

Reducing the Application Deployment Costs Via Caching

In this lesson, we will discuss a real-world example of how the deployment cost of an application can be reduced by using a cache.

We'll cover the following

- Real Life Use Case
- Conclusion

Real Life Use Case

In this lesson, I am going to share an insight from a stock market-based gaming app that I developed and deployed on the cloud.

The game had several stocks of companies listed on the stock market and the algorithm would trigger the price movement of the stocks every second, if not before that.

Initially, I persisted the updated prices of the stocks in the database as soon as the prices changed, to create a stock price movement timeline at the end of the day. But so many database writes cost me a fortune. The number of writes every hour was just crazy.

Eventually, I decided to not persist the updated price every second in the database rather use *Memcache* to persist the stock prices. And then run a batch operation at regular intervals to update the database.

Memcache was comparatively a lot cheaper than the disk-based database access. The cache served all the stock price requests, & the database did not have the updated values until the batch operation ran.

Conclusion

This tweak may not be ideal for a real-life Fintech app but it helped me save a truck-load of money & I was able to run the game for a longer period of time

track load of money & I was able to run the game for a longer period of time.

So, this is one instance where you can leverage the caching mechanism to cut down costs. You might not want to persist each and every information in the database rather use cache to store not so mission-critical information.

Now let's look into some of the caching strategies we can leverage to further enhance the performance of our apps.