AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

DAT324

Boost application performance with Amazon ElastiCache for Memcached

Karthik Konaparthi (he/him)
Senior Product Manager
In-Memory Database Services, AWS

Itay Maoz (he/him)
Head of Engineering
In-Memory Database Services, AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Data-modernization drivers

Importance of in-memory data stores and use cases

Amazon ElastiCache for Memcached

Customer stories and use cases

Demo: Caching use case

Q&A

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

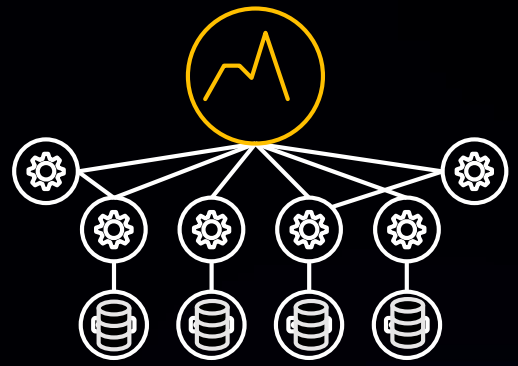
Data-modernization drivers

Explosion of data



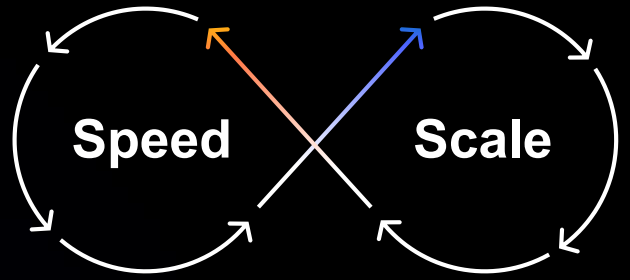
Data grows 10x
every 5 years

Microservices change data and application requirements



Growing need for multiple databases and data models within the same application

Applications require global scale and rapid response



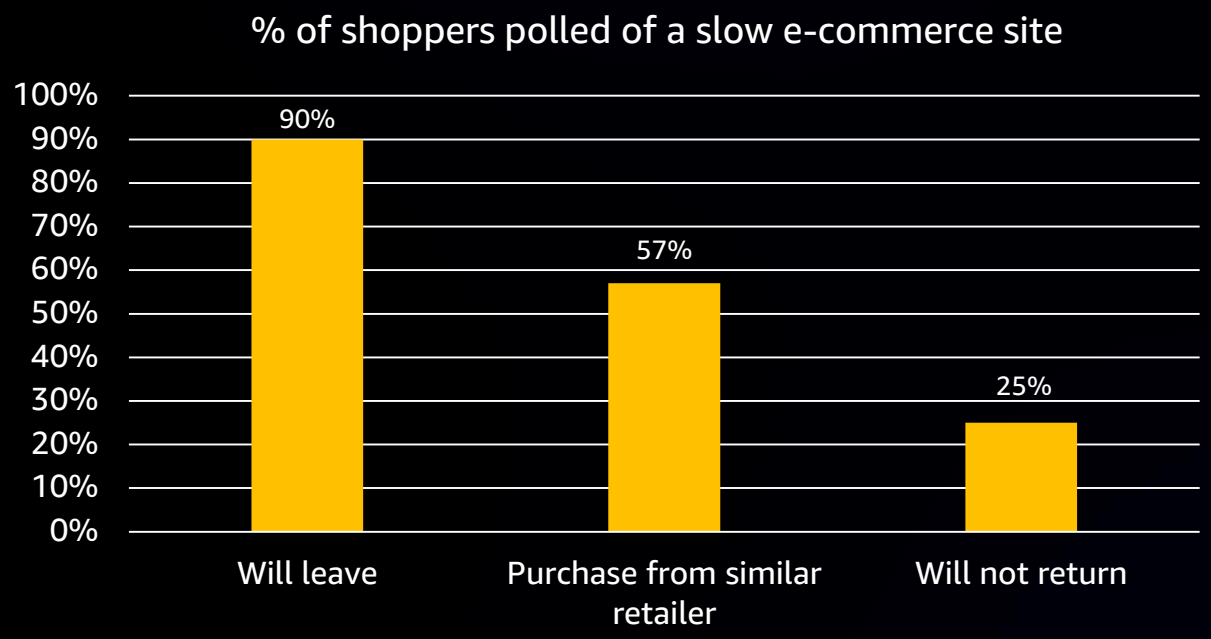
Time-to-insights and actions in milliseconds at internet scale

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

All applications can use more speed, many demand it

- FAST: Memory is at least 200x faster than SSDs



"A 100-millisecond delay in website load time can hurt conversion rates by 7%"

"A 2-second delay in web page load time will double the number of customers leaving the website"

– Akamai study[1]

1 https://www.akamai.com/newsroom/press-release/akamai-releases-spring-2017-state-of-online-retail-performance-report

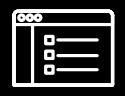
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Modern application requirements

REQUIRE MORE PERFORMANCE, SCALE, AND AVAILABILITY





Ecommerce



Media streaming



Social media



Online gaming



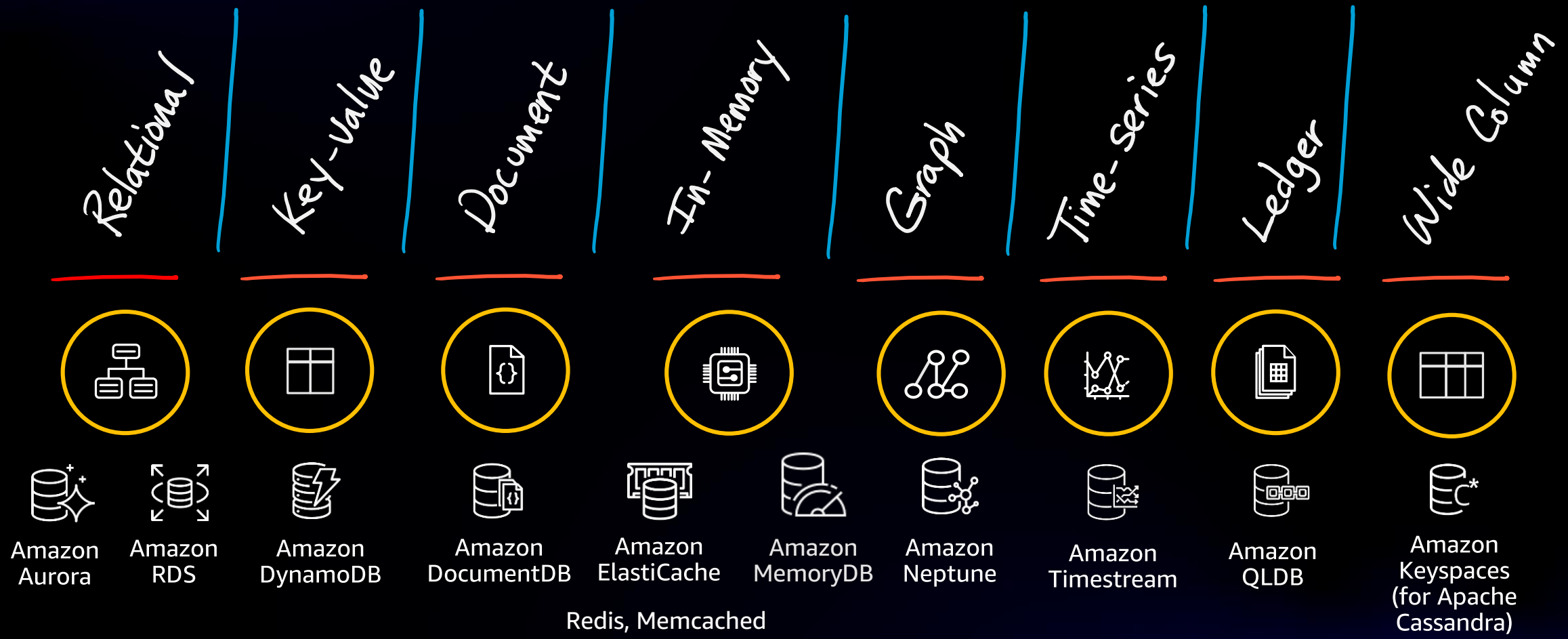
Shared economy

Users	Millions
Data volume	Terabytes – Petabytes
Locality	Global
Performance	Microsecond latency
Request rate	Millions per second
Access	Web, mobile, IoT, devices
Scale	Up-down-out-in
Economics	Pay-as-you-go
Developer access	Open API

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Purpose-built databases

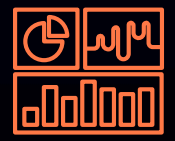


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use cases for in-memory data stores



Caching



Real-time analytics



Gaming leaderboards



Geospatial



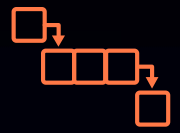
Media streaming



Session store



Chat apps



Message queues



Machine learning

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon ElastiCache for Memcached

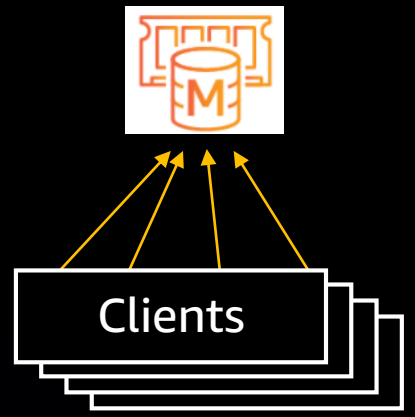
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is Memcached?

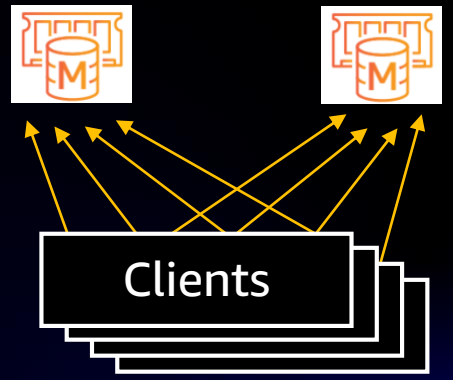
Initially released in 2003, Memcached provides:

- Simple, in-memory, LRU cache
- Simple key-value (string-string) store
- Supports strings, objects
- Multi-threaded
- Sharding via client-side library
- Easy to scale
- No persistence
- TCP and UDP protocols
- Open source

Single-node instance



Sharded instance



aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

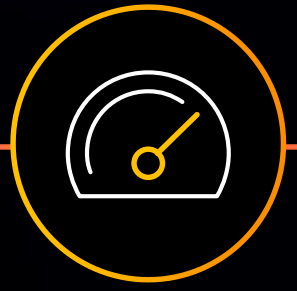
Amazon ElastiCache for Memcached: Fully managed service

Memcached compatible



Fully compatible with open source Memcached

Extreme performance



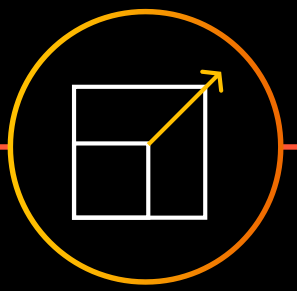
In-memory caching service and data store for high throughput and microsecond response times

Fully managed



AWS-managed hardware and software setup, configuration, monitoring, Auto Discovery

Easily scalable



Scale requests and memory with up to 80 nodes and 49.65 TiB per cluster

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use cases: Caching



Challenge

Retrieving data from disk-based databases and applications is slow. Workarounds such as local caching and proprietary solutions are expensive, don't scale, and difficult to manage

Solution

Use ElastiCache for Memcached, an in-memory cache, to store frequently used data in memory for microsecond latency and high throughput

Use cases

Database query caching and full web page caching

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use cases: Session store



Challenge

Need low latency and high concurrency data stores to process fast-moving ephemeral session data such as authentication tokens, cookies, and session state

Solution

Use ElastiCache for Memcached as an in-memory key-value store with key expiration strategies to store and manage ephemeral session data for microsecond responses

Use cases

Web and mobile apps

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon ElastiCache for Memcached customers

Software and internet



zoom

ATLASSIAN

reddit

sumologic

AUTODESK

Travel

airbnb

Expedia

Financial services

robinhood

Bitstamp

coinbase

Gaming

zynga

EPIC GAMES

Nintendo

Social media

Snap Inc.

ancestry

Telecommunication

twilio

docomo

Other

Peloton

instacart

duolingo

Media and entertainment

A+E NETWORKS

The Walt Disney Company

ALBMA

hulu

BuzzFeed

Logistics and operations

esri

Industrials and services

Sabre

dedalus
mais que cloud computing

Publishing

PBS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon ElastiCache for Memcached: Why use over Redis?

- Avoid code refactoring, reuse of libraries
  - Many apps use Memcached today (API familiarity)
- Better price/performance for certain use cases
  - Memcached is multi-threaded, can handle more operations by scaling up compute capacity
- Simpler (limited data structures support)
- Built-in check and set for preventing race conditions

More detailed comparison: Redis vs. Memcached

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon ElastiCache for Memcached: Data types and commands

- Two data structure types – string and integer
- Commands: ADD, APPEND, CAS, DECR, DELETE, FLUSH_ALL, GET, GETS, INCR, PREPEND, REPLACE, SET, STATS
- CLI example set/get operation:

  SET mykey 0 60 5 [Enter]
  input: 12345 [Enter]
  output: STORED

  GET mykey [Enter]
  output: VALUE mykey 0 5
  output: 12345

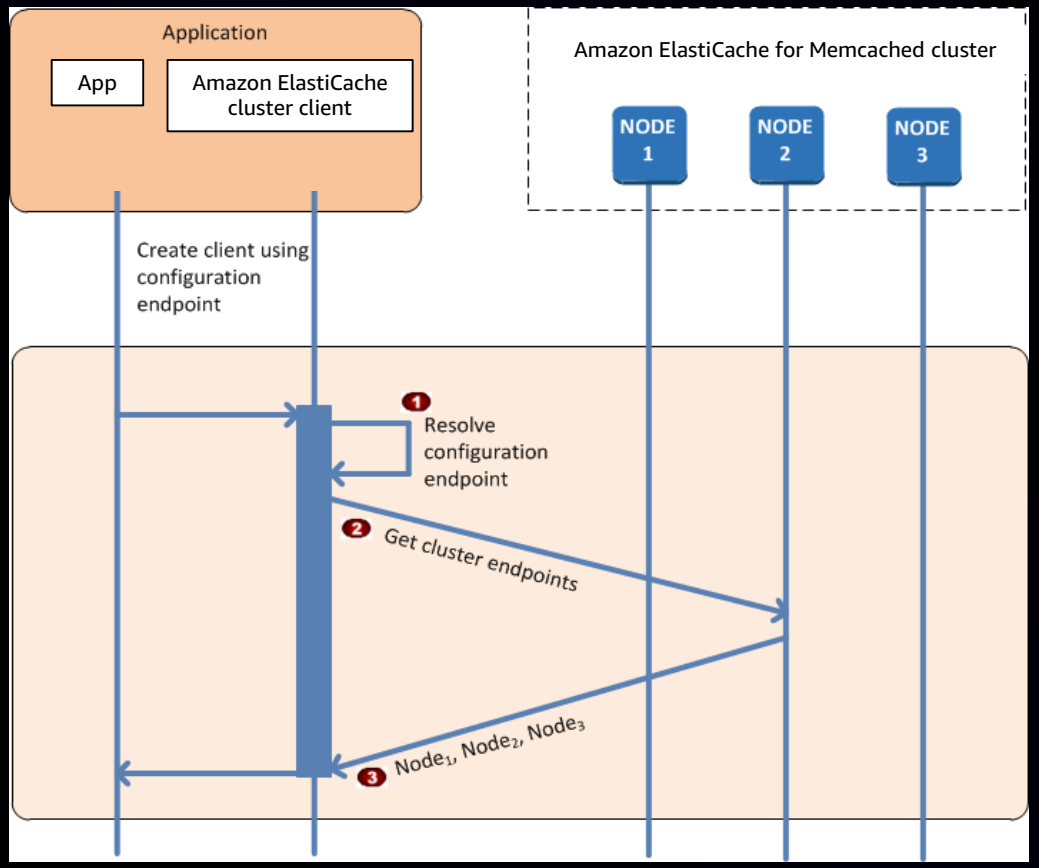
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon ElastiCache for Memcached: Auto Discovery

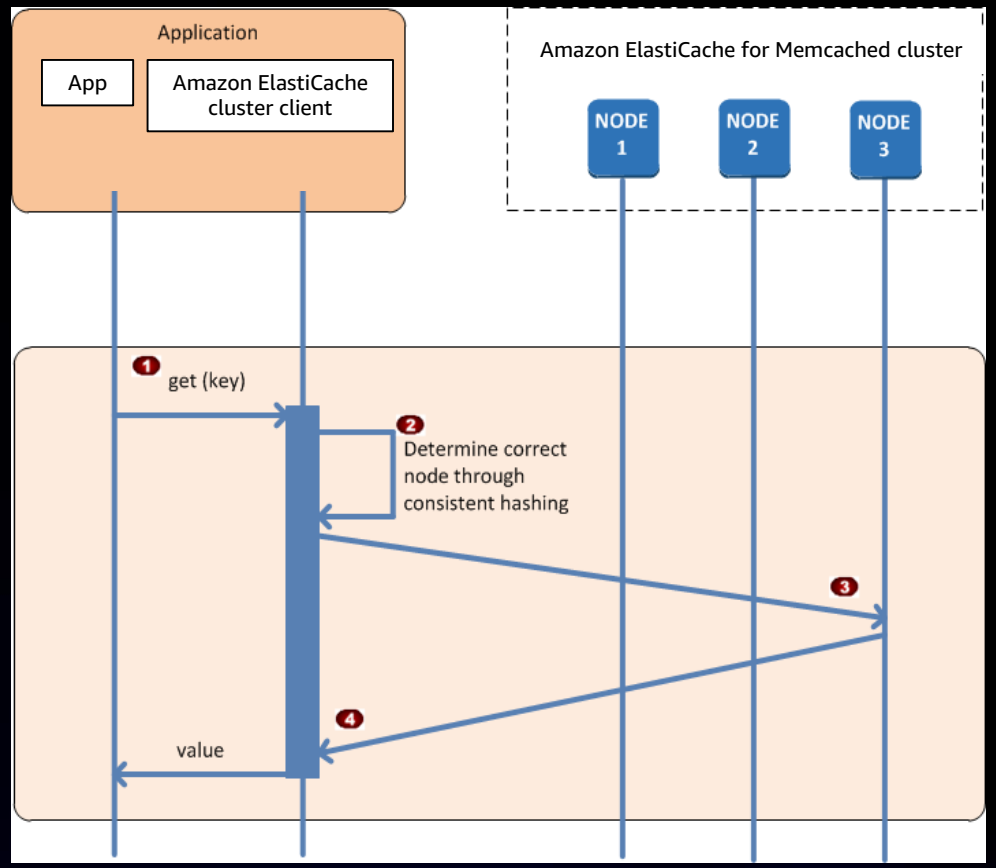
- Client application connects to the cluster endpoint instead of hard coded nodes in the cluster
    - Application automatically retrieves the nodes from main configuration endpoint
    - Node increase/decrease operations are ”seen” by the Auto Discovery client – typically polling the cluster once per minute
    - Supported by Amazon ElastiCache–provided Memcached client libraries

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon ElastiCache for Memcached: Auto Discovery



Initial connection

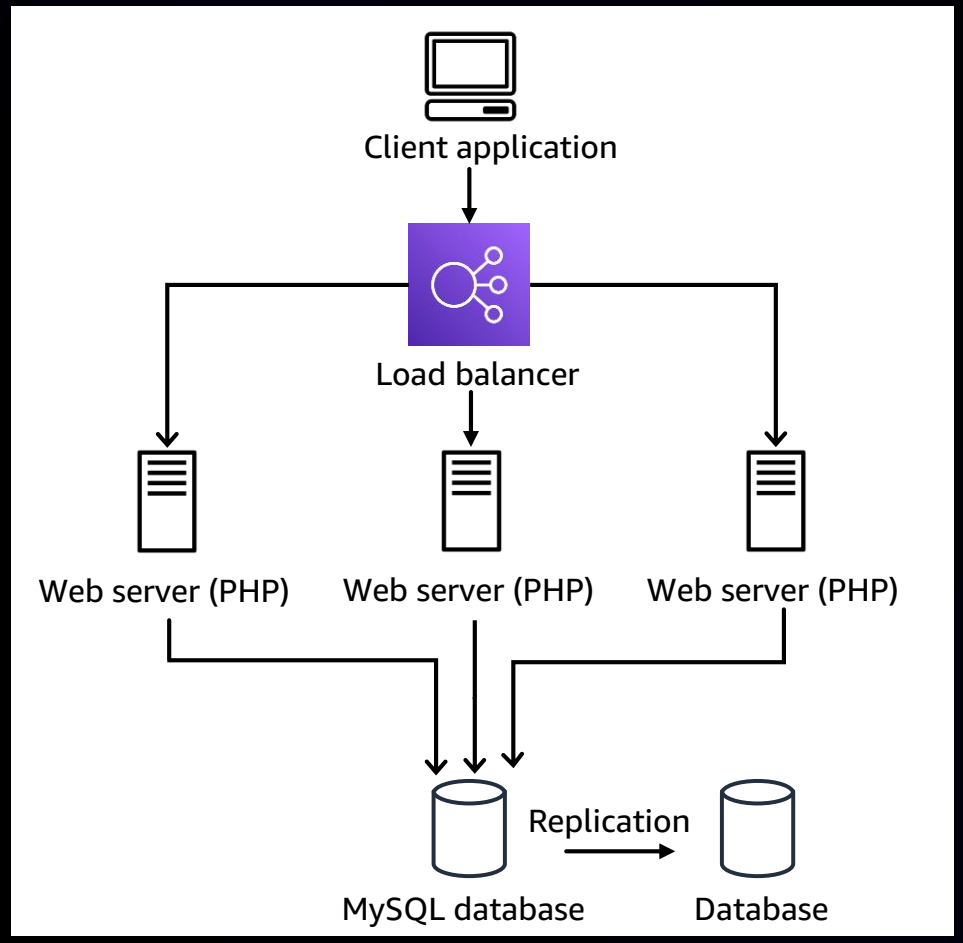


Normal cluster operations

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Seconds to microseconds

Reduce latency of a web application with just a few lines of code



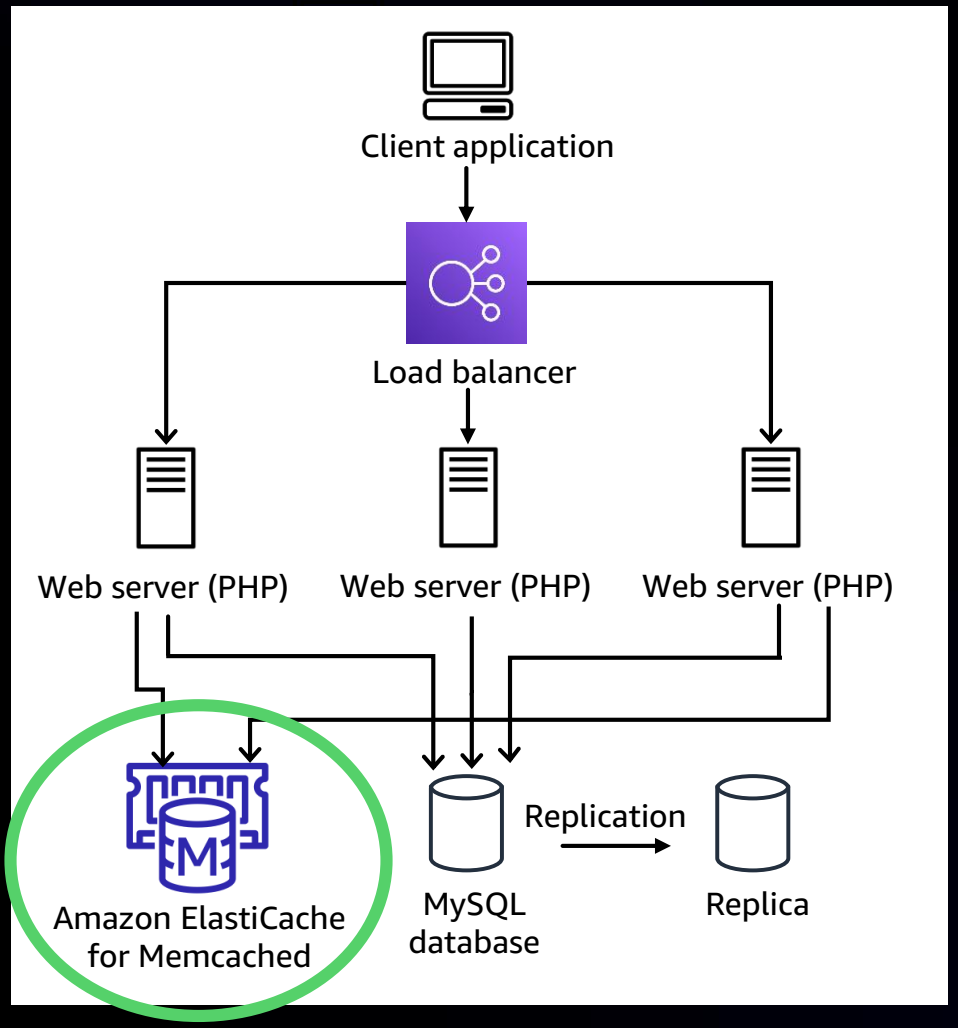
Latency: up to **seconds**

With a large number of items, the database becomes a bottleneck, and high latency causes customers to leave the website

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Seconds to microseconds

Reduce latency of a web application with just a few lines of code



Latency: **microseconds**
Throughput: **8x (and more)**
Cost Savings: **up to 87%**

With ElastiCache for Memcached, latency goes down drastically

Simple lazy load solution

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Seconds to microseconds

1. Using lazy load pattern to reduce latency of an expensive SQL query

Number of rows in MySQL table	Render time without cache (ms)	Render time with Amazon ElastiCache (ms)	Performance boost
10,000	12	0.24	50x faster
100,000	118	0.24	492x faster
1,000,000	1,167	0.24	4,862x faster
10,000,000	11,757	0.24	48,987x faster

Test Setup:

- Emulating an expensive SQL query (table scan) for finding 10 items
- Repeating with lazy load pattern using Amazon ElastiCache for Memcached

Configuration:

- RDS MySQL: m6g.4xlarge
- Amazon ElastiCache for Memcached: m6g.xlarge
- Direct connectivity between web server and cache/database

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Unlimited scaling, 87% cost savings

2. Using lazy load pattern to maximize throughput and reduce cost

RDS MySQL instance	Amazon ElastiCache instance	Max RPS	Cost/1M requests
m6g.xlarge	No cache	28,192	$0.180
m6g.2xlarge	No cache	57,536	$0.176
m6g.4xlarge	No cache	133,008	$0.152
m6g.8xlarge	No cache	270,109	$0.150
m6g.12xlarge	No cache	423,669	$0.144
m6g.16xlarge	**No cache**	**535,642**	**$0.151**
Adding cache			
m6g.2xlarge	m6g.xlarge	570,948	$0.026
m6g.2xlarge	m6g.2xlarge	1,013,824	$0.020
m6g.2xlarge	4 m6g.xlarge	1,609,591	**$0.019**
m6g.2xlarge	4 m6g.8xlarge	**4,054,426**	**$0.042**

Takeaways:

- Scaling up MySQL **is limited**; Amazon ElastiCache's vertical **& horizontal** scaling enables high throughput
- At high throughput, Amazon ElastiCache can improve price/performance by up to **87%**
- **With few lines of code,** you can significantly boost your application performance with Amazon ElastiCache

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Resources

Resource	Link
Product page	Amazon ElastiCache for Memcached
AWS Caching page	AWS Caching overview
Documentation	Amazon ElastiCache for Memcached user guide
Product video	Introduction to Amazon ElastiCache for Memcached
AWS community forum	Amazon ElastiCache community forum
Demo	Session management using Amazon ElastiCache for Memcached

Your AWS customer account team is happy to discuss your solution requirements, bring in solution architect experts, and ensure your success with Amazon ElastiCache

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Karthik Konaparthi
konapk@amazon.com

Itay Maoz
itay@amazon.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.