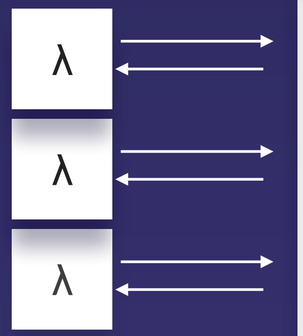
Bob	7	3
Frank	6	3

Fred	3	2
Dale	3	2
Sam	3	2

Paul	2	1
John	1	1
James	1	1

Optimistic Concurrency

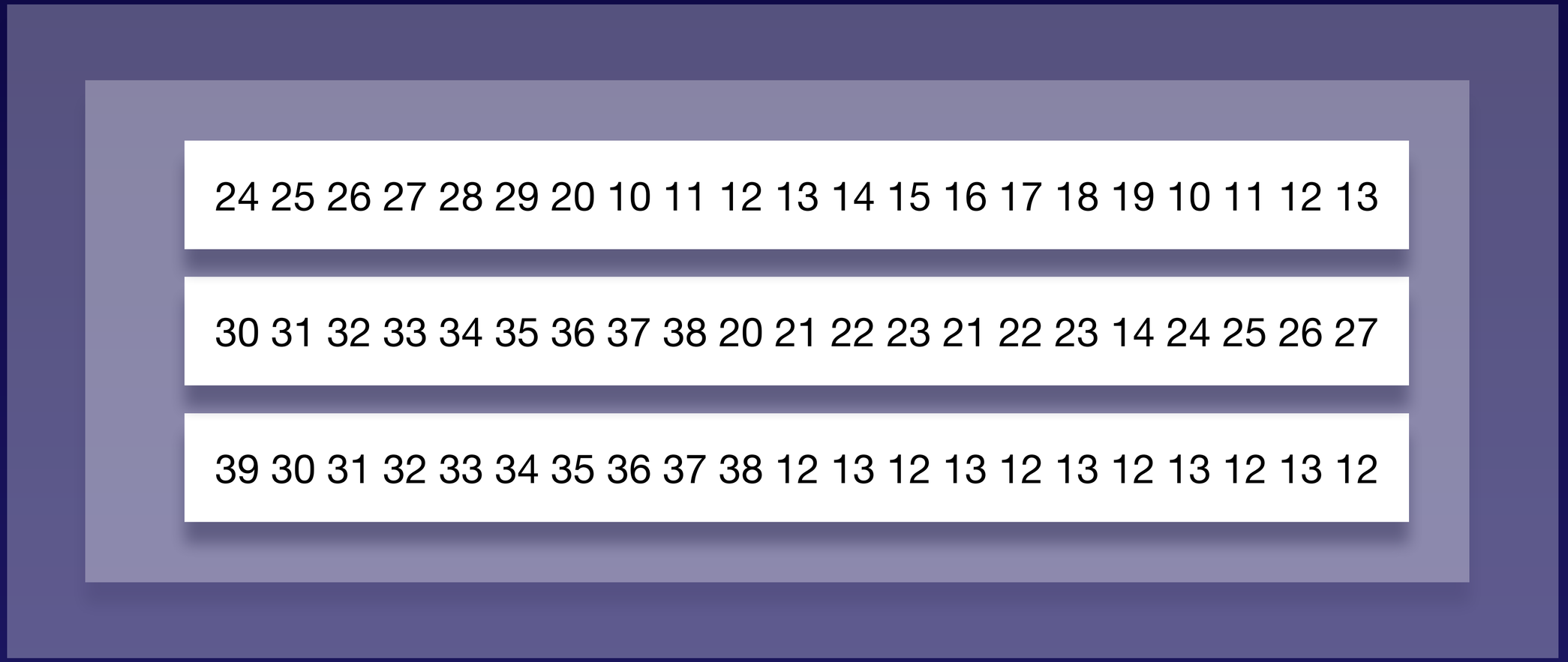
A CLOUD GURU



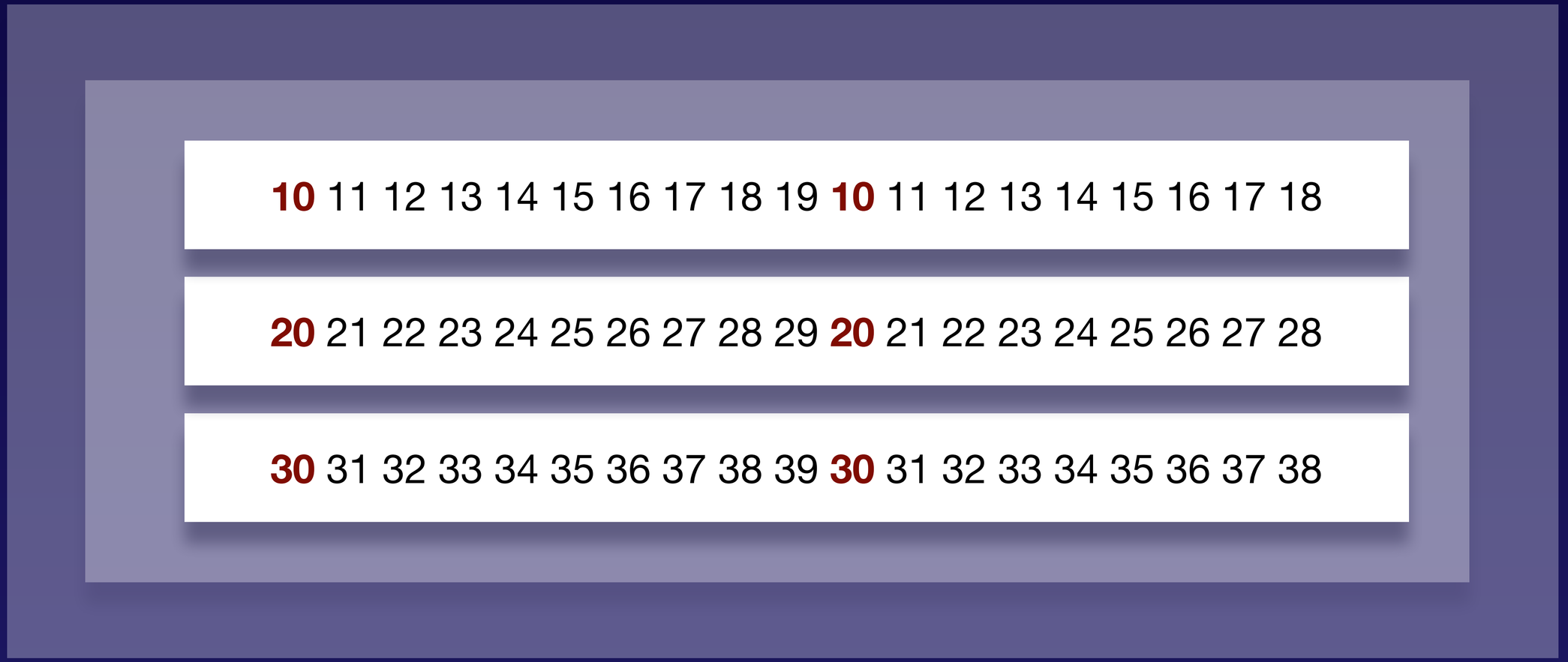
Bob	7	3
Frank	6	3

Fred	3	2
Dale	3	2
Sam	3	2

Paul	2	1
John	1	1
James	1	1


24 25 26 27 28 29 20 10 11 12 13 14 15 16 17 18 19 10 11 12 13
30 31 32 33 34 35 36 37 38 20 21 22 23 21 22 23 14 24 25 26 27
39 30 31 32 33 34 35 36 37 38 12 13 12 13 12 13 12 13 12 13 12

Record Batching?


10 11 12 13 14 15 16 17 18 19 10 11 12 13 14 15 16 17 18
20 21 22 23 24 25 26 27 28 29 20 21 22 23 24 25 26 27 28
30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38

SQS FIFO

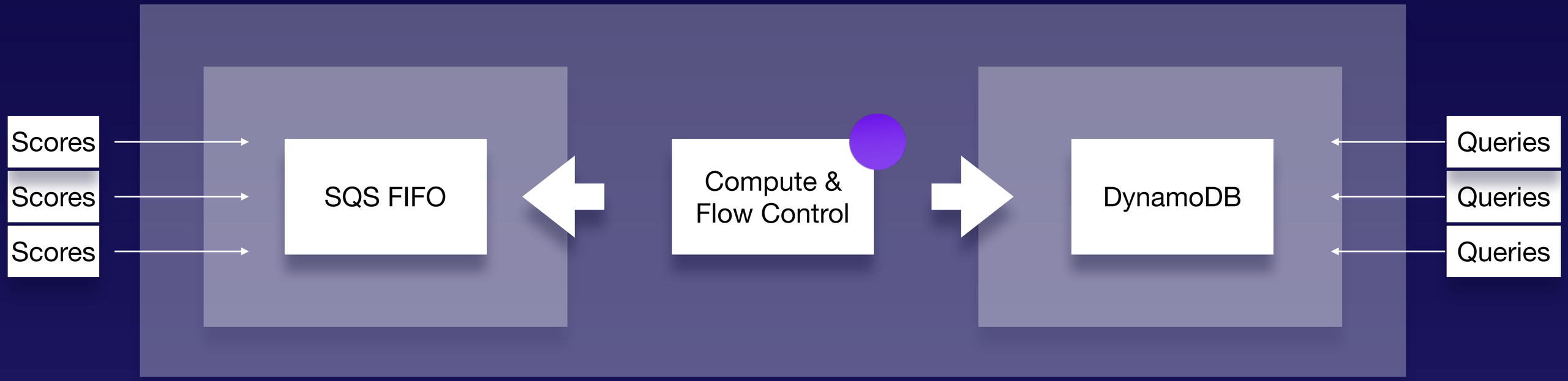
A CLOUD GURU

A CLOUD GURU

Compute & Flow Control

Architecture

A CLOUD GURU



Autoscaling Goals

A CLOUD GURU

1 **Maximise utilisation**

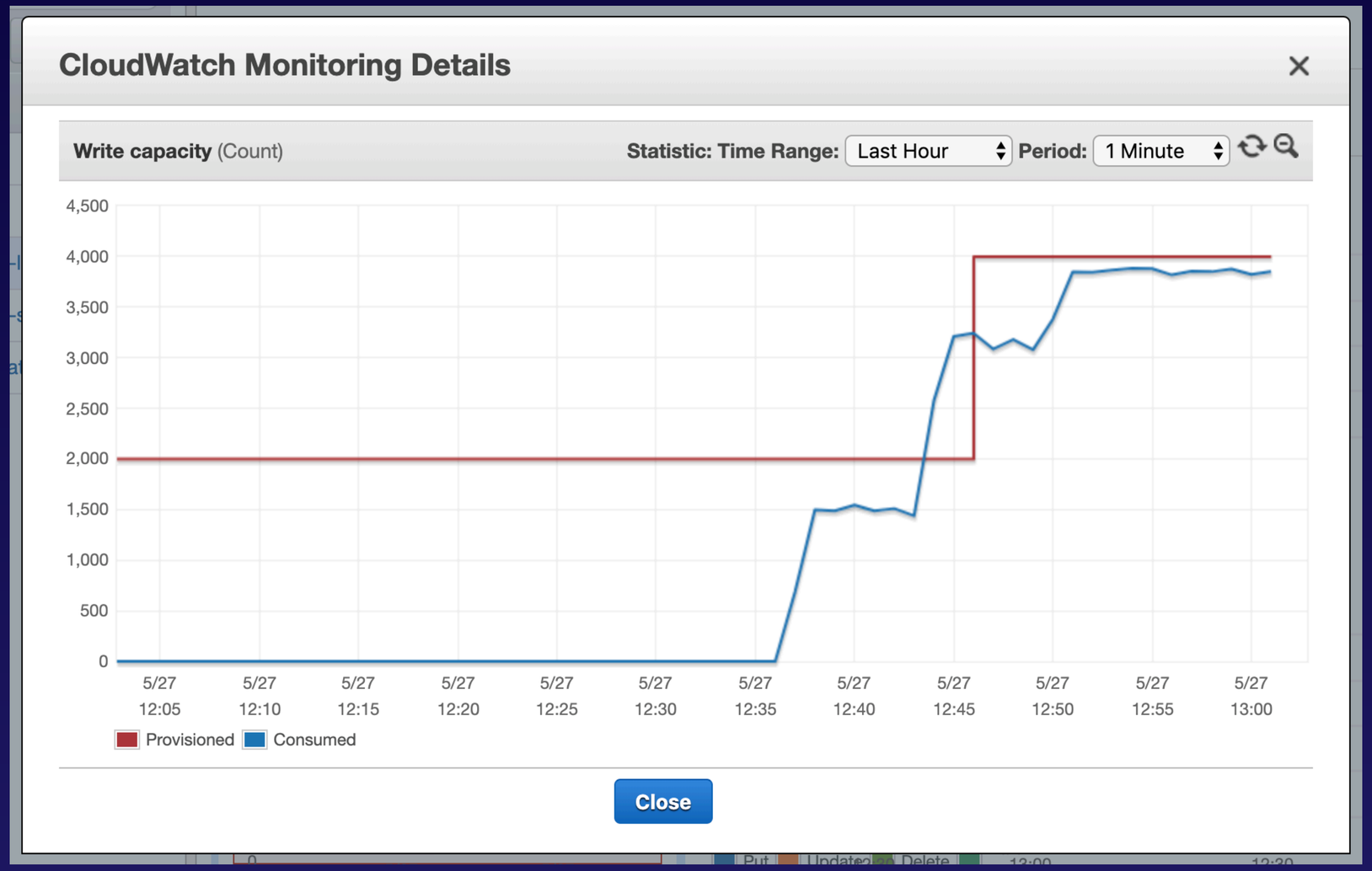
Correct amount of worker nodes

2 **No throughput errors**

Can happen based on **poor schema design**, **not spreading writes out evenly**, **too many requests p/s**

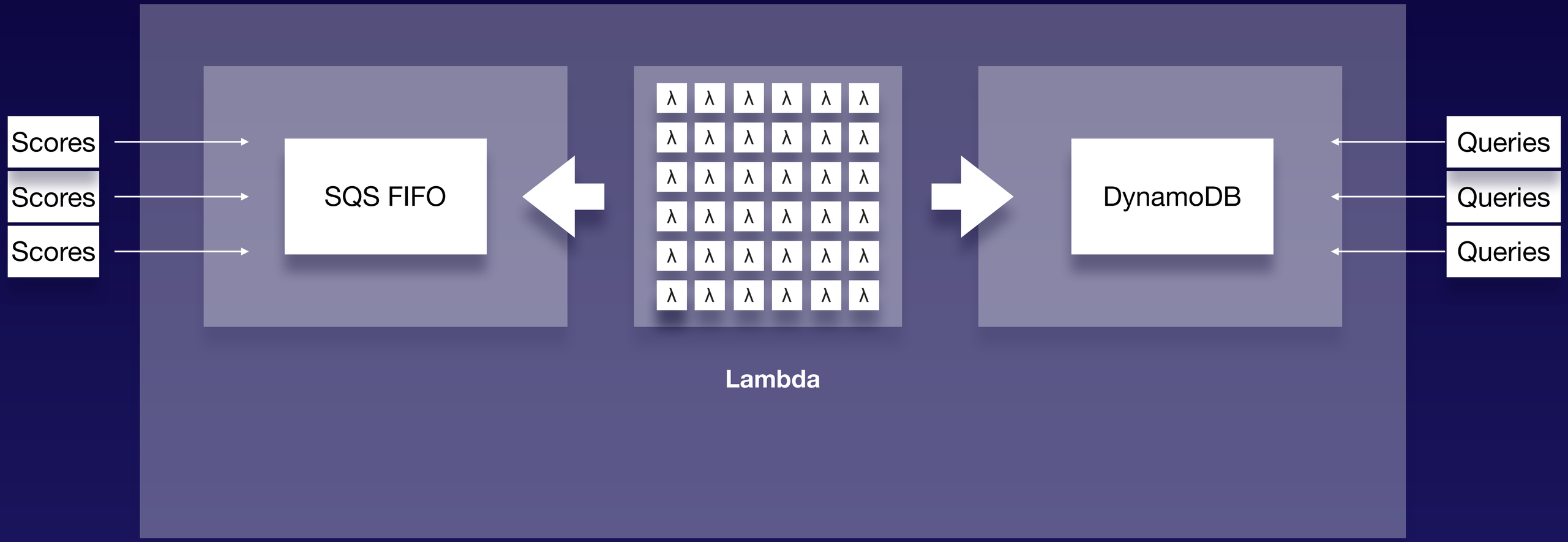
The Goal

A CLOUD GURU



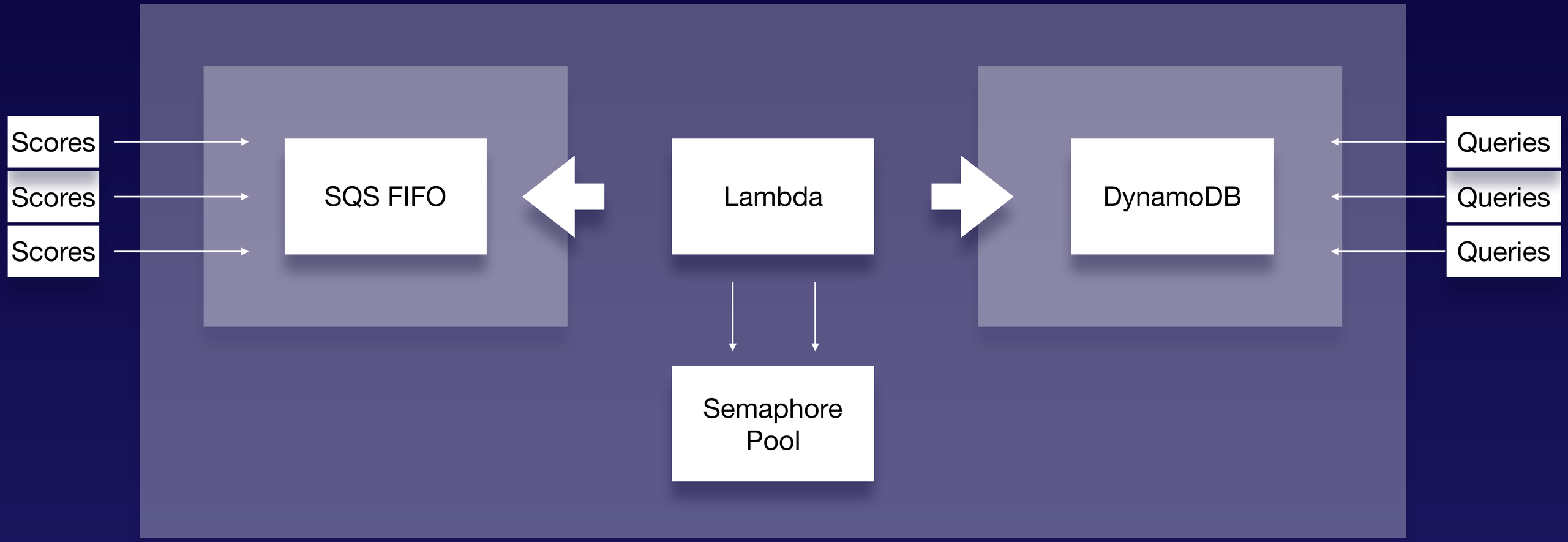
Worker Flow Control

A CLOUD GURU



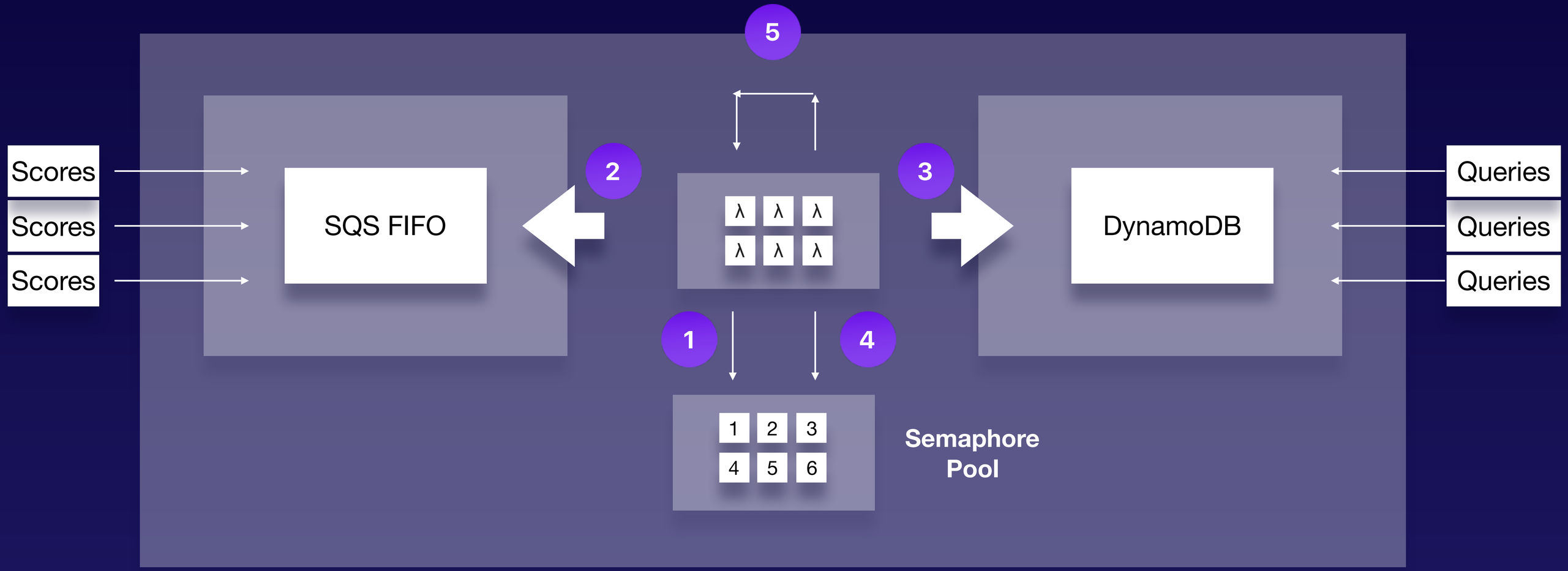
Worker Flow Control

A CLOUD GURU



Worker Flow Control

A CLOUD GURU



1 Try Acquire Semaphore

2 Get Score

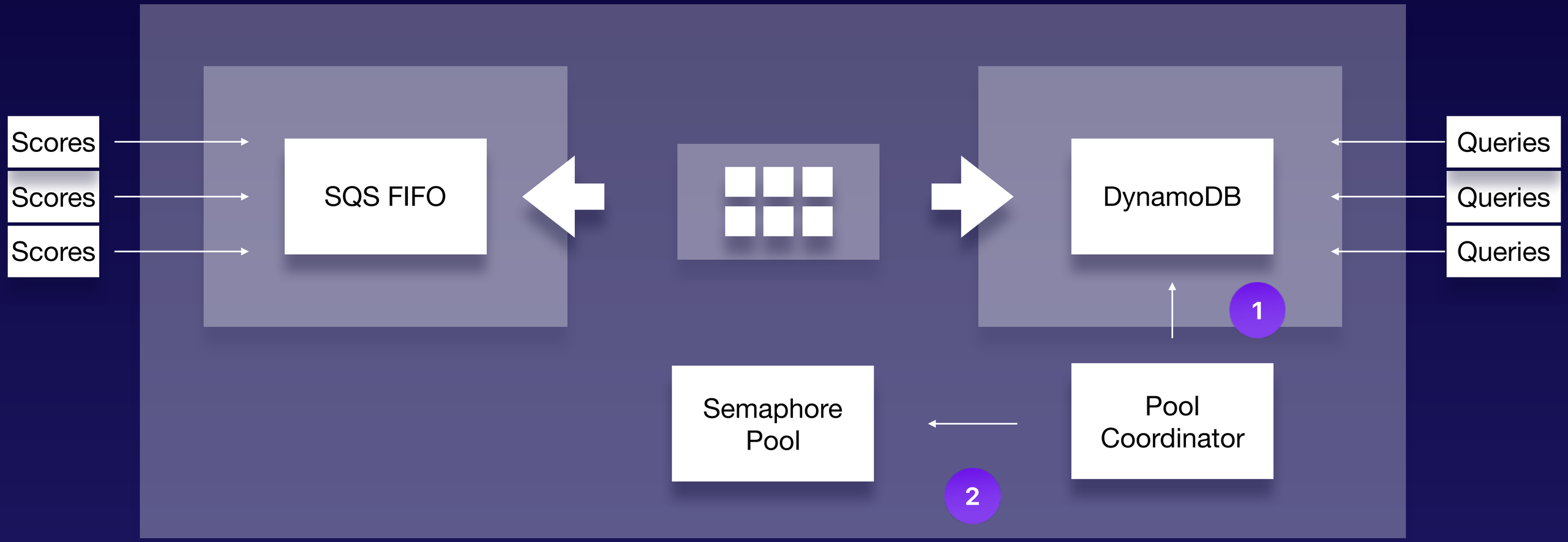
3 Write Score Updates

4 Return Semaphore

5 Trigger New Worker

Semaphore Management

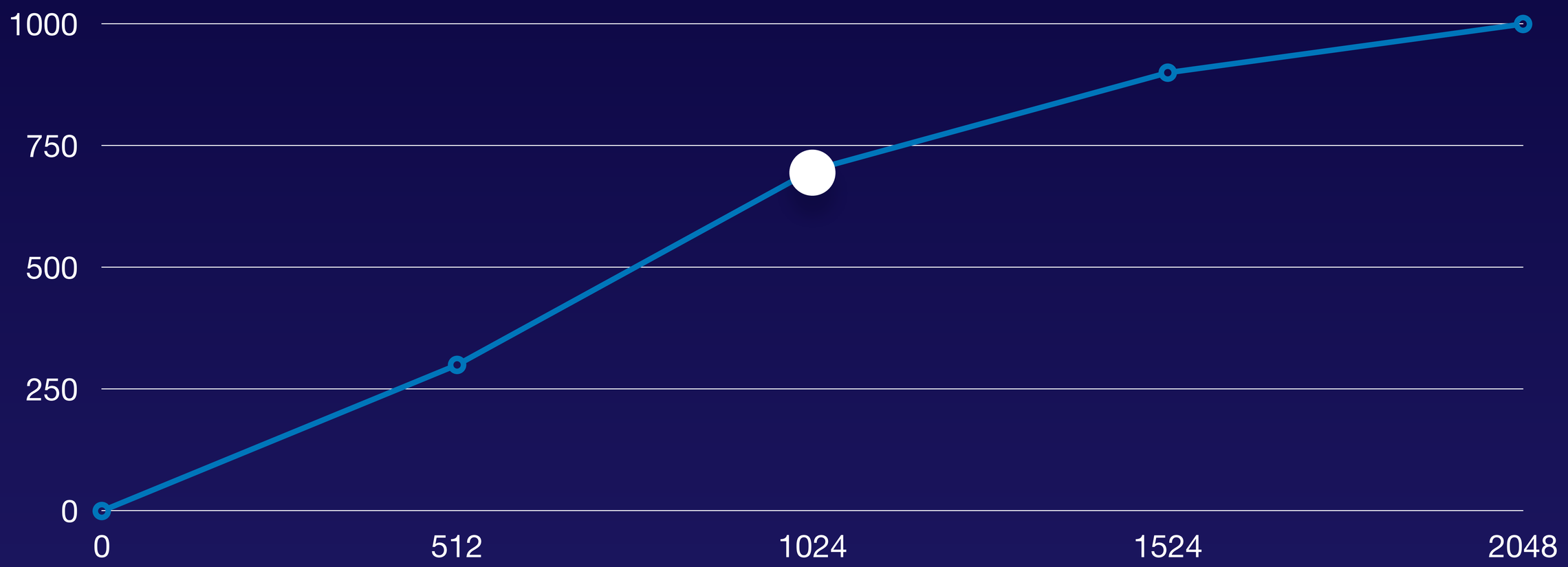
A CLOUD GURU



1. Look at table statistics
2. Update semaphore count

Semaphore Count

A CLOUD GURU



Semaphore Count

A CLOUD GURU

$$\text{Math.ceil}\left(\frac{\text{Table Capability}}{\text{Read / Write speed of worker}}\right) = \text{No. of Semaphores}$$

Semaphore Count

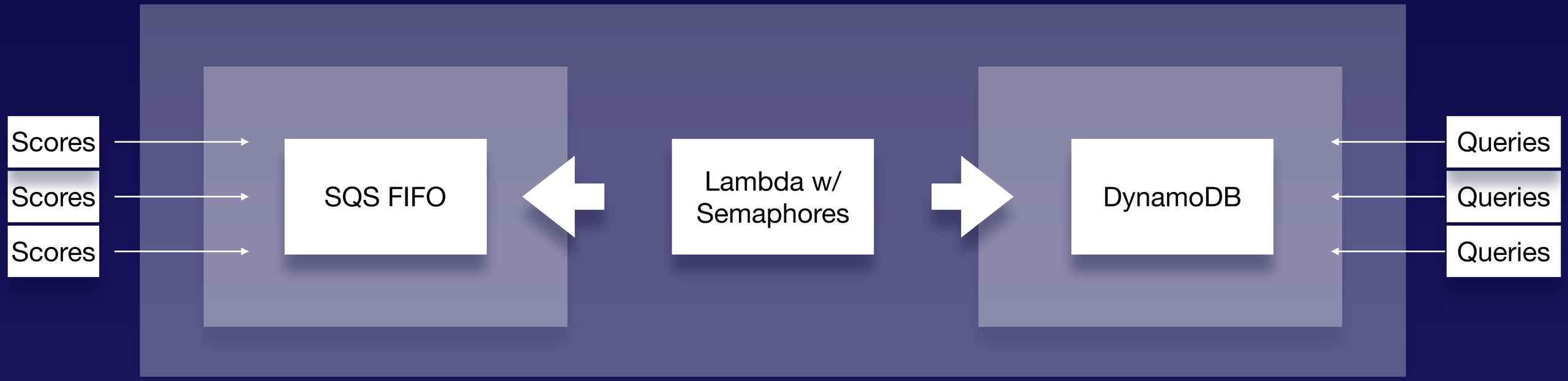
A CLOUD GURU

$$\text{Math.ceil}\left(\frac{2100\ \text{RCU / WCU}}{700\ \text{RW / s}}\right) = 3\ \text{Semaphores}$$

It’s complicated

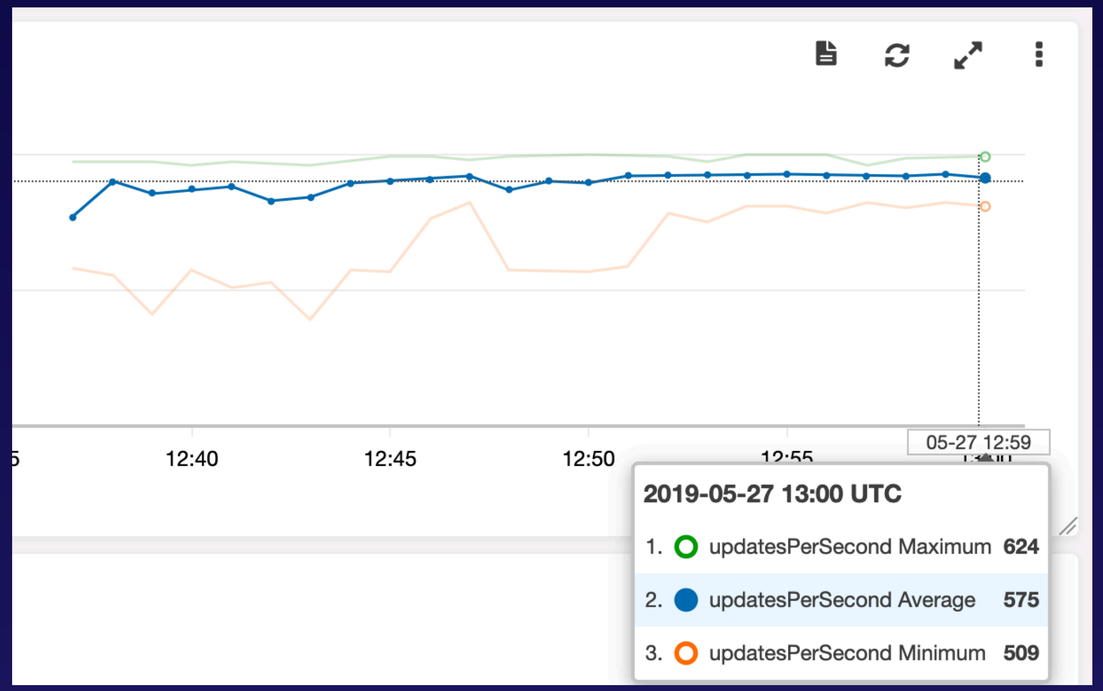
Architecture

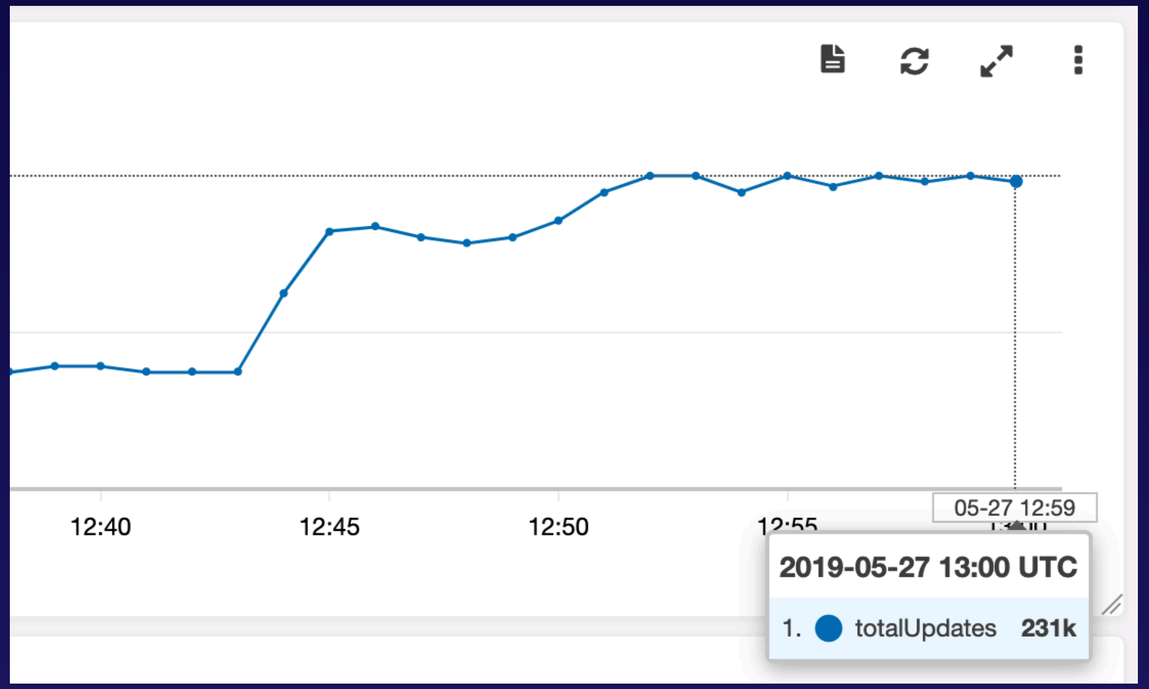
A CLOUD GURU



Result

A CLOUD GURU





3833 P/s

Questions?

Extras

TITLE TEXT

Materialised View Updates

A CLOUD GURU

+ 1

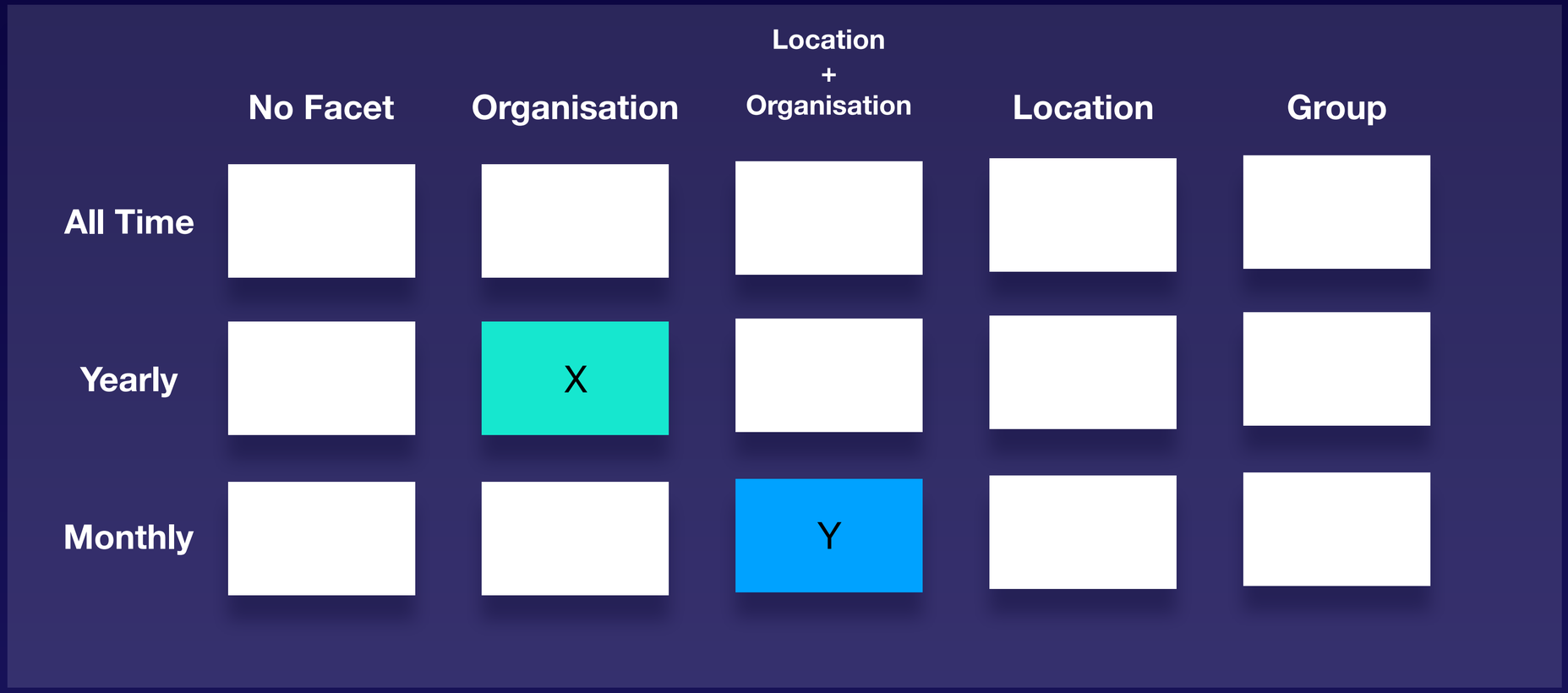
5 x 5

	No Facet	Organisation	Location + Organisation	Location	Group
All Time					
Yearly					
Monthly					
Weekly					
Daily					

TITLE TEXT

Index Updates

A CLOUD GURU



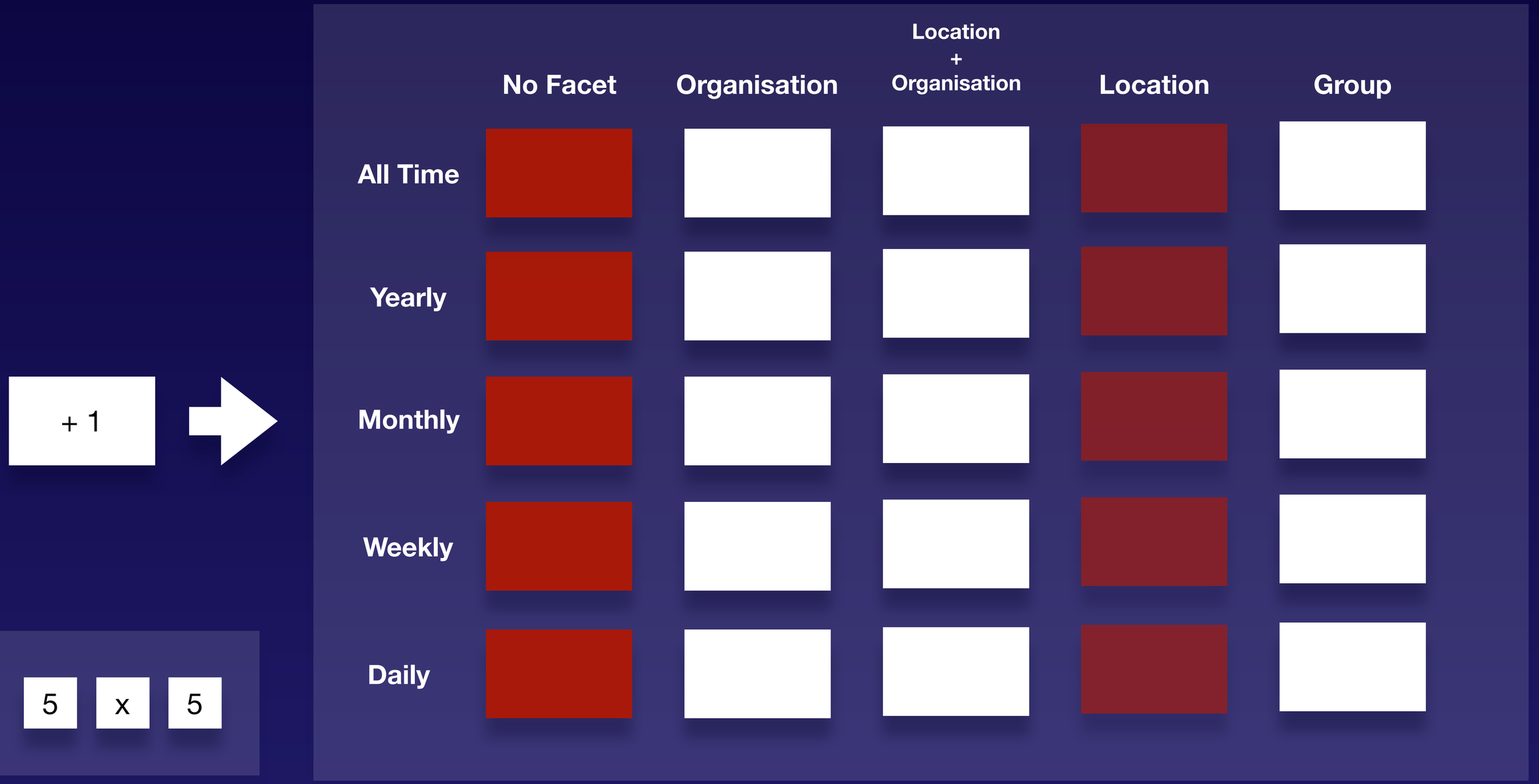
X = (Year=2018)-(Organisation=123)

Y = (Month=2018/12)-(Organisation=123)-(Location=Melb)

TITLE TEXT

Index Updates

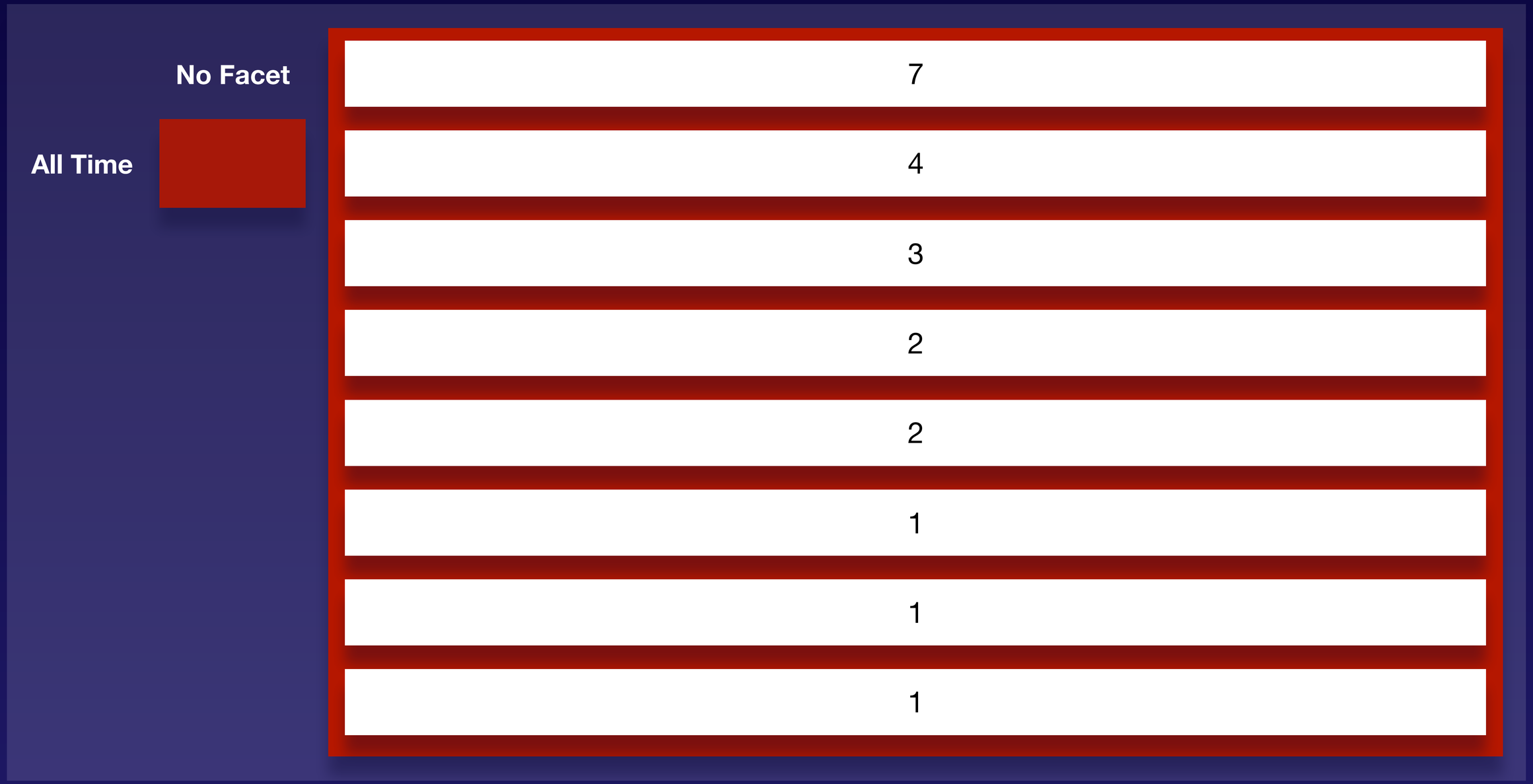
A CLOUD GURU



TITLE TEXT

Index Updates

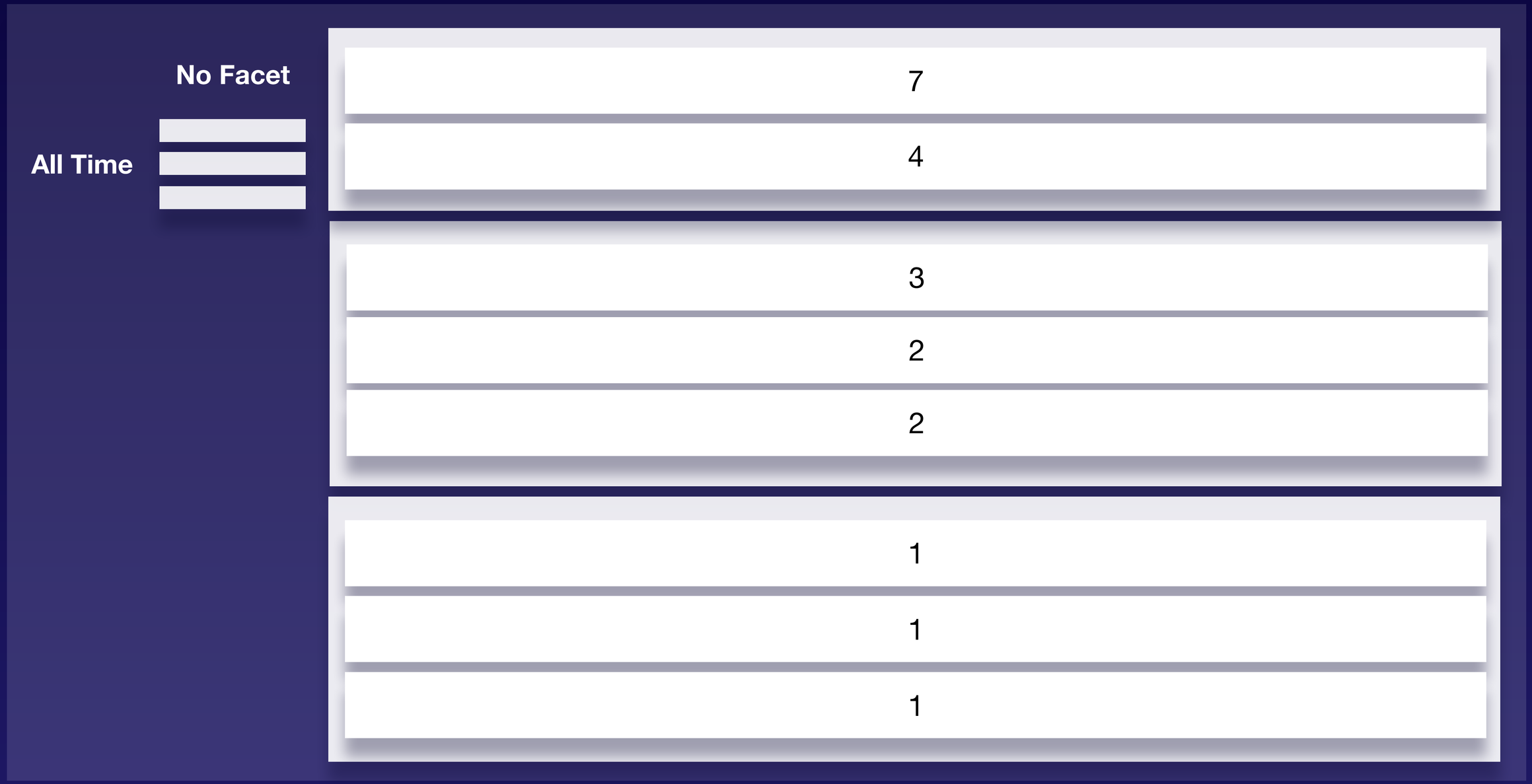
A CLOUD GURU



TITLE TEXT

Index Updates

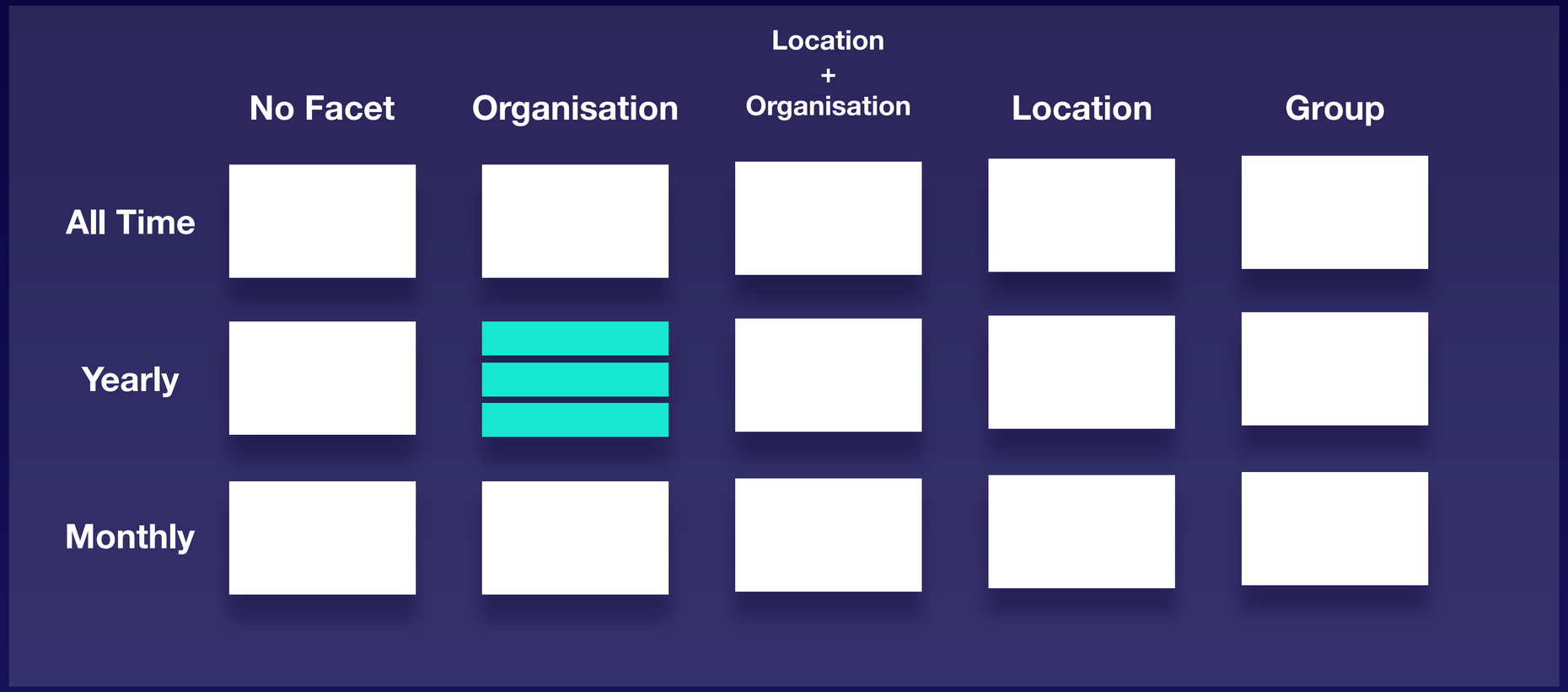
A CLOUD GURU



TITLE TEXT

Index Updates

A CLOUD GURU



X = (Year=2018)-(Organisation=123)_3

X = (Year=2018)-(Organisation=123)_2