Optimize Amazon RDS and Aurora costs with Amazon ElastiCache for Redis

Ed Cotton

WW Sr GTM, In Memory Databases

Roberto Luna Rojas

WW Sr Specialist Solution Architect, In Memory Databases

Amazon ElastiCache - Fully Managed Service

In-memory data store

Redis

Extreme performance

Fully managed

Scalable

Versatile

Universal Cache

Financial advantages of caching RDS/Aurora

"Save up to 55% in cost and gain up to 80x faster read performance using ElastiCache with RDS for MySQL (vs. RDS for MySQL alone)."

- Significantly lower cost than scaling RDS for the same workload
- Only pay a per instance fee and no separate I/O charge
- It can scale independently from RDS at a lower cost
- ElastiCache optimizations such as Enhanced I/O multiplexing, obtain up to 72% better price/performance, and with data tiering, obtain up to 60% better price/capacity

How ElastiCache Saves You Money

Extreme performance at a fraction of the cost

Data Tiering

Enhanced I/O Multiplexing

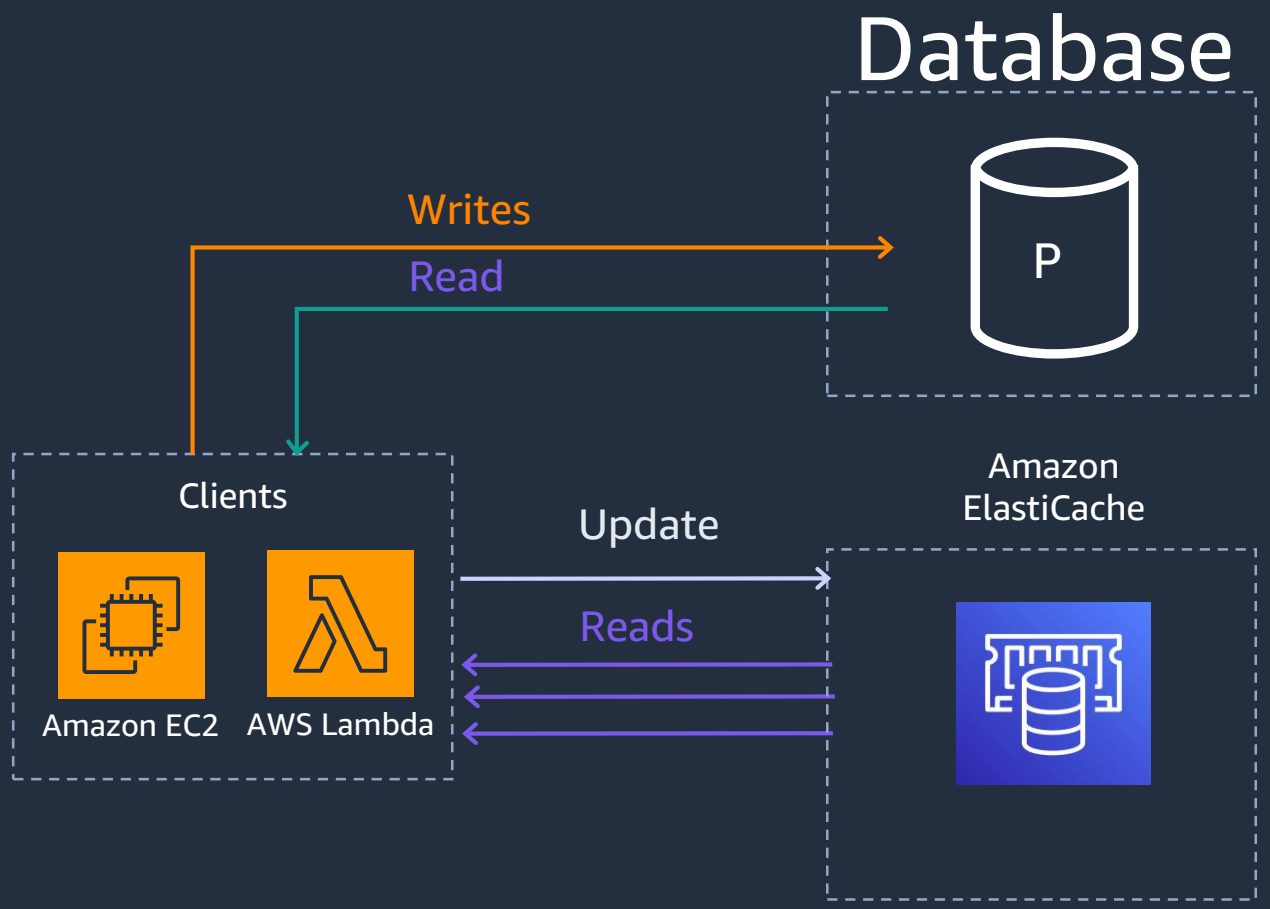
Global Datastore

Reserved Instances

No Transaction Charges

Adding ElastiCache to your RDS/Aurora Workloads

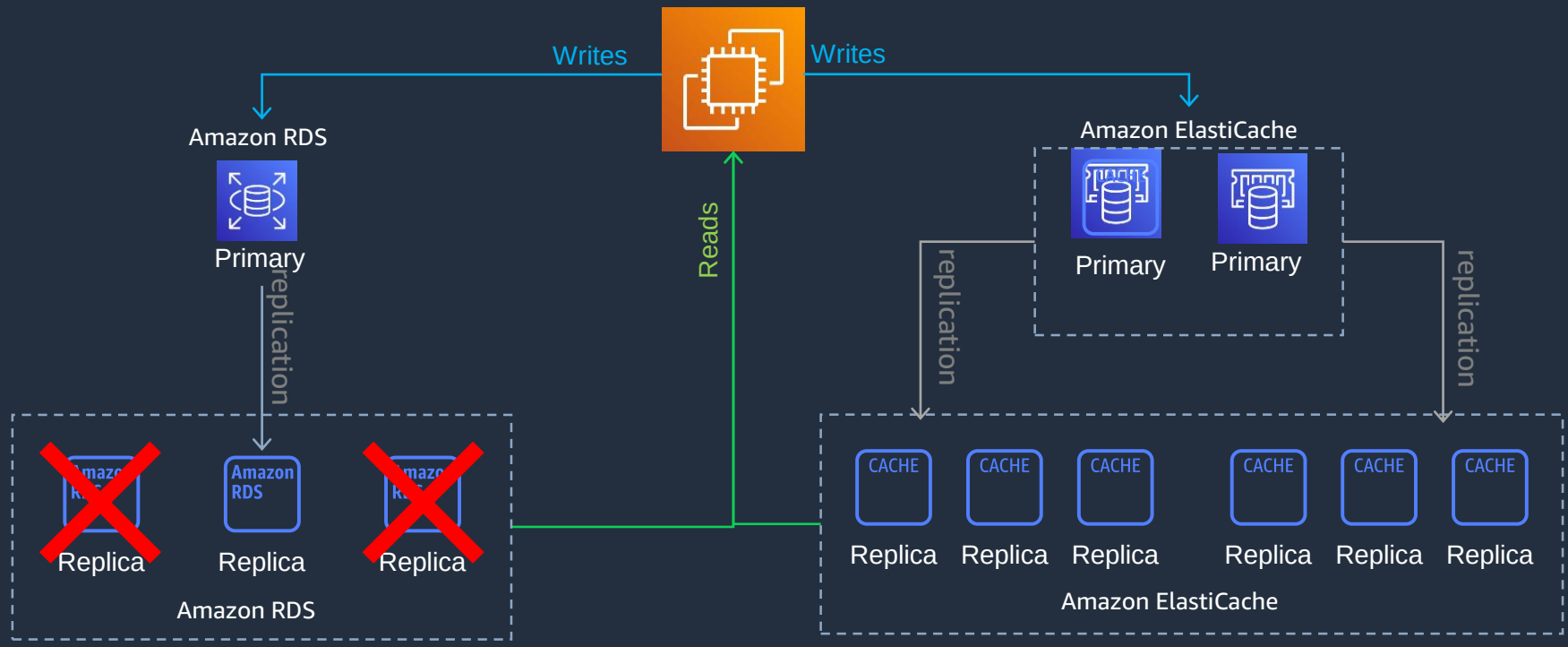
Adding Amazon ElastiCache



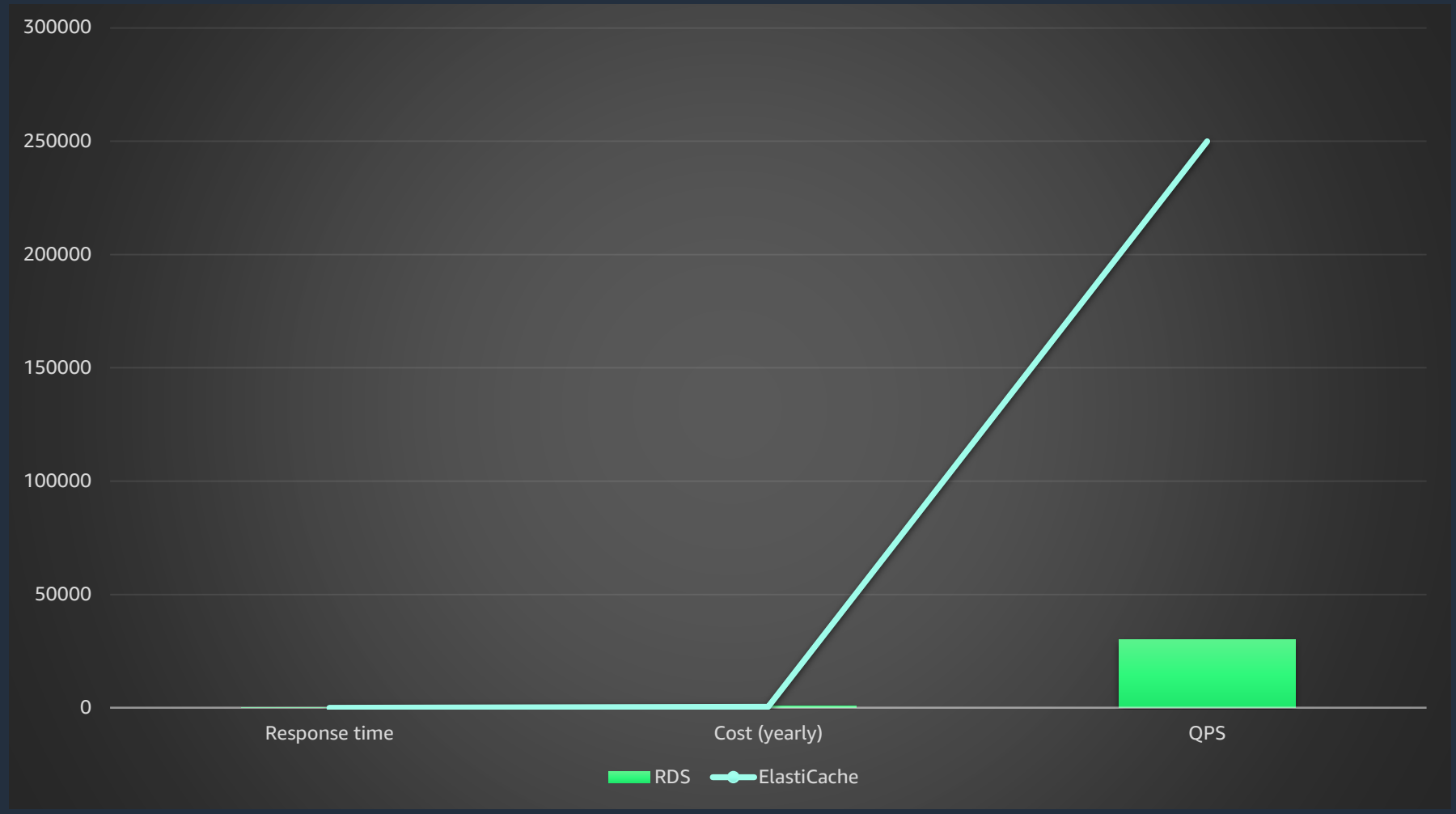
Adding a caching service

- Improves latency by 80x
- Increased read capacity by 4x
- Scales vertically and horizontally
- Scales to multiple regions
- Fraction of DB replication cost
- Reduces database sprawling

Reduce read-replica footprint and save costs with ElastiCache



Better cost-performance-throughput ratio with ElastiCache



Lazy Loading or Cache Aside

```
> SELECT
    customers.name, SUM(orders.num_items)
  FROM
    customers, orders
  WHERE
    customers.id = X AND
    orders.customer_id = customers.id
```

MD5 Hash

"ecf361704f15aa0e26e3b24e1ce6d1d6"

Key

ecf361704f15aa0e26e3b24e1ce6d1d6

~32 bytes

Value

"\x80\x03M\xd2\x06cdecimal\nDecimal\n…"

Demo

Caching on Python using FastAPICache with Redis backend

```
@app.get("/passenger/{passenger_id}")
@cache(expire=60)
def read_passengers(passenger_id: int, request: Request,
response: Response):
     with Session(engine) as session:
          passenger = session.get(Passenger, passenger_id)
          if not passenger:
               raise HTTPException(status_code=404,
detail="Passenger not found")
          return passenger
```

Getting Started with ElastiCache for Redis

Amazon ElastiCache for Redis

Getting Started with Amazon ElastiCache

Boost Your PostgreSQL Database Performance with Amazon ElastiCache for Redis

Thank you!