Predictive Auto Scaling of In-Memory Data Stores

TEAM MEMBERS:
- AKSHAY RAUL
- RAHUL IYER
- SHWETHA KALYANARAMAN
Motivation

- Caching is a major part of many application infrastructures
- Service needs to be resilient, reliable, and scalable
- Costly to maintain multiple nodes to service clients
- Auto-Scaling to the rescue!
What are In-Memory Data Stores?

- Type of non-relational database
- Relies on memory for storage
- Helps in minimizing response time
Introduction to Redis

- In-Memory data store
- Simple key value store
- Supports sharding and replication
- Lightweight and multi-platform
- Provides a lot of performance metrics
Setup Infrastructure: AWS

- ECS (Docker containers running Redis)
- Easy to scale
- `redis-cli` to capture Redis’ metrics
- `collectd` to capture system level metrics
- Aggregated metrics to be sent out to the prediction engine
Load Testing

- Huge set of supported operations
- Easy to program
- Varied workload intensities
Metrics Considered

- CPU/Memory usage from Redis
- Request Latency
- System’s CPU/Memory usage from collectd
Approach

- Based on Utilization factor
- Predicting Future Workload → Predicting the need for Autoscaling
- Using Prediction Algorithms
Prediction Algorithms

- Recurrent Neural Network
- ARIMA MODEL
RNN

- Make use of sequential information
- Output is dependant on previous computations
- RNNs are used in a lot of areas:
  - Speech Recognition
  - Machine Translation
I have a pen. I have an apple?
ARIMA Model

- Stands for Autoregressive integrated moving average
- Model is fitted to time series data either to better understand the data or to predict future points in the series (forecasting).
- Used in short-term forecasting that requires 40 historic data points
- It uses autocorrelation to correlate data points which are at specific time interval apart
ARIMA Model

- **AR**: Autoregression. A model that uses the dependent relationship between an observation and some number of lagged observations.

- **I**: Integrated. The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.

- **MA**: Moving Average. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.
Algorithm Skeleton

Utilization_Factor=(Load Present+Predicted Load)

If Utilization_Factor > 0.70
    Autoscale!!

If Utilization_Factor < 0.30
    Scale down!!
Benefits of Auto Scaling

- Better fault tolerance
- Better availability
- Better cost management
- Optimization of resource needs
Work Distribution

● AKSHAY - Data Cleaning, Data Selection and Training the ARIMA Model for prediction. Experimenting by varying lag observations, raw observations and moving average component to evaluate the performance of the model.

● SHWETHA - Data Pre-Processing and Training of Recurrent Neural Network for prediction of incoming load. Experimenting with different hidden layers and compare analysis with ARIMA model.

● RAHUL - Setting up the Redis cluster and will be working on load testing to capture metrics. Also, will manage the addition and removal of nodes as per the prediction.
QUESTIONS?
References

References