AWS re:Invent

DAT304

Scale fearlessly with Amazon DynamoDB adaptive capacity

Kai Zhao

Principal Product Manager

Amazon Web Services

AWS re:Invent

aws

Agenda

Become an Amazon DynamoDB scaling expert

New tools and features

Prepare for unpredictable traffic

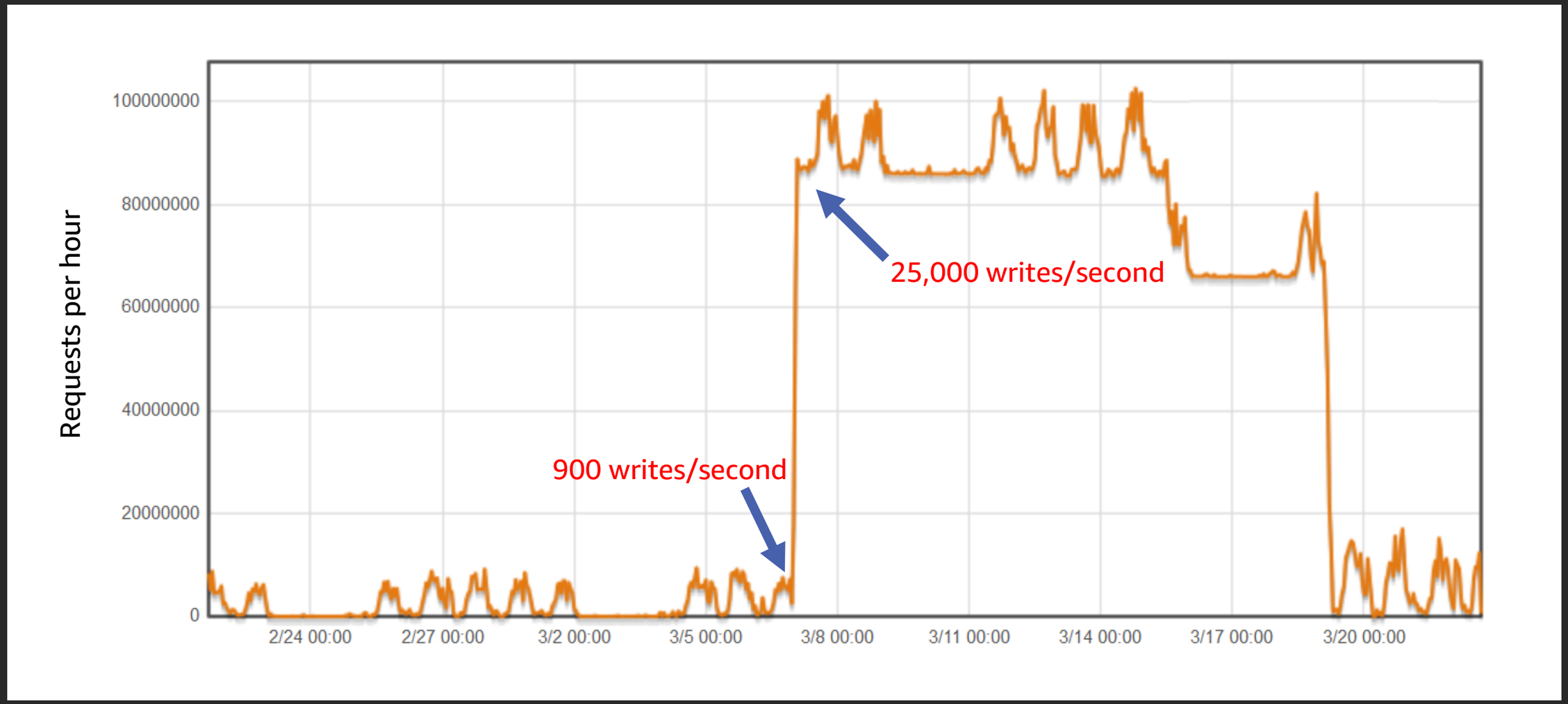
Save money

Common scaling scenarios

Common scaling scenarios

Common scaling scenarios

25x increase



Key takeaways

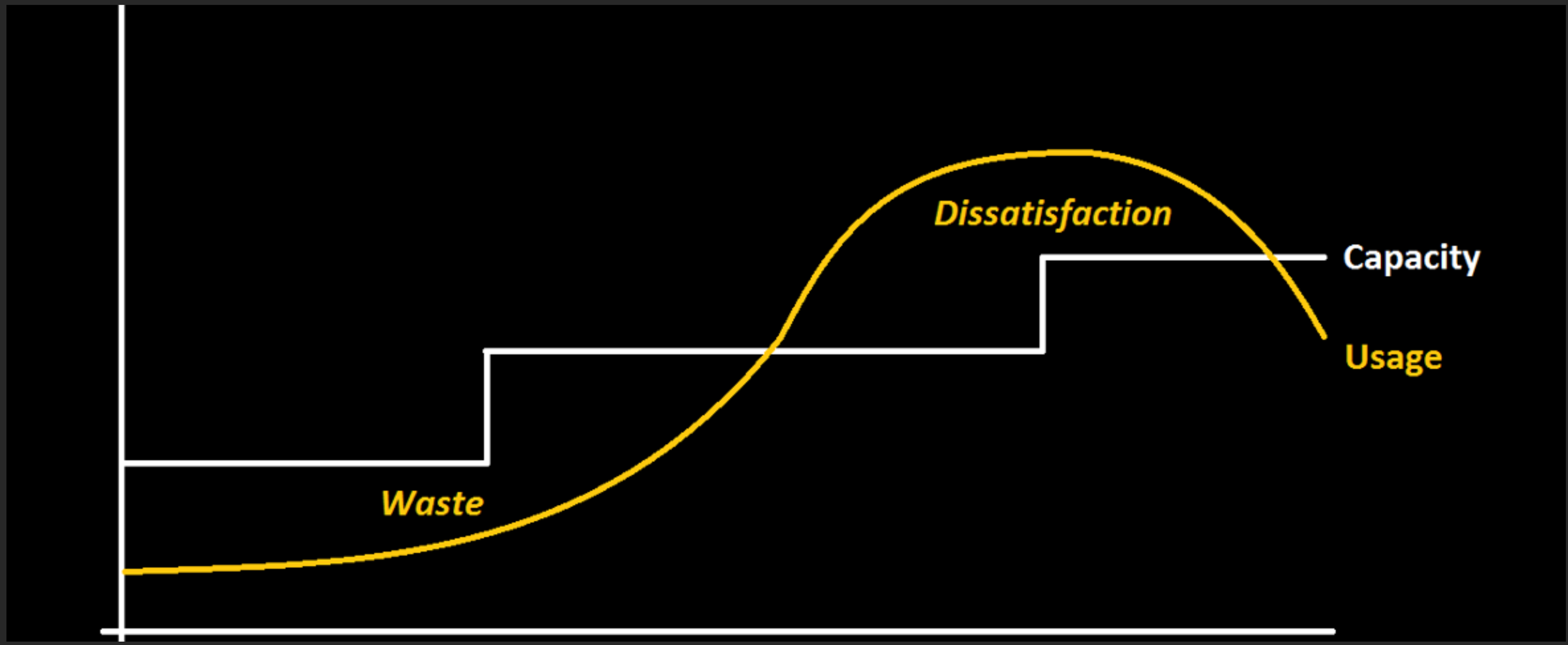
DynamoDB accommodates your workload, not vice versa

Partitions don’t matter; individual keys do

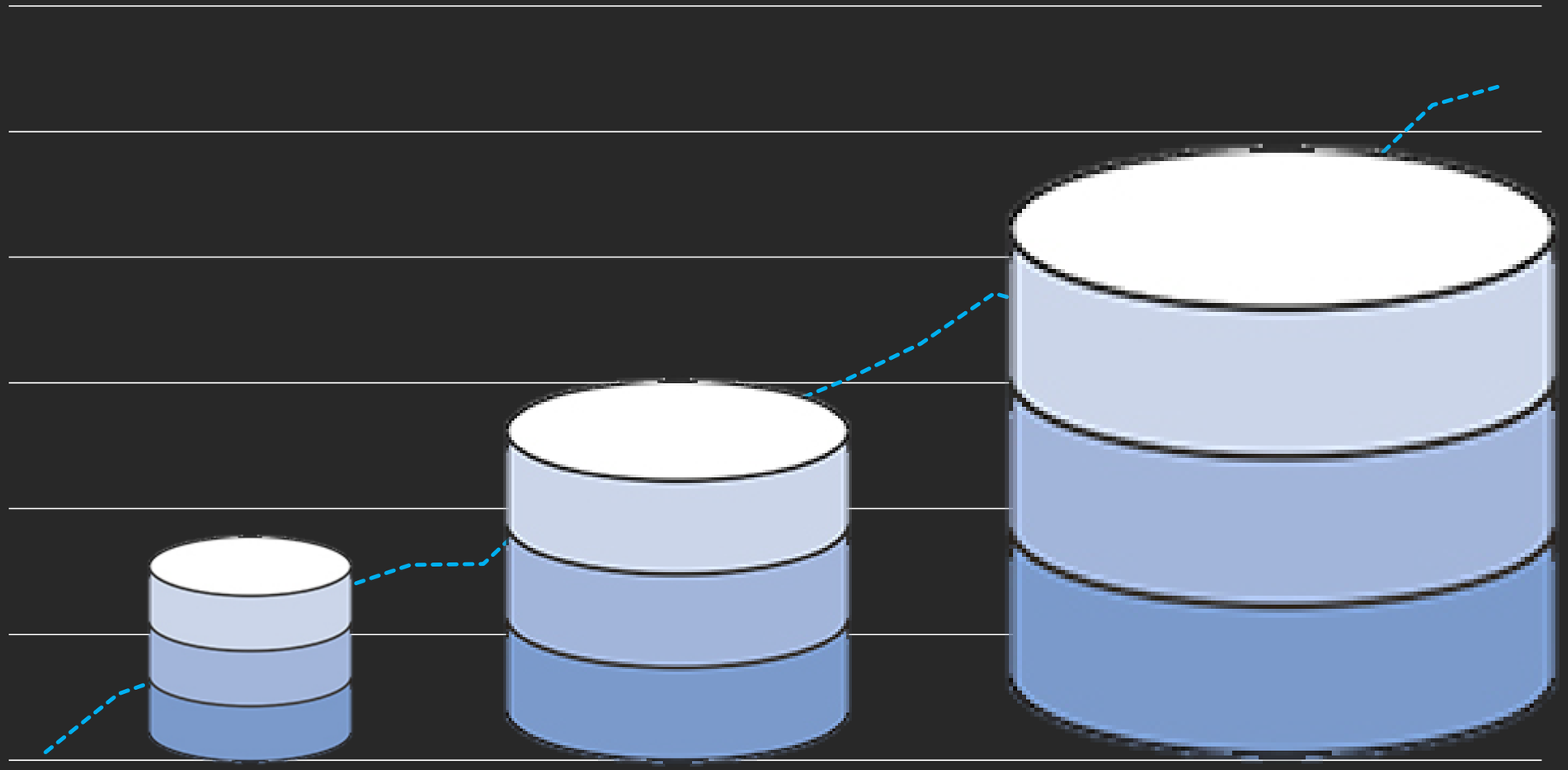
Switch capacity modes in order to optimize cost and performance

DynamoDB's approach to scaling

Database capacity planning



Traditional (vertical) scaling



Dynamo: Amazon's Highly Available Key-value Store

Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels

Amazon.com

ABSTRACT

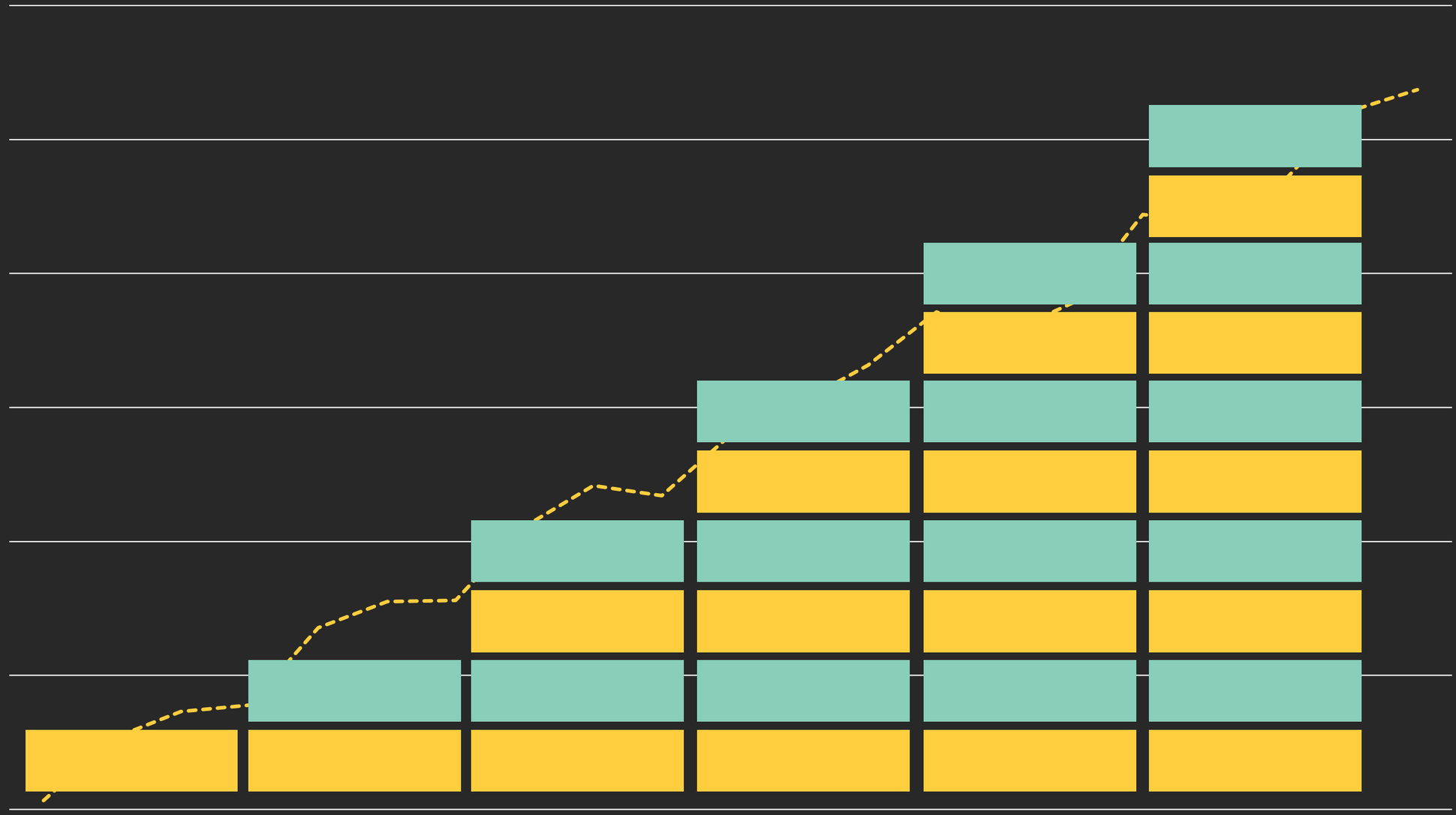
Reliability at massive scale is one of the biggest challenges we face at Amazon.com, one of the largest e-commerce operations in the world; even the slightest outage has significant financial consequences and impacts customer trust. The Amazon.com platform, which provides services for many web sites worldwide, is implemented on top of an infrastructure of tens of thousands of servers and network components located in many datacenters around the world. At this scale, small and large components fail continuously and the way persistent state is managed in the face of these failures drives the reliability and scalability of the software systems.

This paper presents the design and implementation of Dynamo, a highly available key-value storage system that some of Amazon's core services use to provide an "always-on" experience. To achieve this level of availability, Dynamo sacrifices consistency under certain failure scenarios. It makes extensive use of object versioning and application-assisted conflict resolution in a manner that provides a novel interface for developers to use.

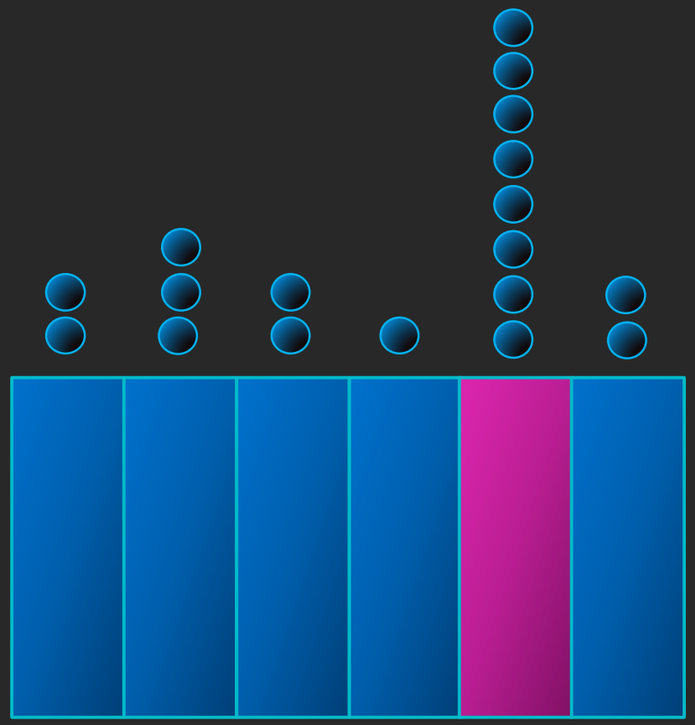
One of the lessons our organization has learned from operating Amazon's platform is that the reliability and scalability of a system is dependent on how its application state is managed. Amazon uses a highly decentralized, loosely coupled, service oriented architecture consisting of hundreds of services. In this environment there is a particular need for storage technologies that are always available. For example, customers should be able to view and add items to their shopping cart even if disks are failing, network routes are flapping, or data centers are being destroyed by tornados. Therefore, the service responsible for managing shopping carts requires that it can always write to and read from its data store, and that its data needs to be available across multiple data centers.

Dealing with failures in an infrastructure comprised of millions of components is our standard mode of operation; there are always a small but significant number of server and network components that are failing at any given time. As such Amazon's software systems need to be constructed in a manner that treats failure handling as the normal case without impacting availability or performance.

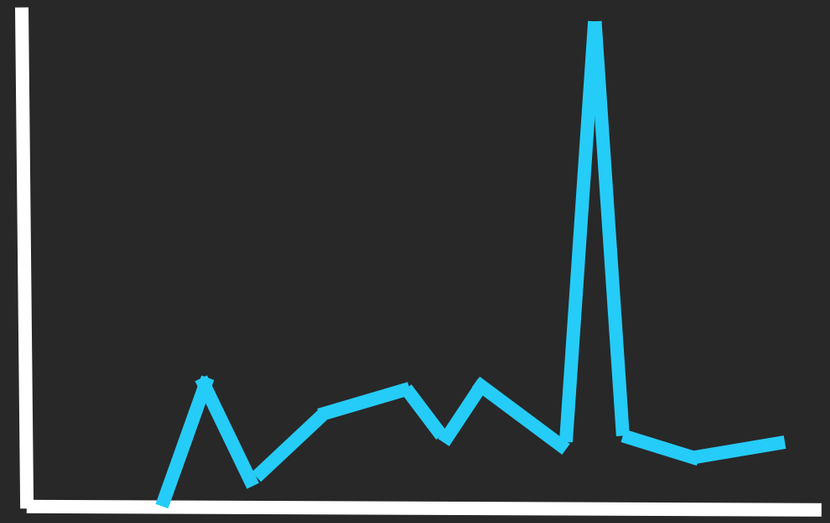
Horizontal scaling



The uneven access problem common to NoSQL



Across data



and time

Just design for uniform access to your data, right?

Can be quite difficult in practice

Most NoSQL databases aren’t very forgiving

If your database isn’t flexible, you must handle more in your application

How does DynamoDB solve this for customers?

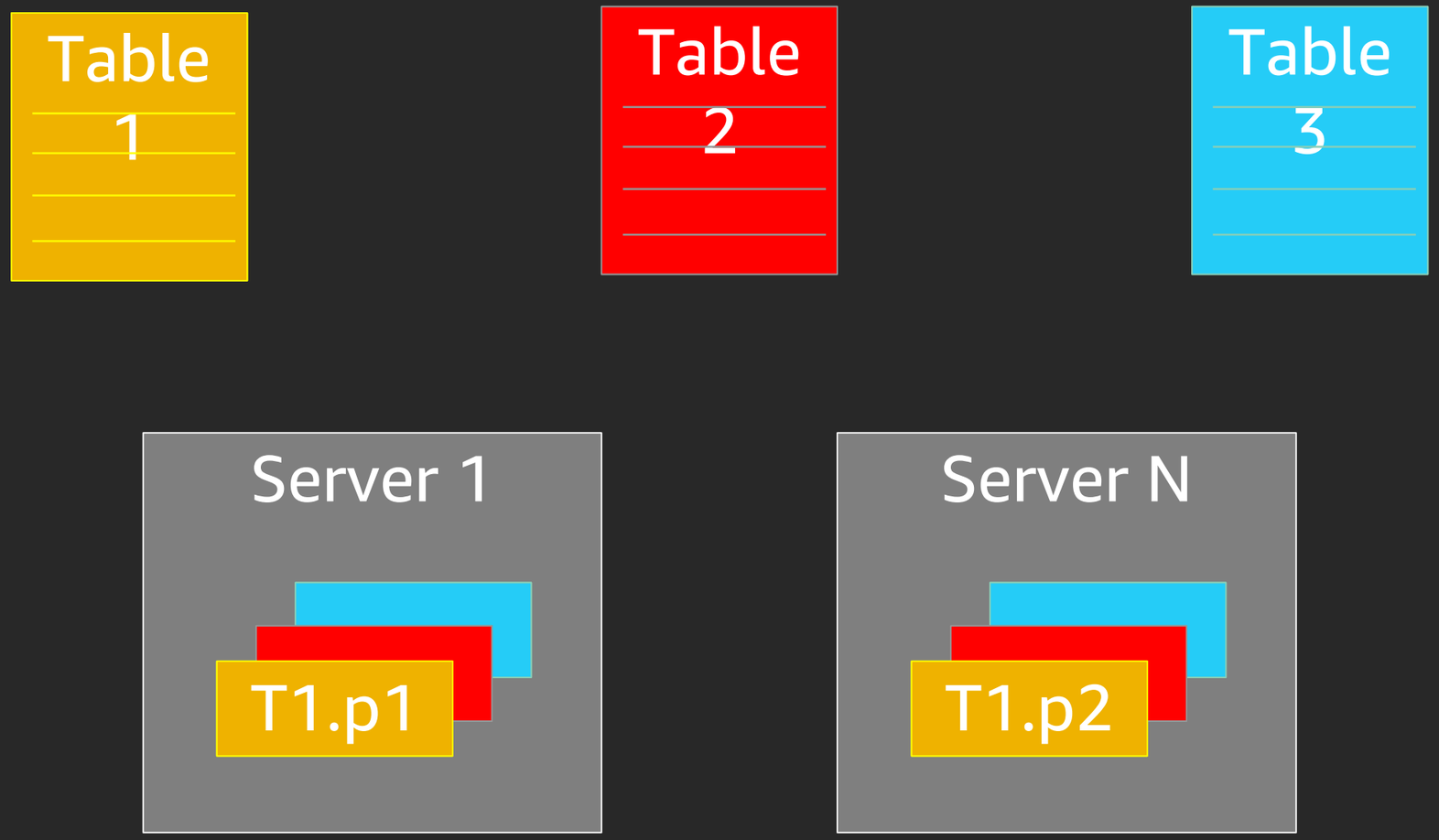
Adaptive capacity

Core functions

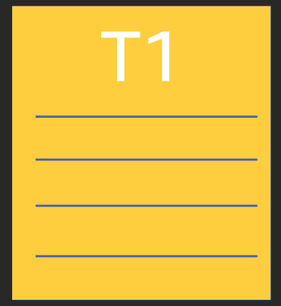
- Dynamic partitioning
- High-traffic item isolation
- Throughput boosting

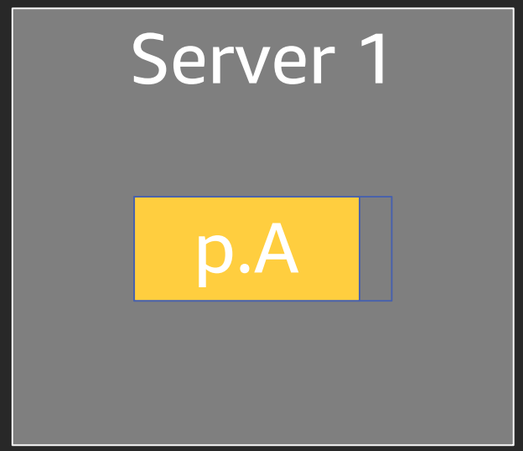
Dynamic partitioning

Behind the scenes

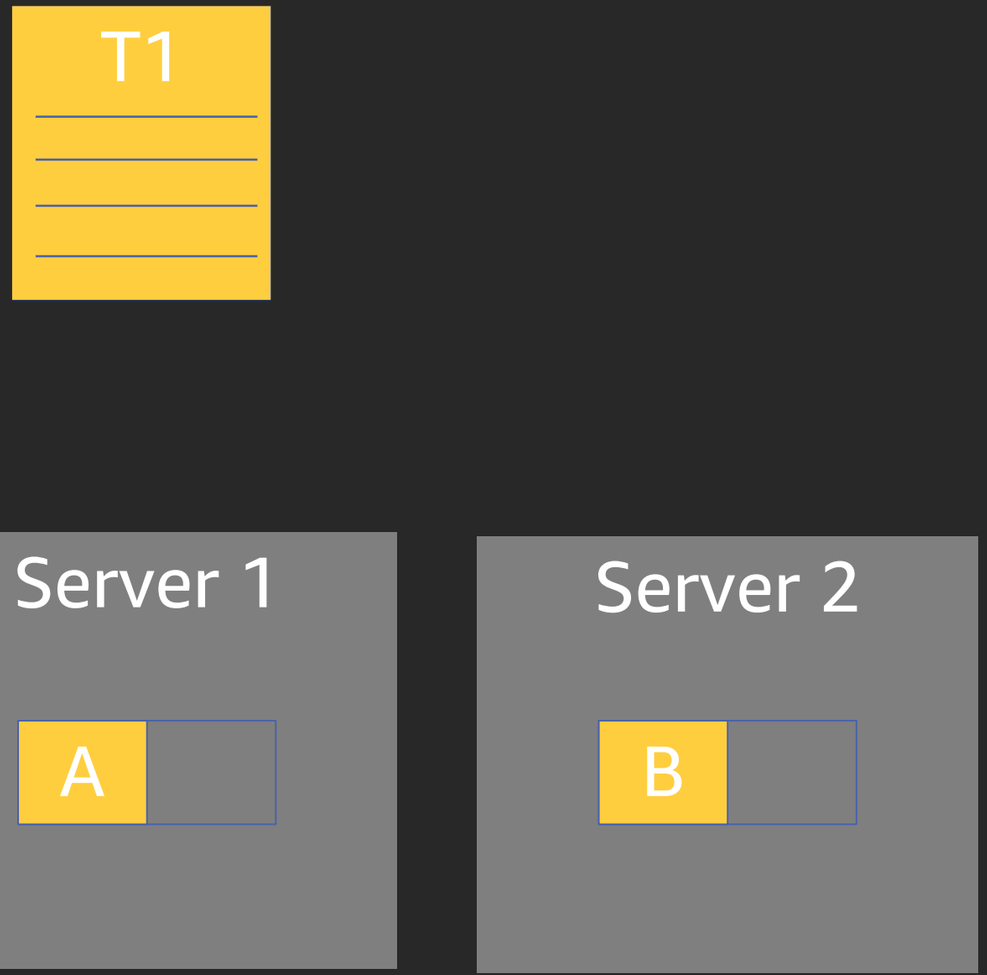


Storage scaling

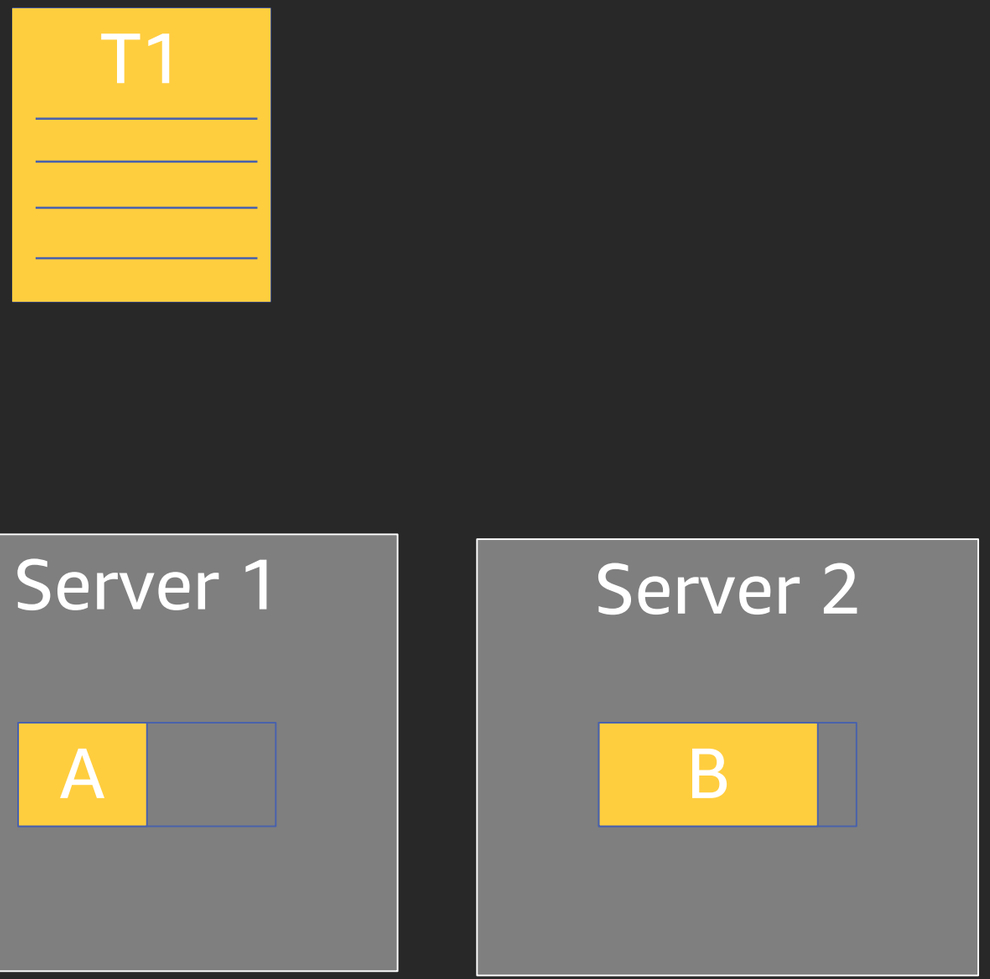




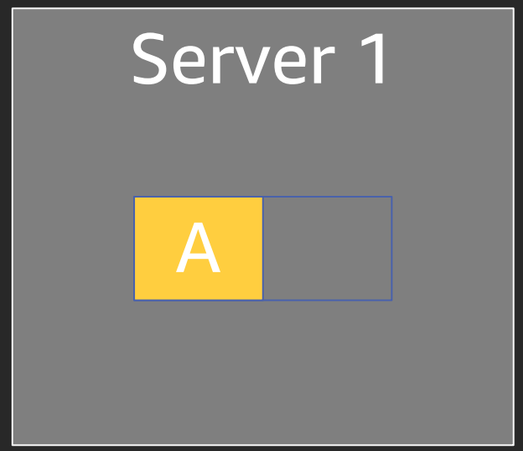
Storage scaling

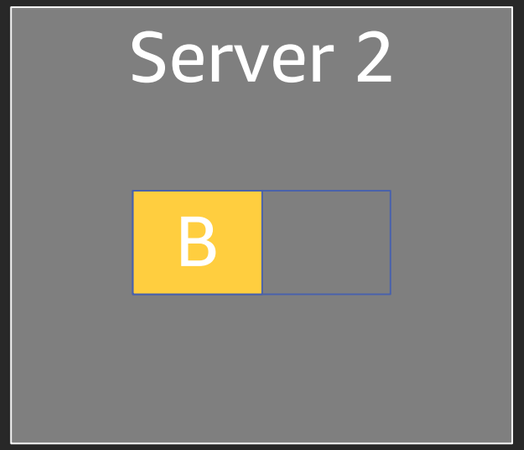


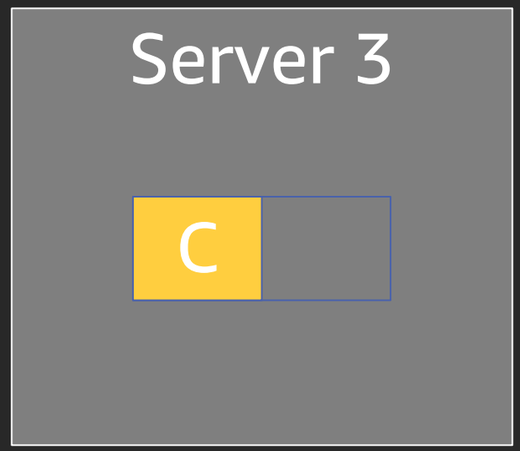
Storage scaling



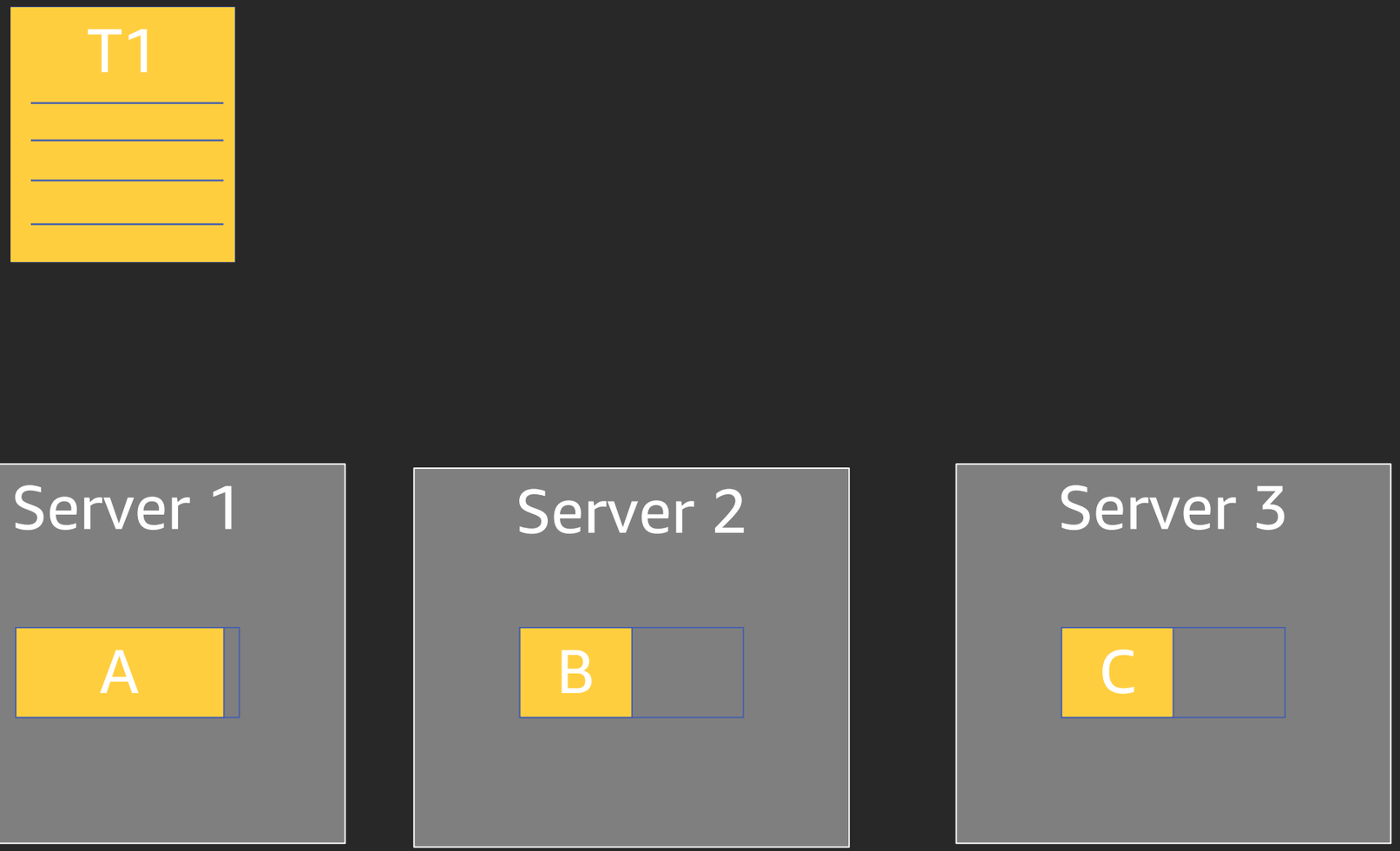
Storage scaling



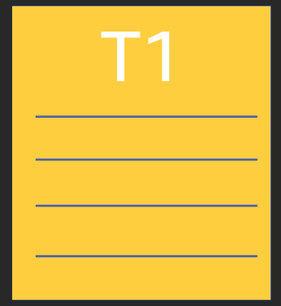


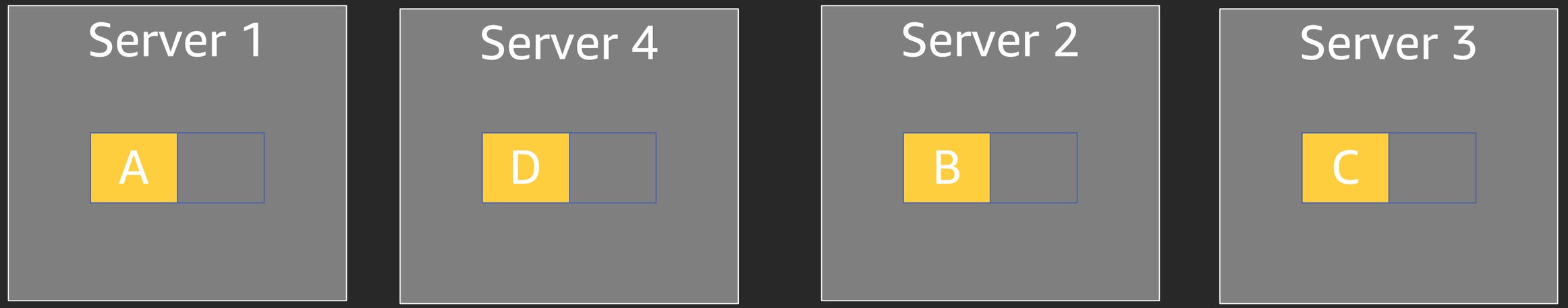


Storage scaling



Storage scaling





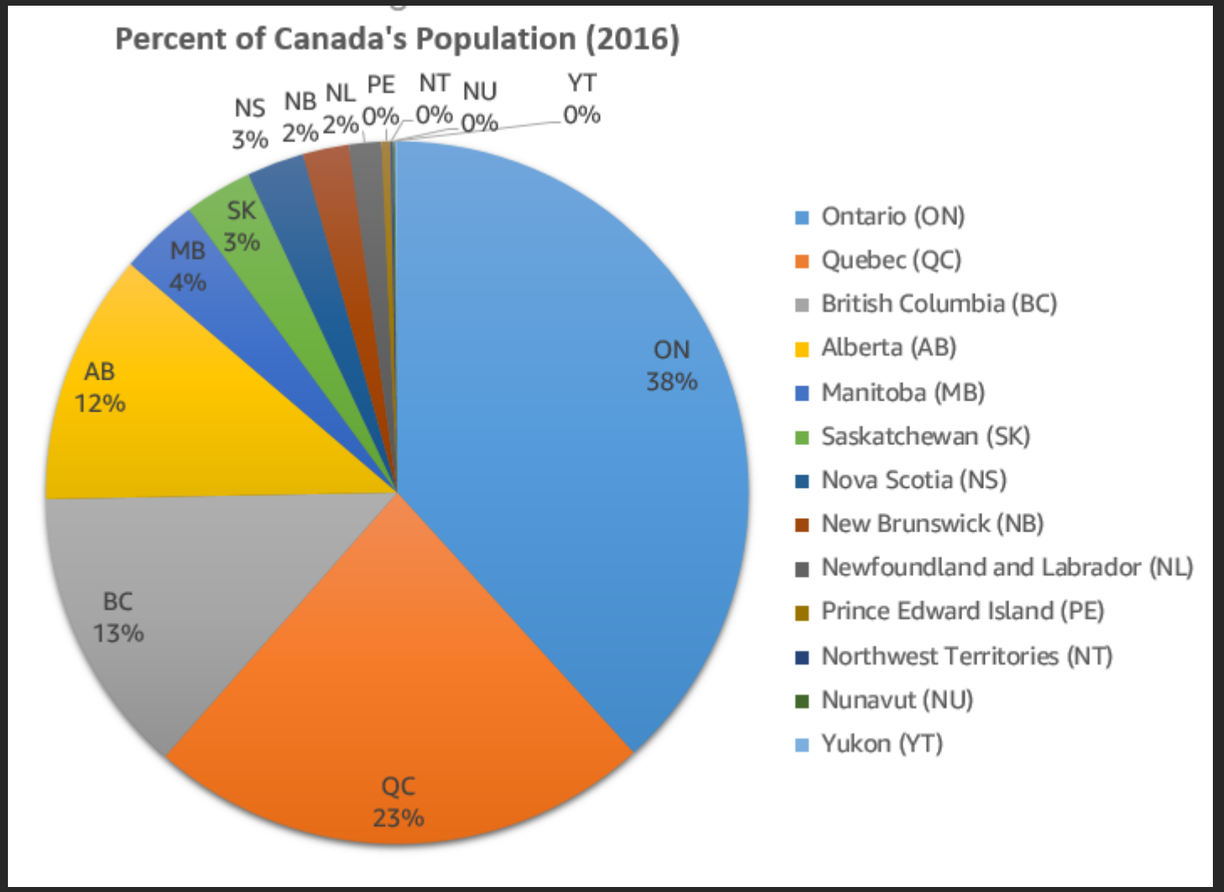
Scenario: Census application

Statistics Canada (Canada's national statistical agency) hires you to build an online census application

You choose DynamoDB with the following key schema:

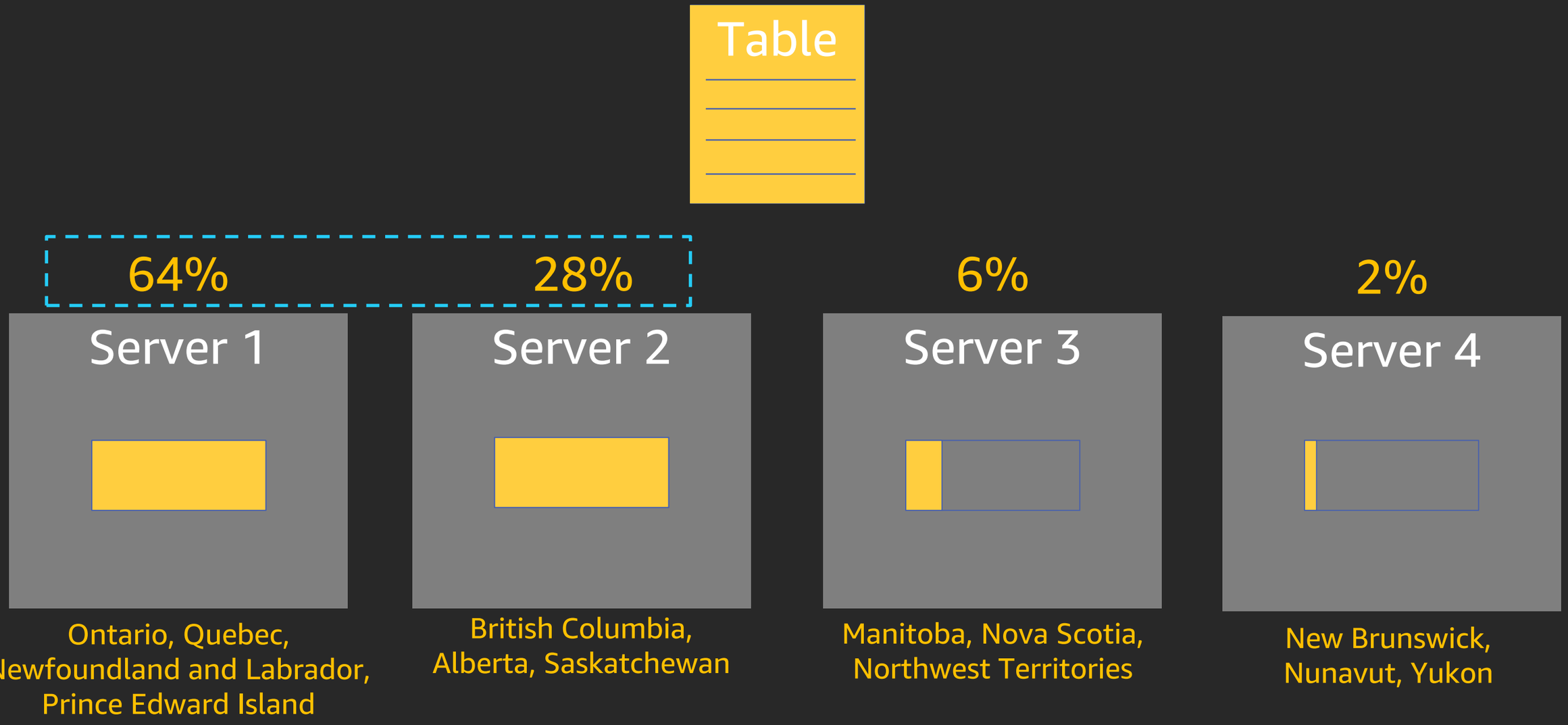
- Partition key: **province**
- Sort key: **id**

What you didn’t realize

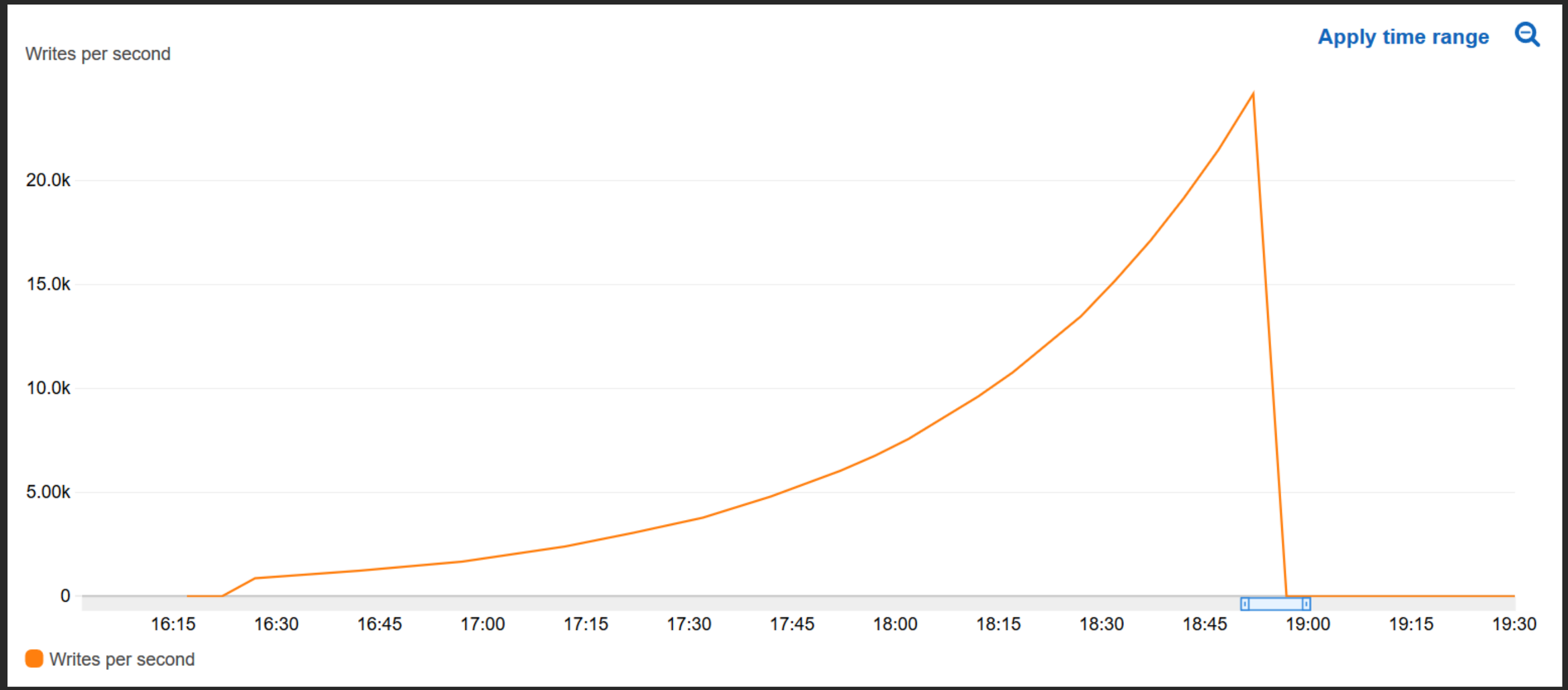


Demo

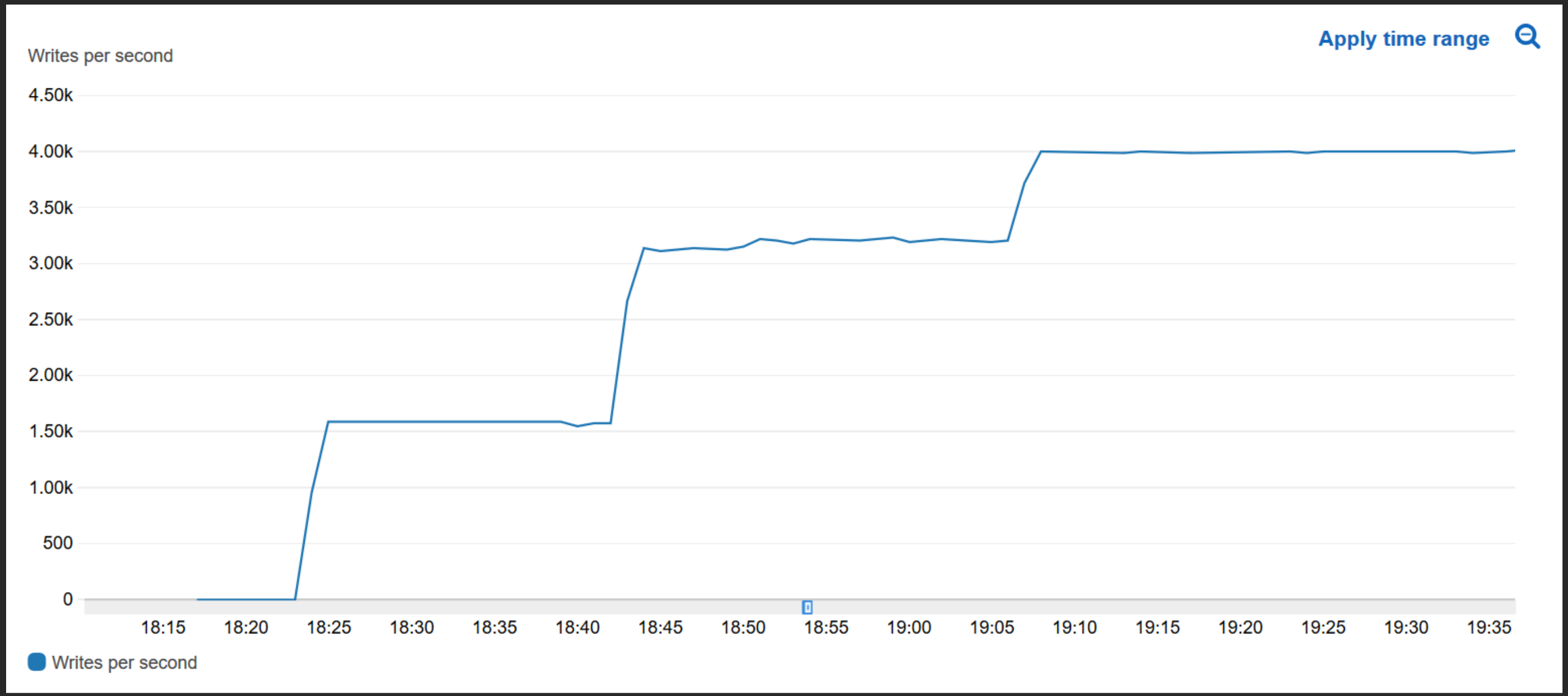
Census traffic



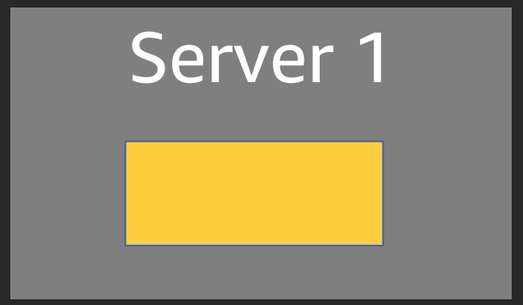
Dynamic partitioning



Dynamic partitioning

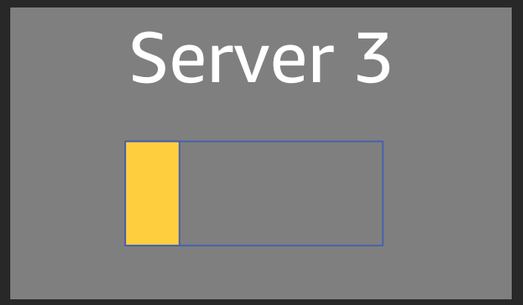


Dynamic partitioning: Behind the scenes

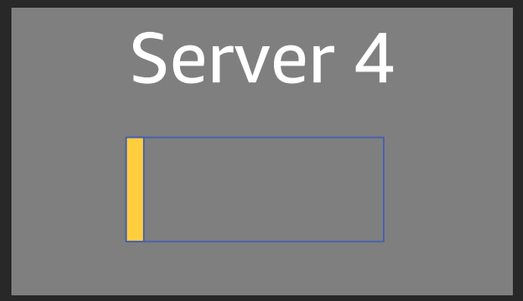


Ontario, Quebec,
Newfoundland and Labrador,
Prince Edward Island

British Columbia,
Alberta, Saskatchewan



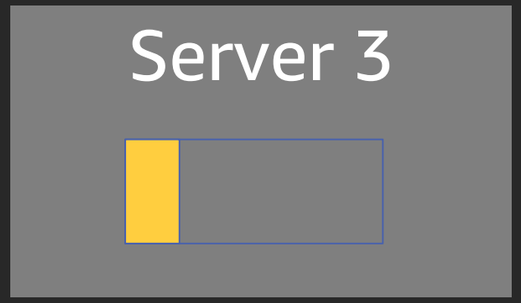
Manitoba, Nova Scotia,
Northwest Territories



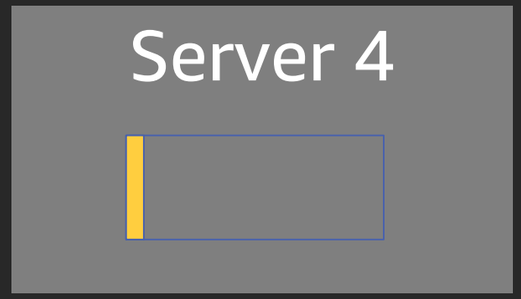
New Brunswick,
Nunavut, Yukon

Dynamic partitioning: Behind the scenes

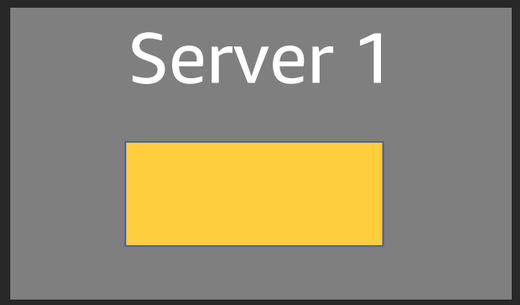
British Columbia, Alberta, Saskatchewan



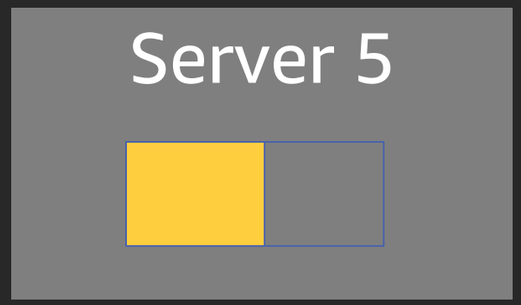
Manitoba, Nova Scotia, Northwest Territories



New Brunswick, Nunavut, Yukon



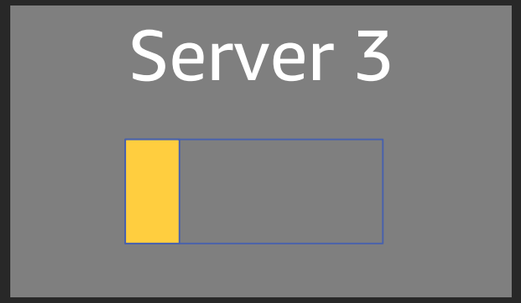
Ontario



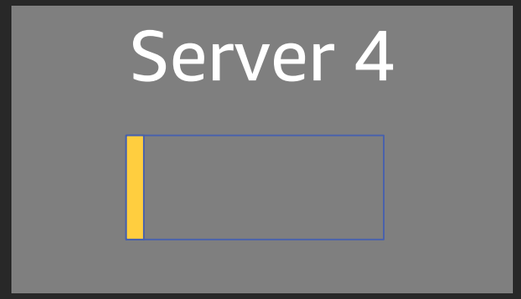
Quebec, Newfoundland and Labrador, Prince Edward Island

Dynamic partitioning: Behind the scenes

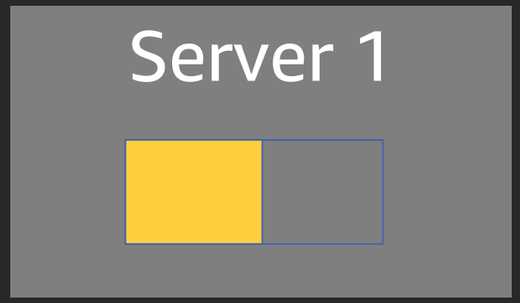
British Columbia, Alberta, Saskatchewan



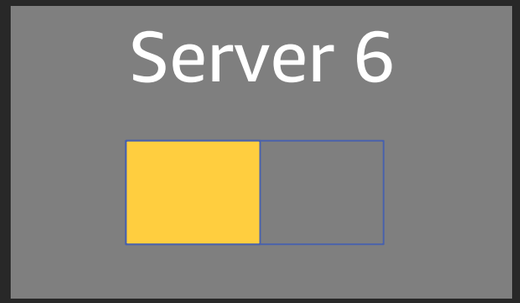
Manitoba, Nova Scotia, Northwest Territories



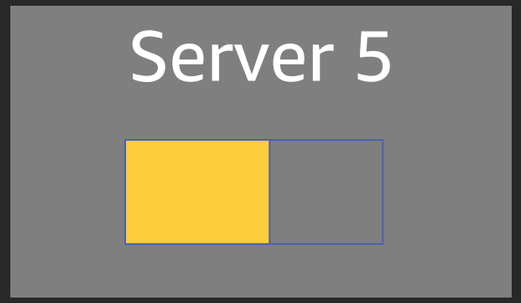
New Brunswick, Nunavut, Yukon



Ontario.A

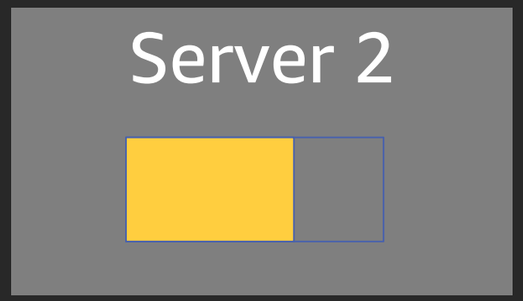


Ontario.B

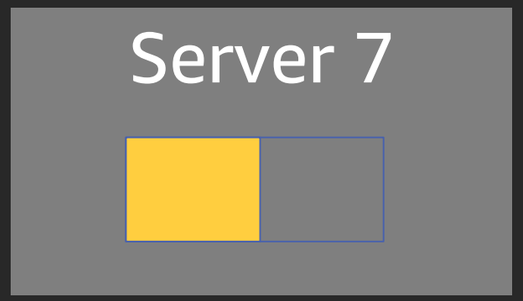


Quebec, Newfoundland and Labrador, Prince Edward Island

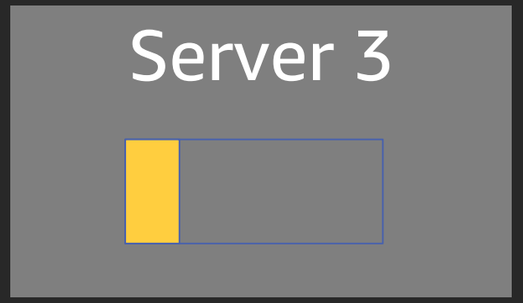
Dynamic partitioning: Behind the scenes



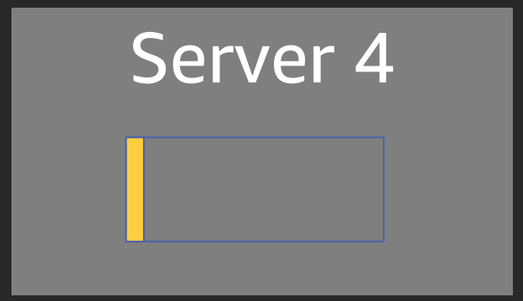
British Columbia



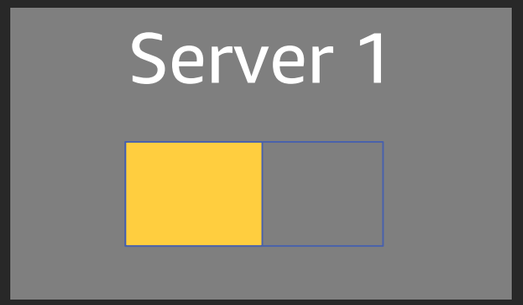
Alberta, Saskatchewan



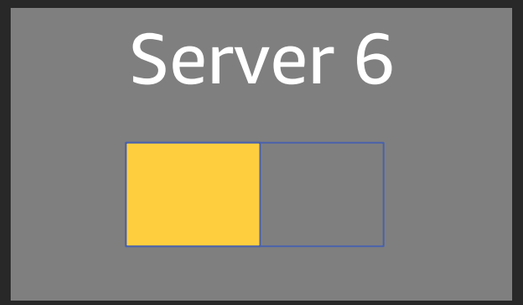
Manitoba, Nova Scotia, Northwest Territories



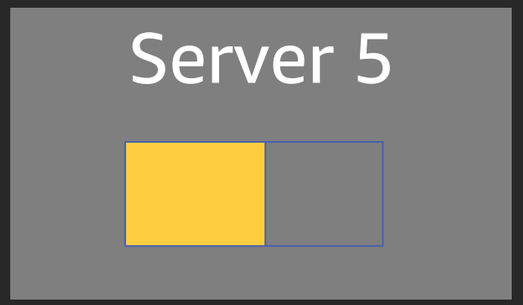
New Brunswick, Nunavut, Yukon



Ontario.A



Ontario.B



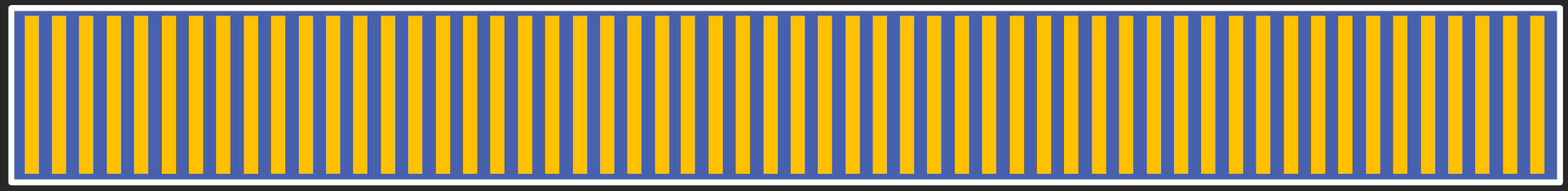
Quebec, Newfoundland and Labrador, Prince Edward Island

Key takeaway 1

DynamoDB adapts to your workload, not vice versa

High-traffic item isolation

High-traffic item isolation (new 2019)

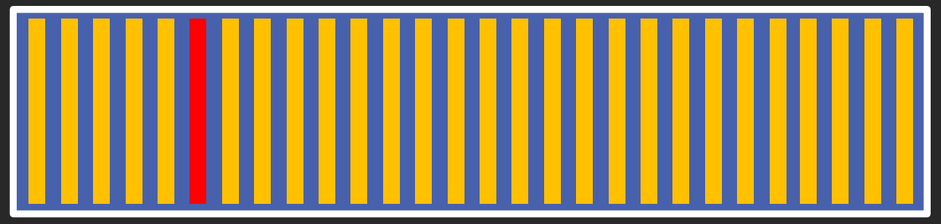


Partition A

High-traffic item isolation (new 2019)

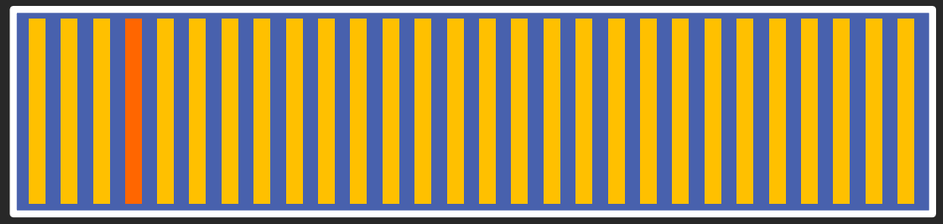
High-traffic item isolation (new 2019)

Item “foo”



Partition A

Item “bar”

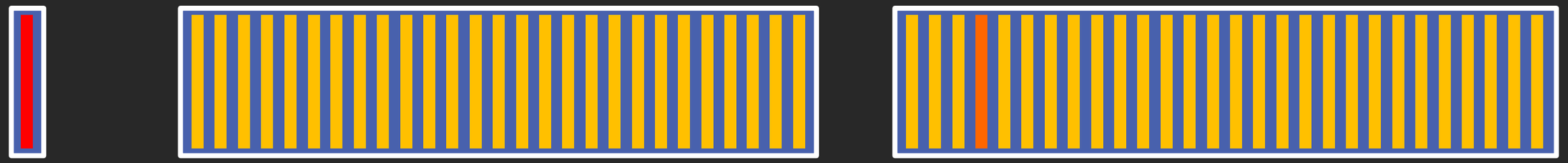


Partition B

High-traffic item isolation (new 2019)

Item “foo”

Item “bar”



Partition C

Partition A

Partition B

Key takeaway 2

Partitions don’t matter, individual items do

Items (keys)

Common questions

“How do I see traffic at a per-item level?”

Amazon CloudWatch contributor insights for DynamoDB

New!

Features

- Key-level activity graphs
- 1-click integration between DynamoDB and CloudWatch

Key benefits

- Identify frequently accessed keys and traffic trends at a glance
- Respond appropriately to unsuccessful requests

Demo

CloudWatch contributor insights for DynamoDB

New!

Pricing

- $0.03 per million events
- Regardless of item size (kilobytes), read versus write, provisioned versus on-demand
- Pay as you go; activate or disable as needed

CloudWatch contributor insights for DynamoDB

New!

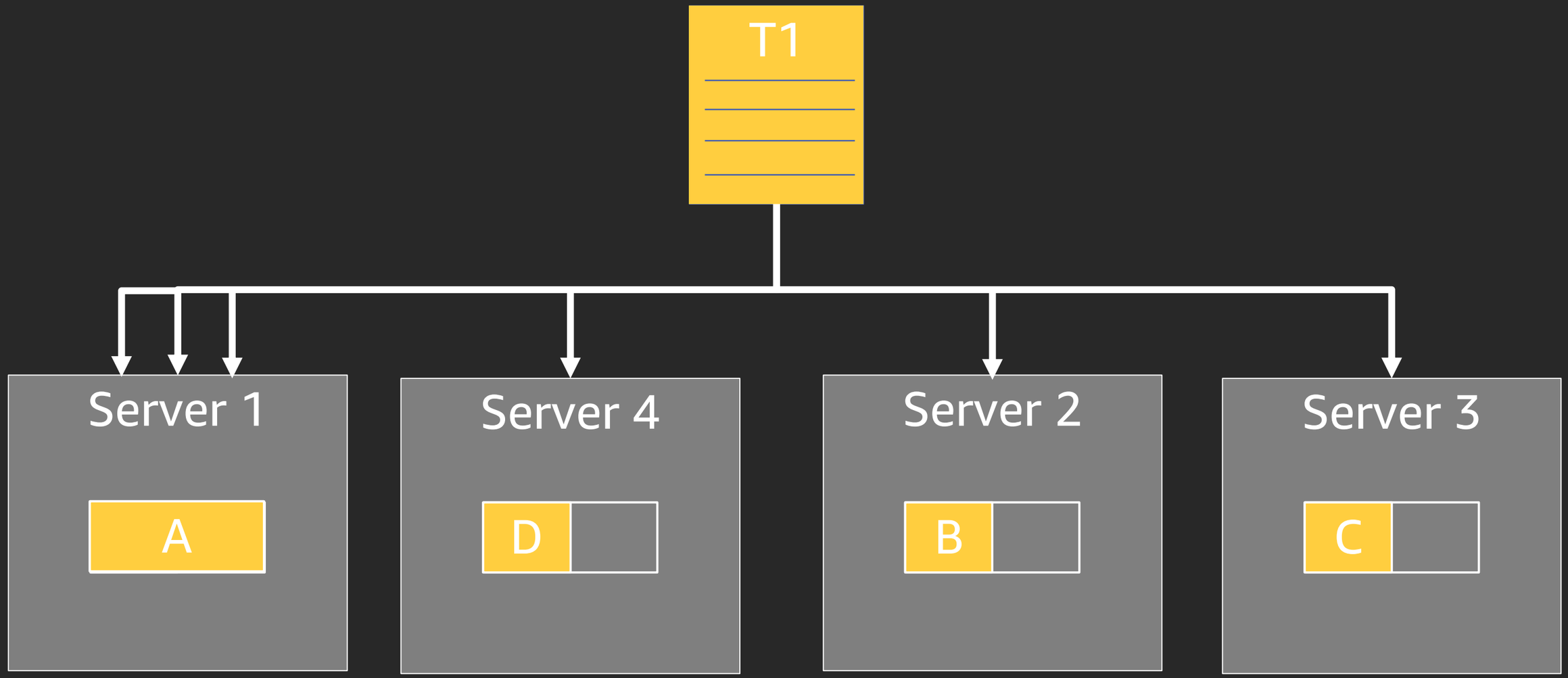
Additional features

- Customizable granularity and time window
- CloudWatch alarms and dashboard support
- GetInsightRuleReport API for programmatic retrieval

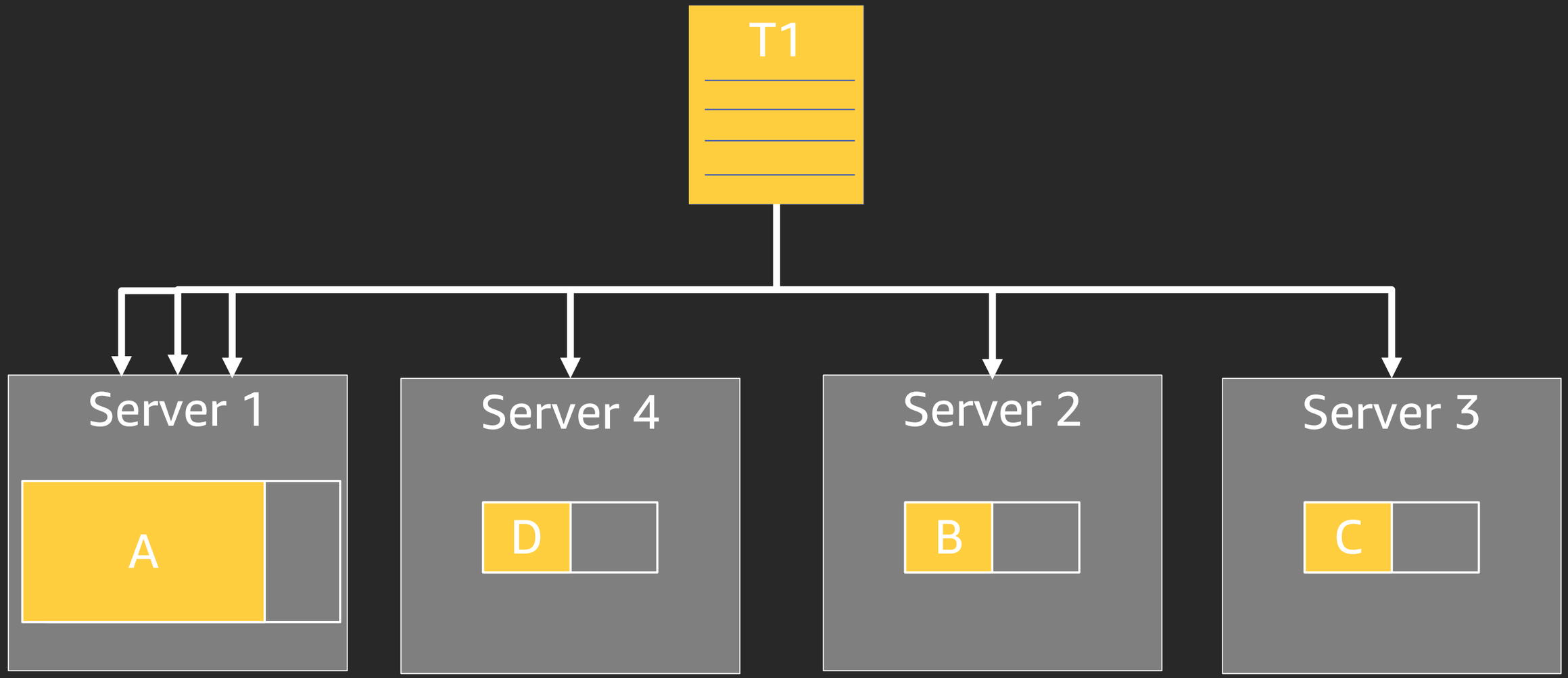
“What if I end up with too many partitions?”

Answer:
Partition count doesn’t matter because of instant adaptive capacity boosting

Adaptive capacity throughput boosting



Adaptive capacity throughput boosting



Instant adaptive capacity (new 2019)

Partitions respond instantly in response to changing traffic

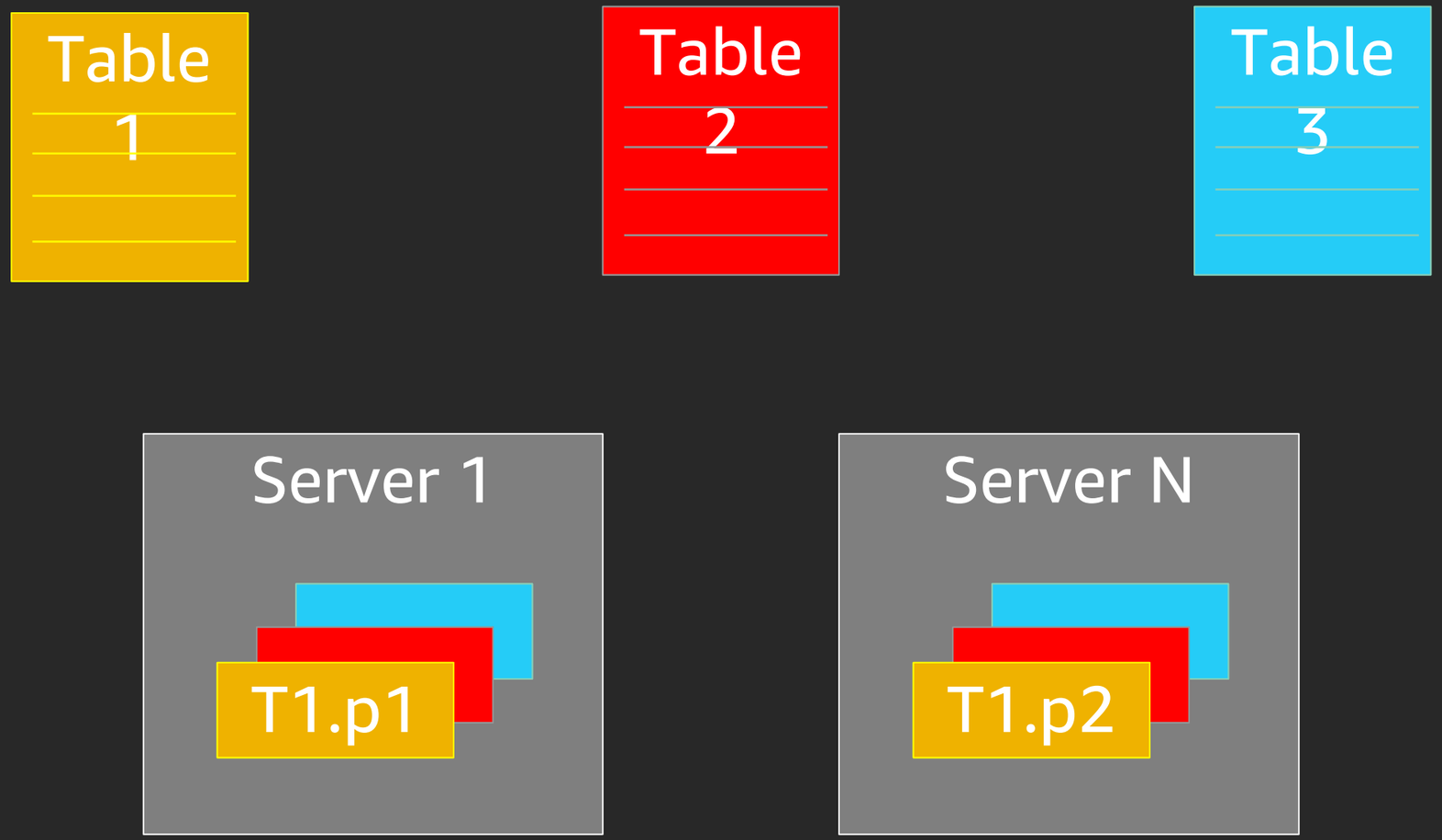
Instant adaptive capacity (new 2019)

DynamoDB looks at what your *table* can do, not what partitions can do

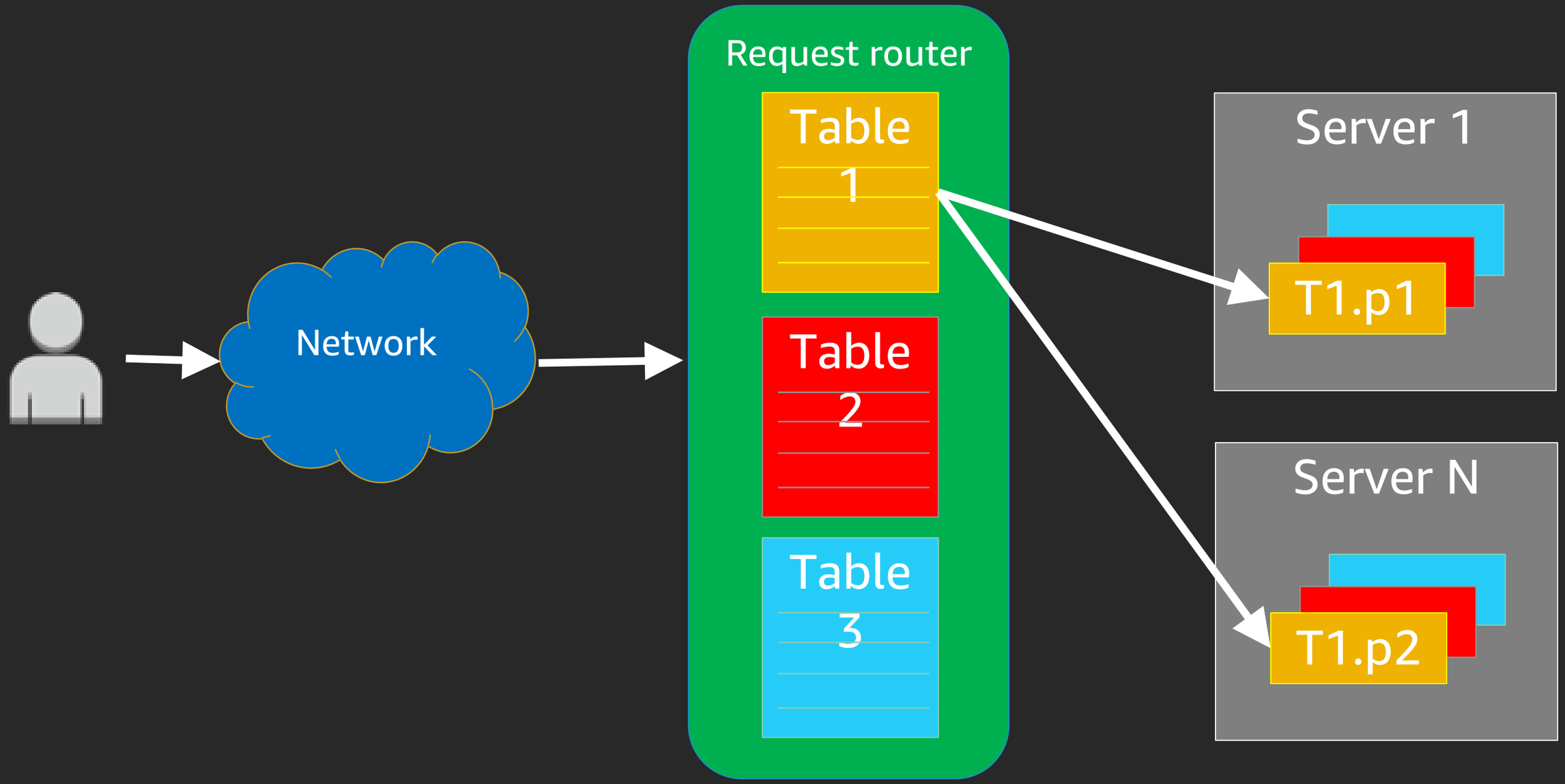
Key benefits

- Instantly accommodates imbalanced workloads
- Sustains imbalanced workloads indefinitely
- Helps you reduce your DynamoDB bill
- On by default, no extra cost

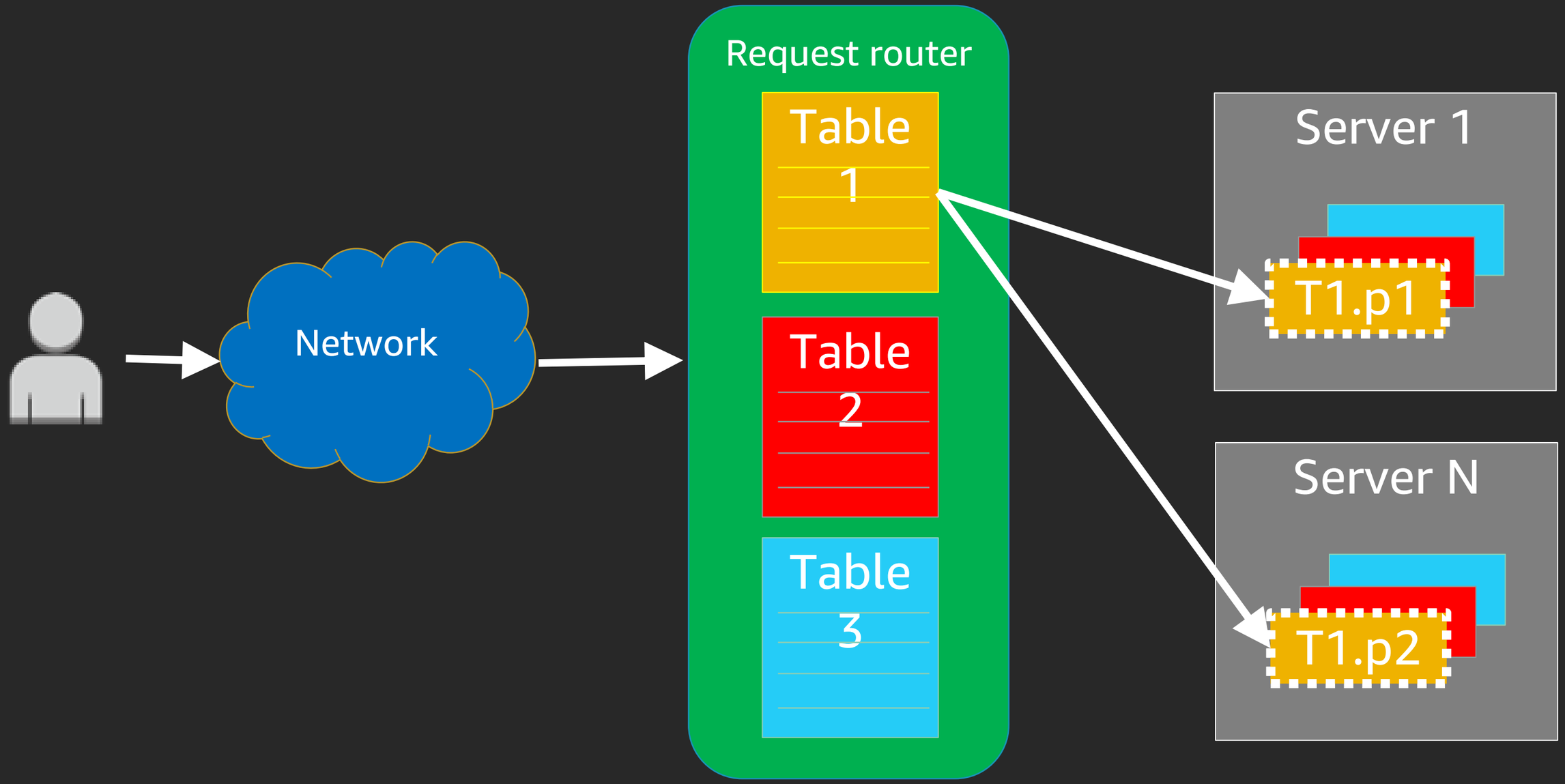
Instant adaptive capacity (new 2019)



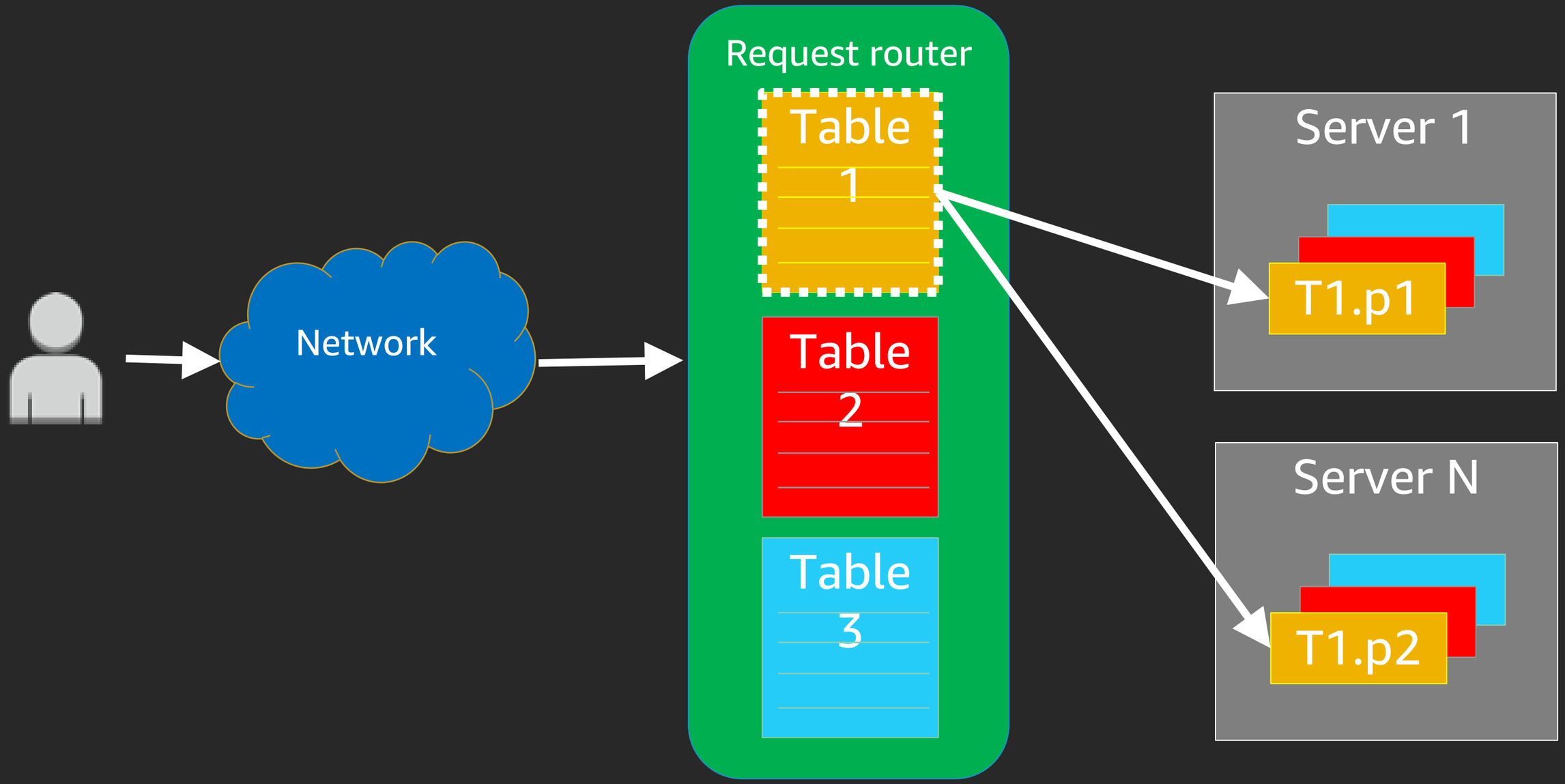
Instant adaptive capacity (new 2019)



Instant adaptive capacity (new 2019)



Instant adaptive capacity (new 2019)



Adaptive capacity recap

Scalability and performance even for imbalanced workloads

- Dynamic partitioning for storage and throughput
- Automatic isolation of frequently accessed items
- Automatic boosting if table is consuming less than provisioned

Partitions don’t matter, keys do

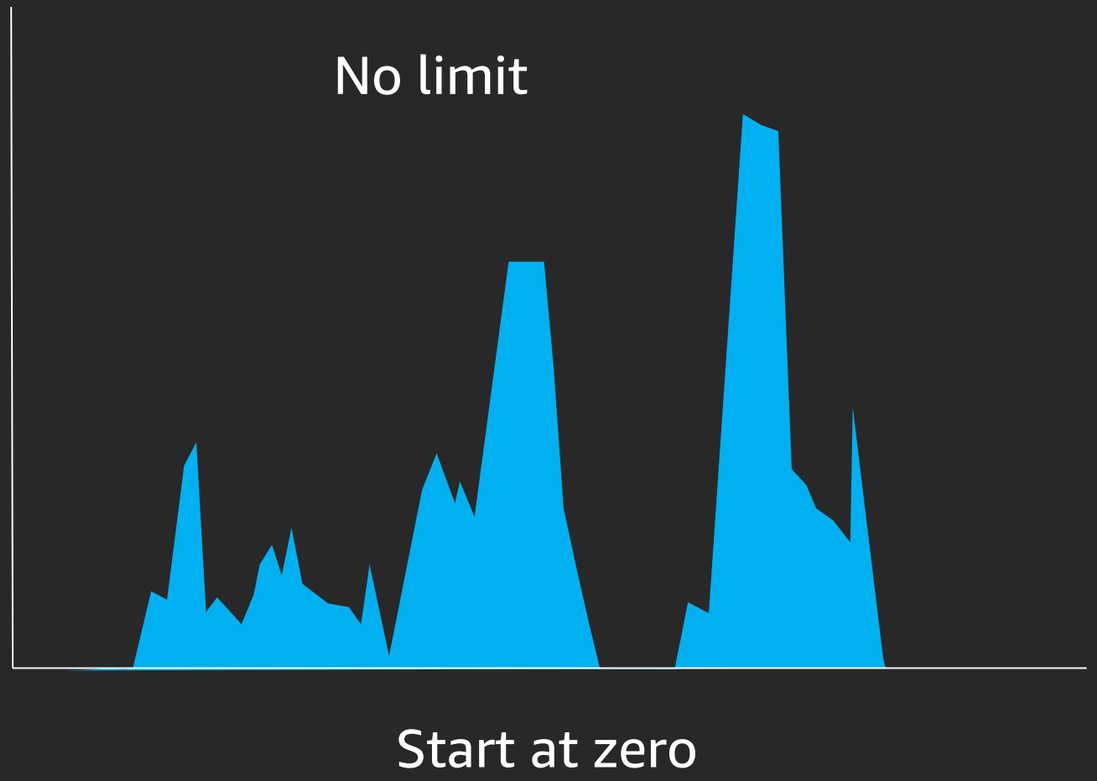
- Use CloudWatch contributor insights for DynamoDB to monitor keys

“How do I reduce my monthly bill?”

Provisioned versus on-demand

DynamoDB on demand

New!



Features

- No capacity planning, provisioning, or reservations—simply make API calls
- Pay only for the reads and writes you perform

Key benefits

- Eliminates trade-offs of overprovisioning or underprovisioning
- Instantly accommodates your workload as traffic ramps up or down

“What are the maximum scaling capabilities of on-demand mode?”

On-demand scaling properties

Starting throughput

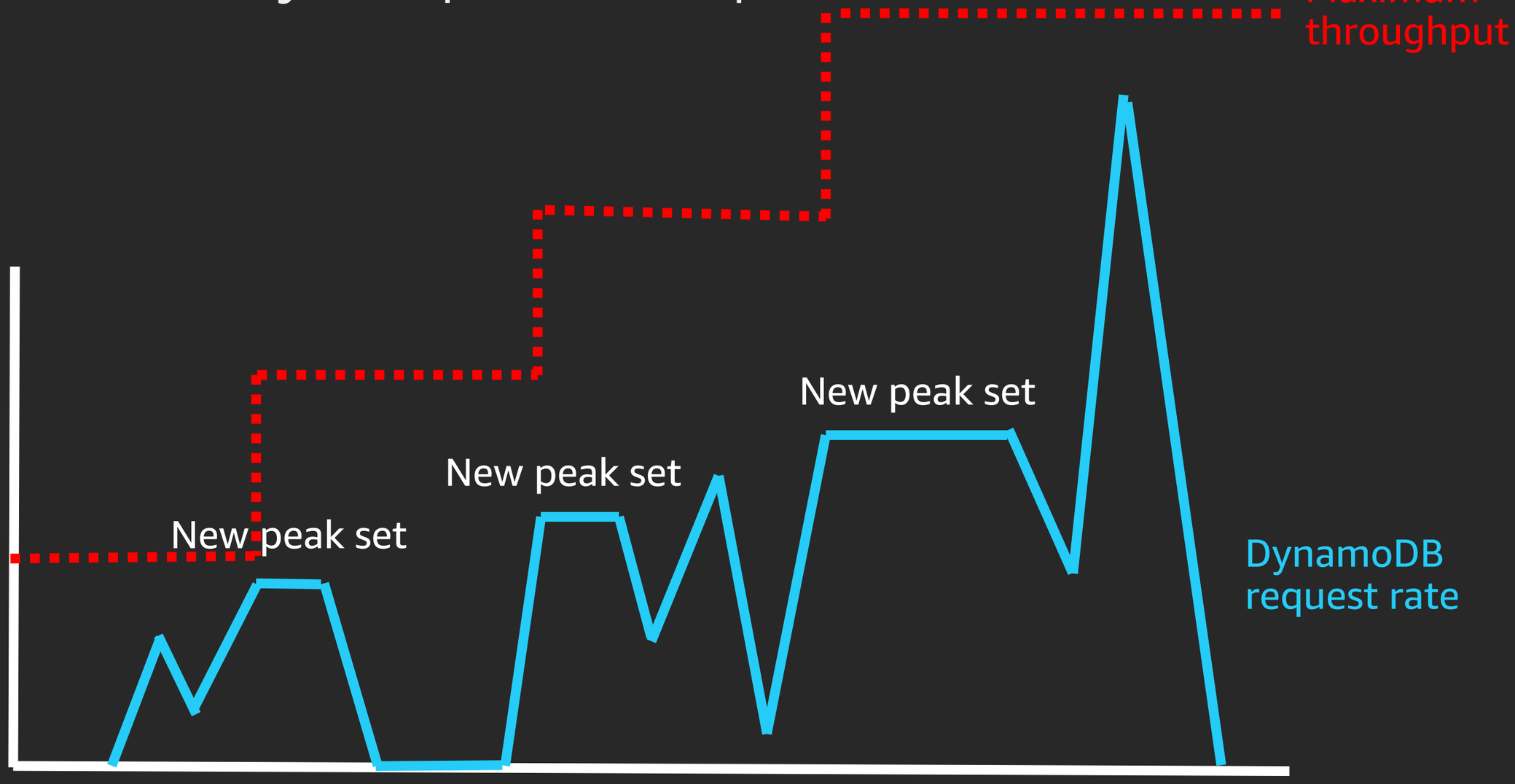
- Up to 4,000 write request units (WRUs): 4,000 writes per second
- Up to 12,000 read request units (RRUs): 24,000 EC reads per second
- Any linear combination of the 2

Maximum throughput

- Designed to be unlimited

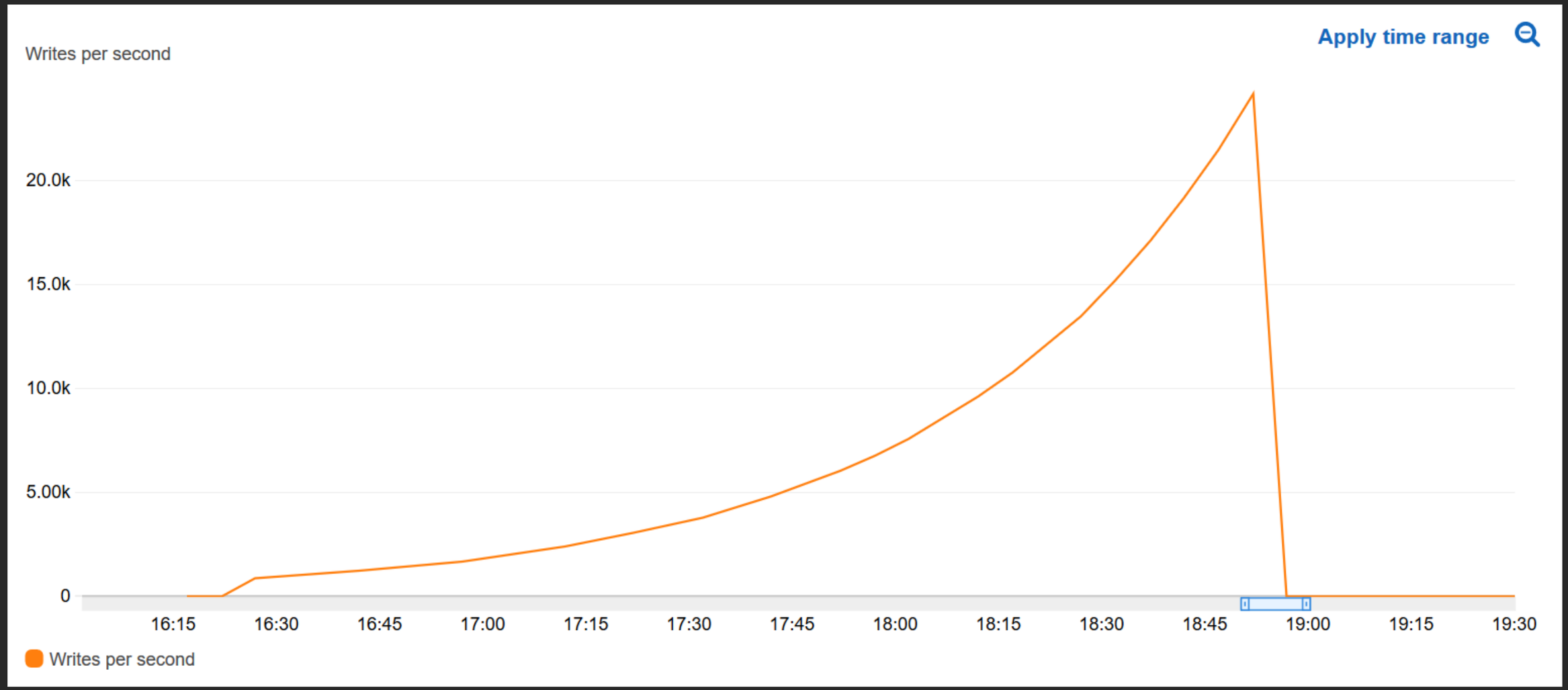
Pay per request: use nothing, pay nothing

“Up to twice your previous peak”

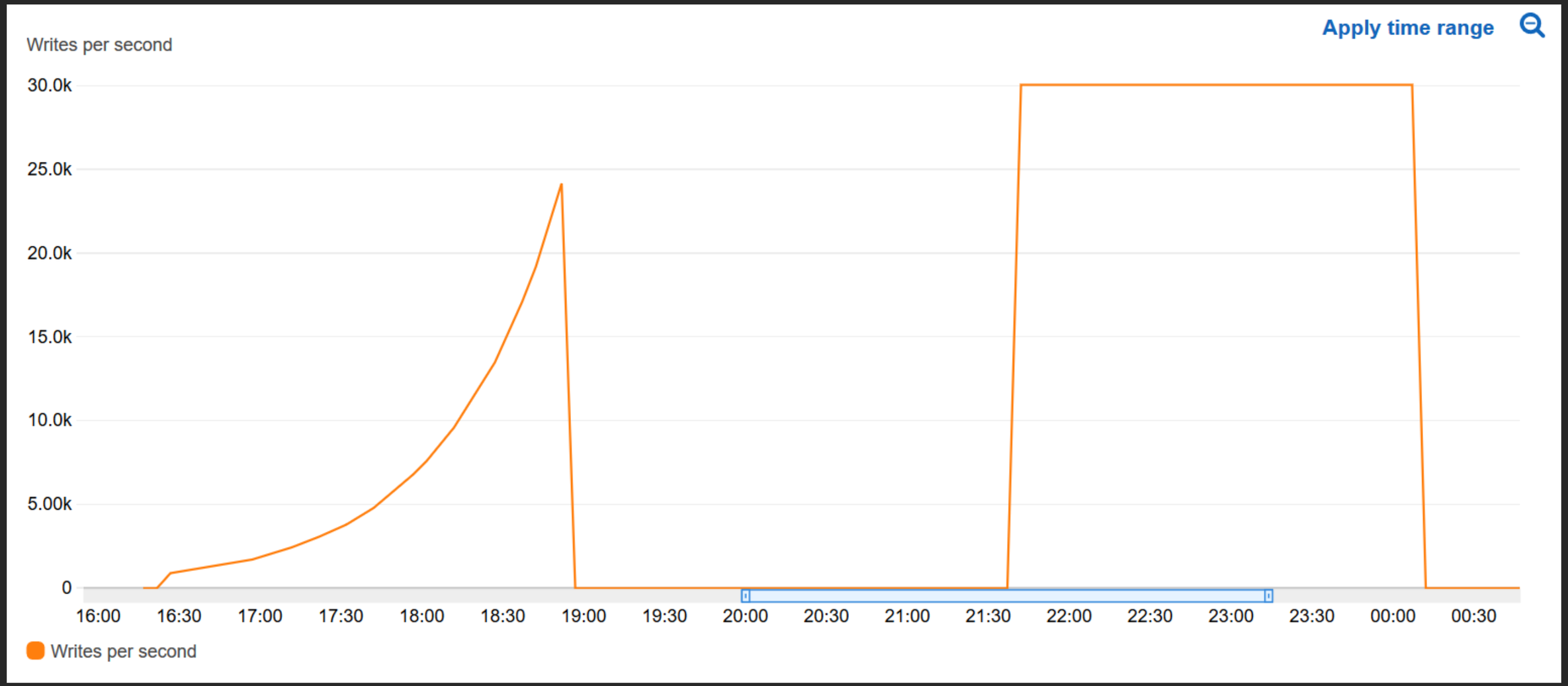


“Up to twice your previous peak”

Dynamic partitioning



Instant high throughput



Things to keep in mind

Initial doubling takes up to 30 minutes

Subsequent traffic scales *instantly*

Continuously monitored

- No need to hit 2x previous peak to get more throughput
- Gracefully accommodates growth without throttling

Provisioned versus on-demand

Time spent managing settings

Predictability

Utilization ratio

Mission critical

Confidence in demand forecast

Use provisioned mode

- Steady workloads
- Gradual ramps
- Events with known traffic
- Ongoing monitoring

Use on-demand mode

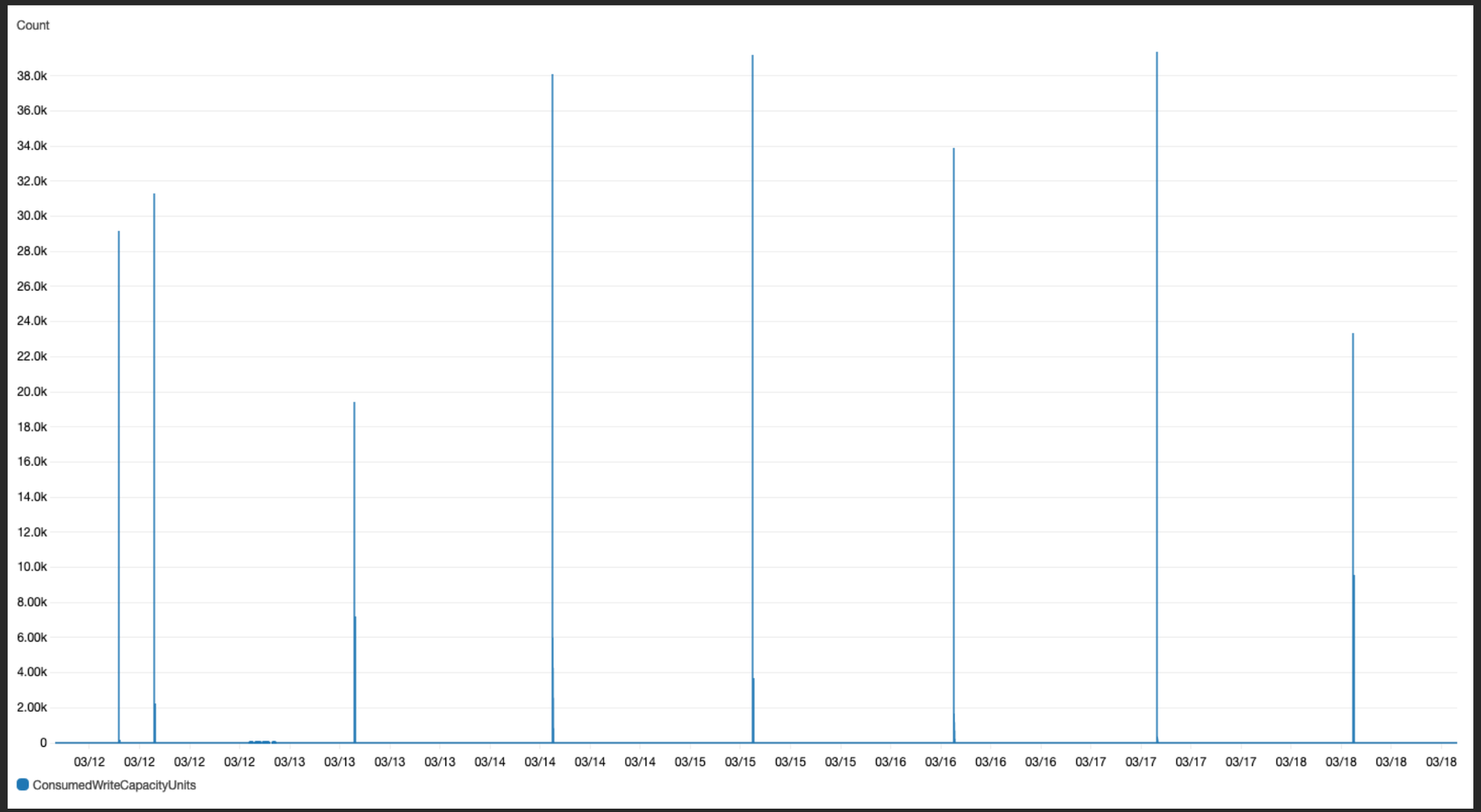
- Unpredictable workloads
- Frequently idle workloads
- Events with unknown traffic
- “Set it and forget it”

Consider your tolerance for operational overhead and overprovisioning

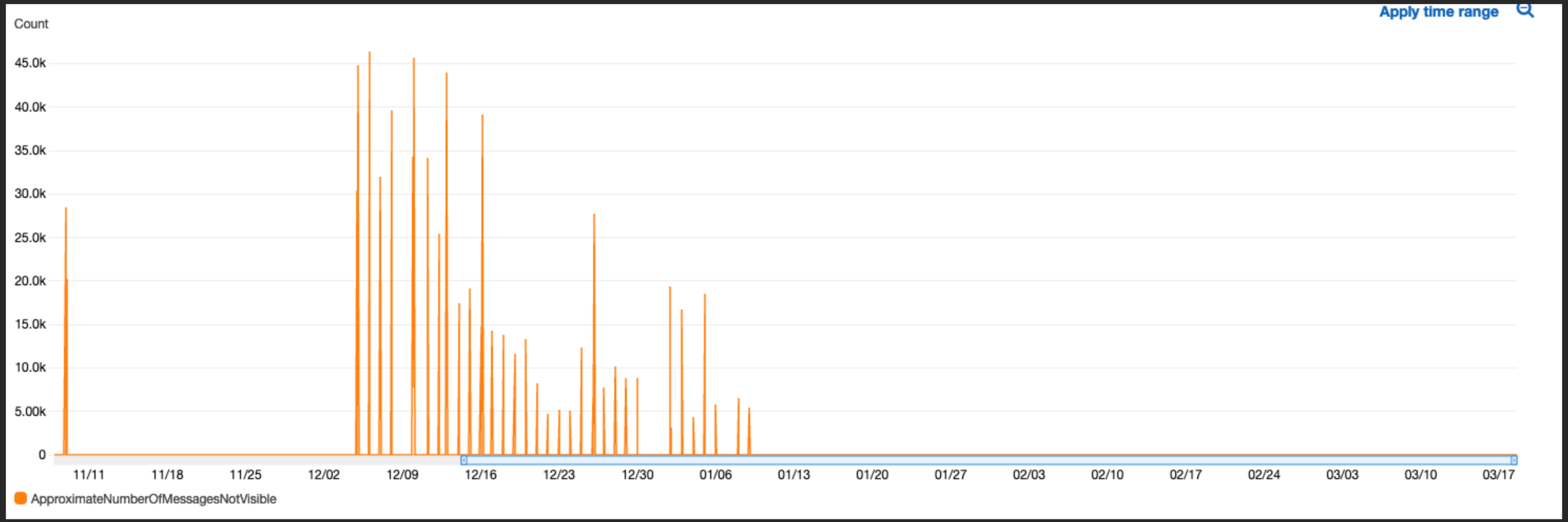
Example 1

Pure on-demand mode

Spiky workloads



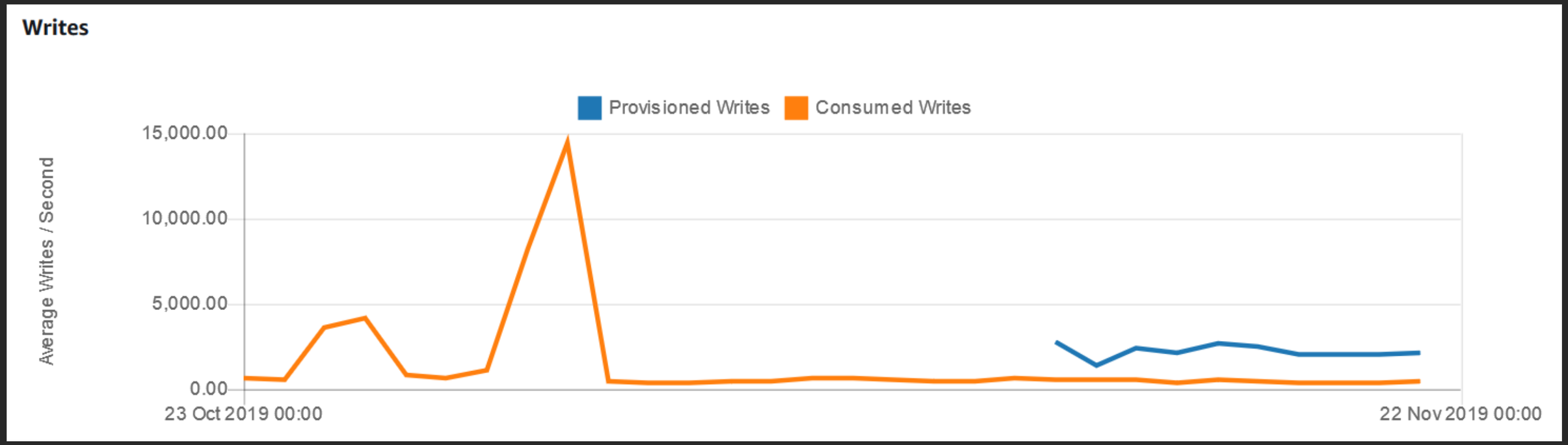
Business impact



Example 2

Post-launch switchover

Conversion after load stabilization



Key takeaway 3

Switch capacity modes in order to optimize cost and performance

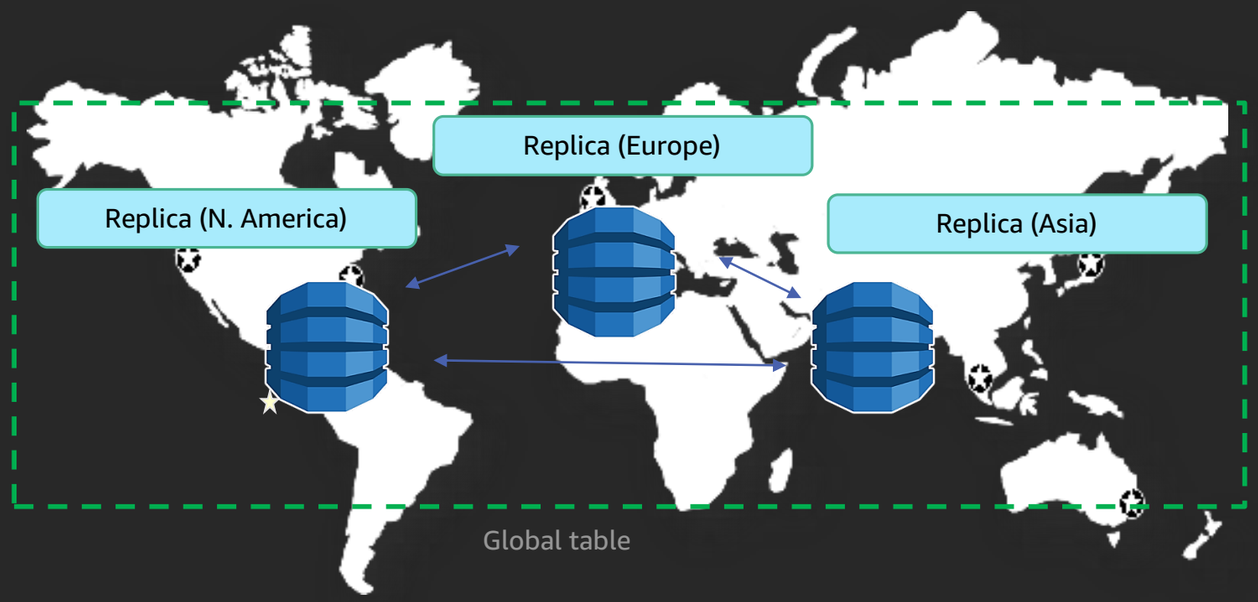
Global tables cost optimization

AWS re:Invent

aws

Enhancements to global tables

New!



Features

- Add regions on the fly
- Convert single-region tables to global
- Reduced write capacity consumption

Key benefits

- Flexible regional footprint
- Save on replicated write costs

Wrap up

Session recap

- DynamoDB accommodates your workload, not vice versa
- Partitions don’t matter, individual keys do
- Try CloudWatch contributor insights for DynamoDB
- Switch capacity modes in order to optimize cost and performance
- More flexible, cost-efficient global tables

Notable DynamoDB launches in 2019

Launch date	What was launched
November 26, 2019	Identify frequently accessed keys and database traffic trends with Amazon CloudWatch Contributor Insights for Amazon DynamoDB
November 26, 2019	Encrypt your Amazon DynamoDB data by using your own encryption keys
November 20, 2019	Convert your single-Region tables to global tables
November 15, 2019	Provision throughput capacity more efficiently now that DynamoDB isolates frequently accessed items automatically
September 16, 2019	Build high-performance data models with NoSQL Workbench for DynamoDB
August 5, 2019	Monitor account limits as you approach them
July 2, 2019	Delete a global secondary index before it finishes building
June 24, 2019	Process larger transactions now that DynamoDB supports up to 25 unique items and 4 MB of data per transactional request
May 23, 2019	Maintain uninterrupted performance indefinitely with instant adaptive capacity
April 26, 2019	Tag tables when you create them
February 3, 2019	Use transactional APIs, on-demand capacity mode, and 20 GSIs with DynamoDB local

Related sessions

DAT301 [Builder's session] Data modeling with Amazon DynamoDB in 60 minutes

---

DAT304 Scale fearlessly with Amazon DynamoDB adaptive capacity

---

DAT325 Amazon DynamoDB: Under the hood of a hyperscale database

---

DAT303 Data security best practices on Amazon DynamoDB

Related sessions

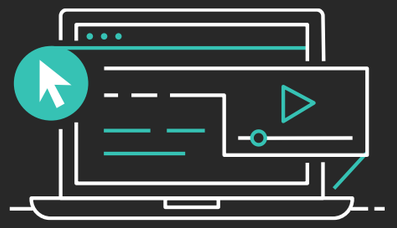
DAT319 How Uber stores financial transactions in ledgers using Amazon DynamoDB

DAT205 How Verizon Media implemented push notification using Amazon DynamoDB

DAT356 How Intuit executed a stream producer & transformer with Amazon DynamoDB

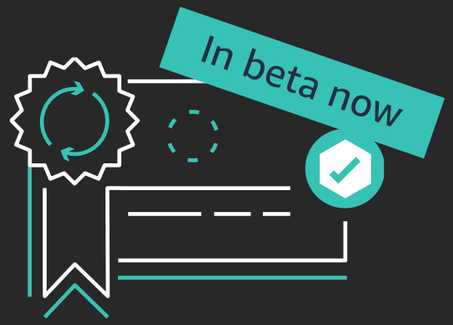
Learn databases with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate database skills



25+ free digital training courses cover topics and services related to databases, including:

- Amazon Aurora
- Amazon Neptune
- Amazon DocumentDB
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon Redshift
- Amazon RDS



Validate expertise with the new **AWS Certified Database - Specialty** beta exam

Visit aws.training

aws training and certification

Thank you!

Kai Zhao

kazh@amazon.com

AWS re:Invent

aws

Please complete the session survey in the mobile app.