Containerization and Resource Prediction using Time Series Analysis

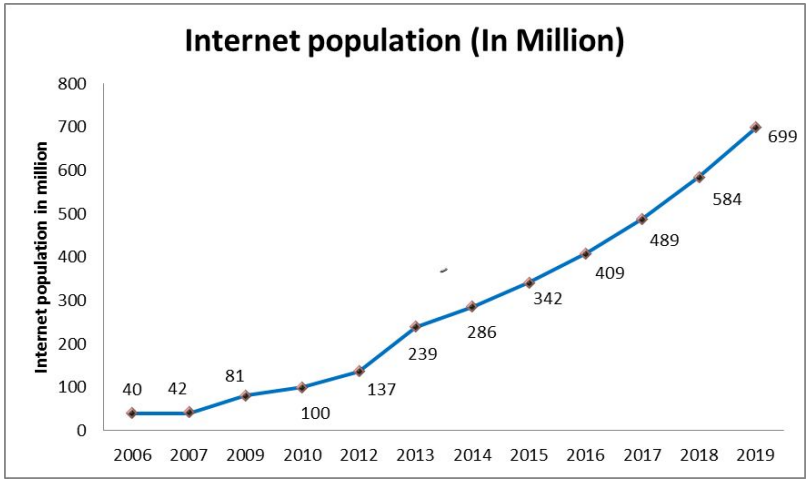
Anshuman Singh
15JE000969
Int. M.Tech. Mathematics & Computing
Supervisor: Prof. S. Gupta

Why Containerization?

With exponential growth of clients, servers moving to cloud.

What are clouds?
=> Collection of clusters working together as one entity of resource.

How?
=> Cluster Managers



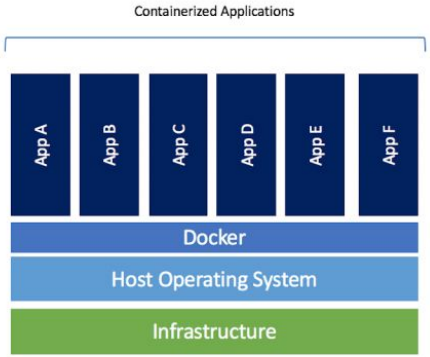
Why Containerization?

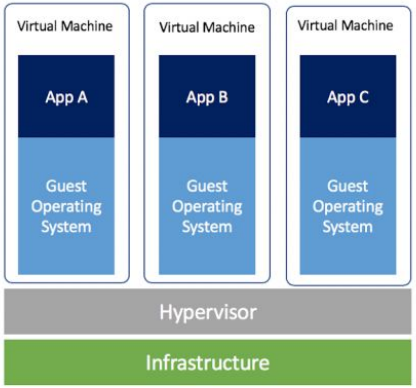
Need for virtualization?

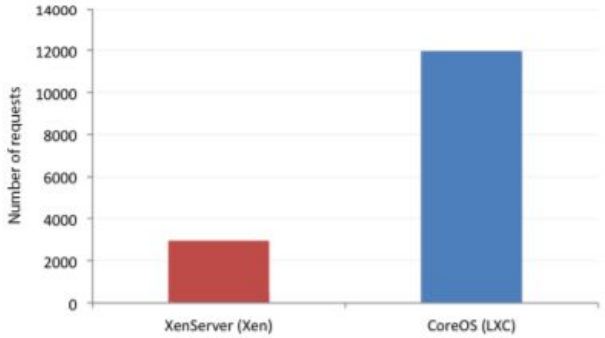
Virtualization vs Containerization?

Performance Comparison?

Benefits of containers over virtualization.







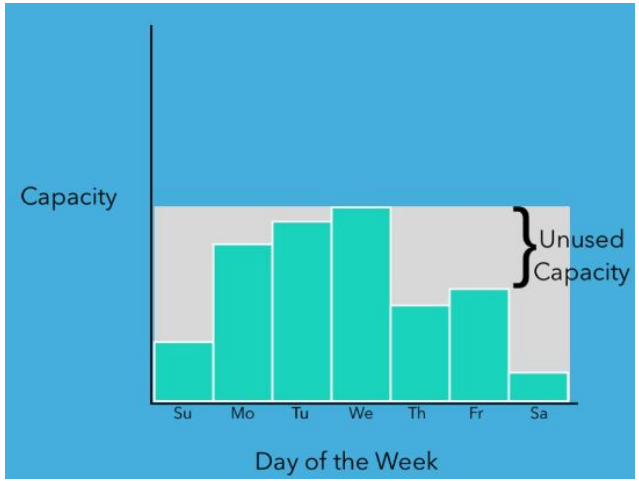
A Used Case (Need for Autoscaling)

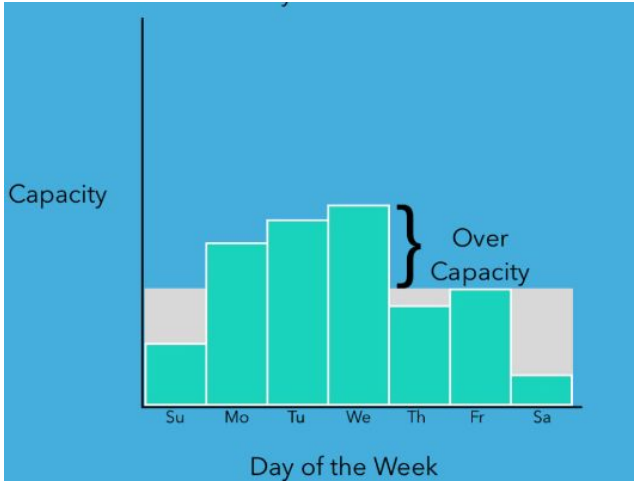
EFFECTIVE RESOURCE UTILIZATION(ERU)
VS
QUALITY OF SERVICE(QOS)

Suppose Netflix release a new TV show.

Goal: Stream the TV show efficiently resource wise.

A Used Case (Need for Autoscaling)



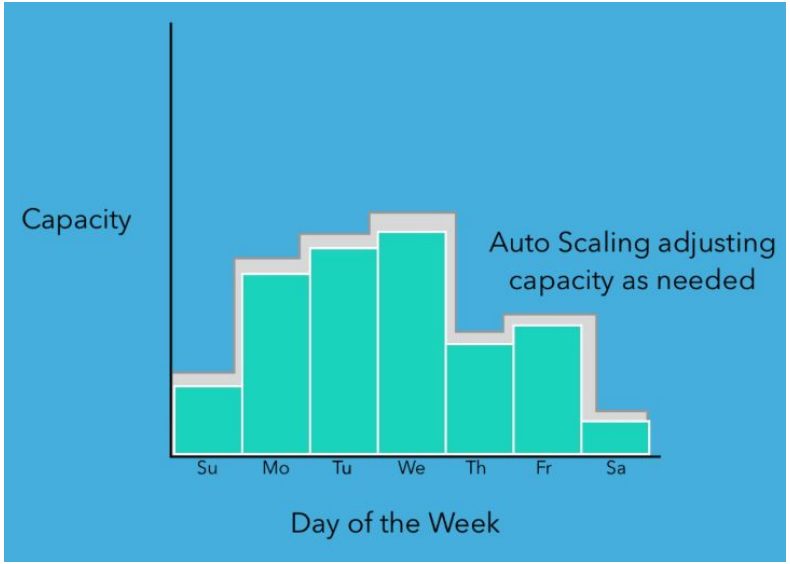


QoS

ERU

Balance?

Balance of ERU and QoS



How do we reach here? Kubernetes Autoscaling.

Current Autoscaling in Kubernetes

What is Kubernetes?

Current Autoscaling Algo →

Algorithm KHPA algorithm. It returns the number of Pods to be deployed

Input: U_{target}, *ActivePods*
// Target utilization and the set of active Pods
Output: P // The target number of Pods to deploy

```
1: while true do
2:     for all i ∈ ActivePods do
3:         U_i = getRelativeCPUUtilization(i);
4:         U = U ∪ {U_i}
5:     end for
6:     P =ceil( sum( U ) / U_target );
7:     wait(τ) // wait τ seconds, the control loop period
8: end while
```

Problem with current algo

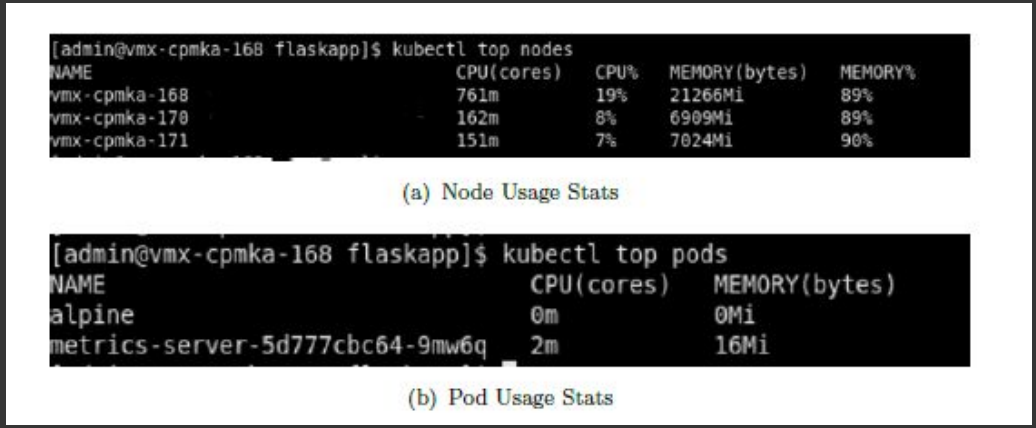
Phase	Time Taken	Description of the process
1	t1	Trigger HPA and calculate total number of replicas to be created and notify the Replication Controller
2	t2	The Controller received results and decide if up scaling or down scaling is required
3	t3	The scheduler detects creation or deletion of new pods and finds appropriate node to run it to.
4	t4	Kubelet starts the new resource downloads the images and initialize new pods into the node

$$T(total)=\sum ti$$

If time taken is large, we get request queue.

Predictive Resource Autoscaling

Dataset Generation



```
[admin@vmx-cpmka-168 flaskapp]$ kubectl top nodes
NAME                CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
vmx-cpmka-168       761m         19%    21266Mi         89%
vmx-cpmka-170       162m         8%     6909Mi          89%
vmx-cpmka-171       151m         7%     7024Mi          90%
```

(a) Node Usage Stats

```
[admin@vmx-cpmka-168 flaskapp]$ kubectl top pods
NAME                              CPU(cores)   MEMORY(bytes)
alpine                            0m           0Mi
metrics-server-5d777cbc64-9mw6q   2m           16Mi
```

(b) Pod Usage Stats

Kubernetes Metrics

Predictive Resource Autoscaling

Why Time Series?

Web Traffic Dataset of any kind has:

1. Trend
2. Periodicity
3. Crests and Troughs at some point in day.
4. Outliers

Let's apply methods to one such dataset.

	ID	Datetime	Count
0	0	25-08-2012 00:00	8
1	1	25-08-2012 01:00	2
2	2	25-08-2012 02:00	6
3	3	25-08-2012 03:00	2
4	4	25-08-2012 04:00	2

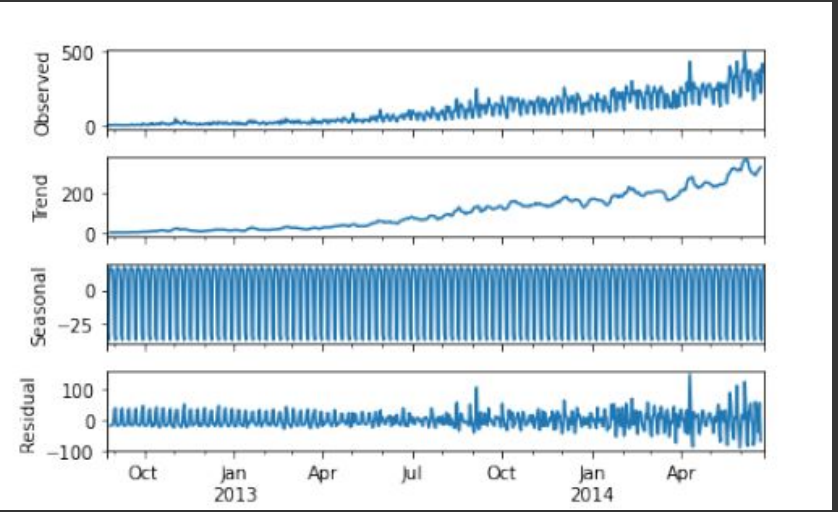
Our Dataset

Components of our series

Trend

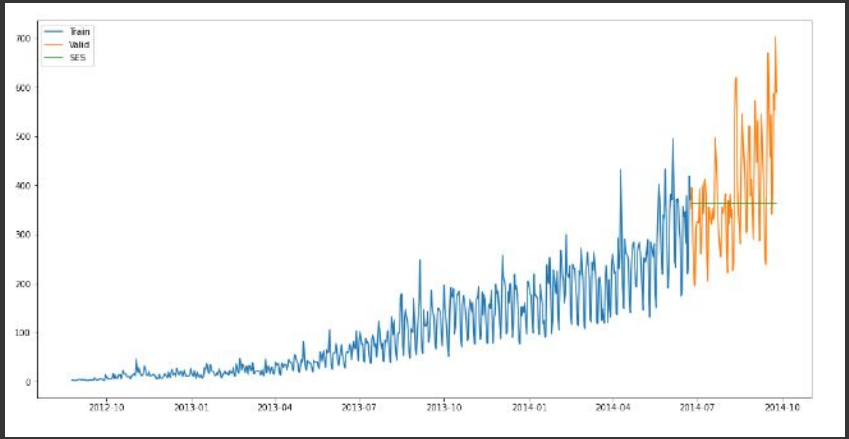
Seasonality

Outliers



Methods

1. **Moving Average Smoothing and Exponential Smoothing:**
   RMSE: 130.44 and 113 respectively.
   (Good for stagnant web servers, like wikipedia)



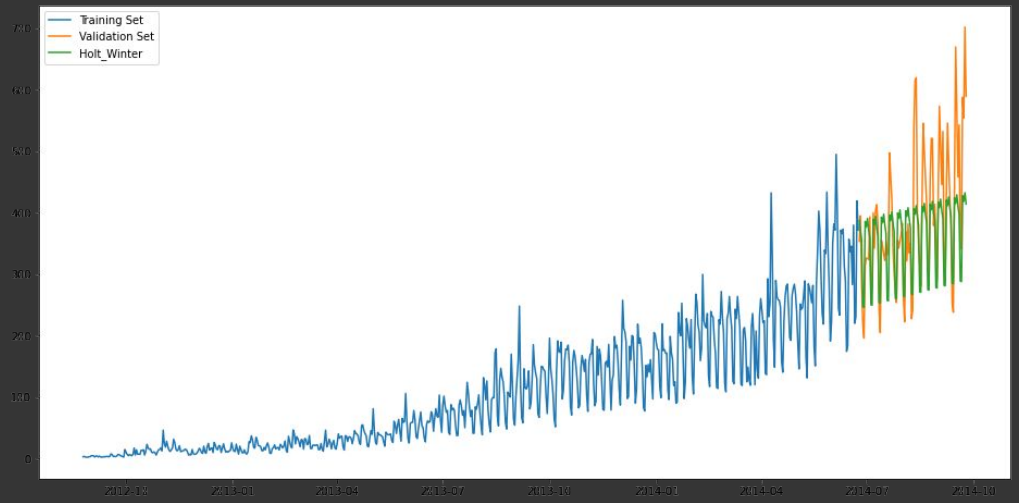
Exponential Smoothing:Not fit for our dataset

Methods

2. Holt's Models

Captured trend and seasonality.
Hard to implement.

RMSE: 100.20



Holt's Winter Model

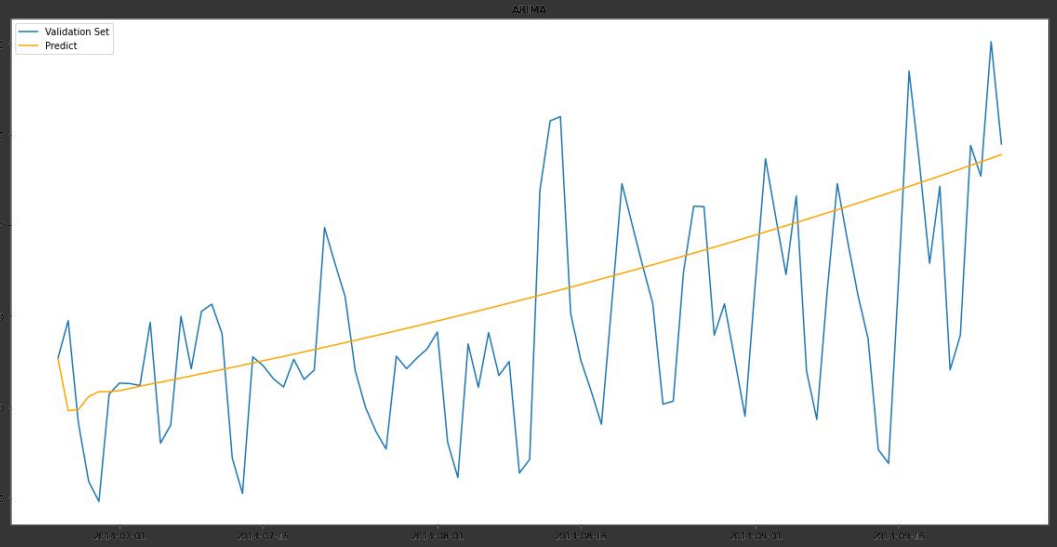
Methods

3. ARIMA

First we made the time series stationary!
RMSE: 44

Hard to implement.

Outliers not cared about.



ARIMA model on test set

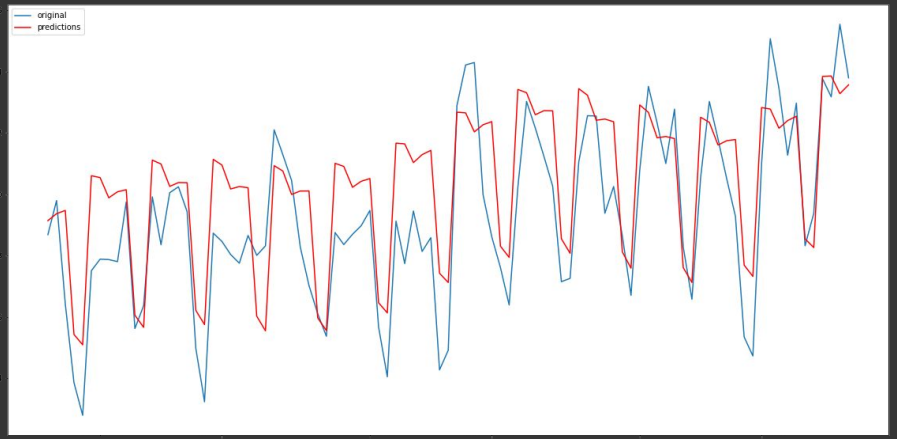
Methods

4. Facebook Prophet

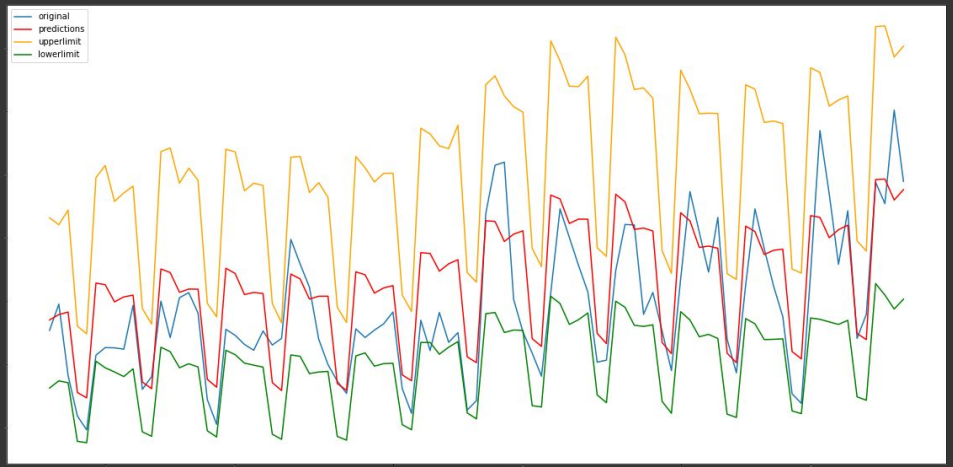
The easiest model to fit dataset of our type.
RMSE: 74

$$y = g + s + h + e$$

g-> Trend
s-> Periodic
h-> Outliers
e-> Idiosyncratic changes



Facebook Prophet’s fit



Facebook Prophet’s upper and lower(ERU vs QoS)

Facebook Prophet Model is easy fit.

Where to use?

1. Autoscaling Resources
2. Cloud Instance Allocations.
3. Cost Optimization

Future Scope

1. Dynamic Model (Ongoing Project)
2. Use RNNs for prediction.
3. Kubernetes code application.

References

1. Scheepers, Mathijs Jeroen. “Virtualization and Containerization of Application Infrastructure : A Comparison.” (2014).
2. Anqi Zhao, Qiang Huang, Yiting Huang, Lin Zou, Zhengxi Chen, and Jiang-hang Song. Research on resource prediction model based on kubernetes container auto-scaling technology. IOP Conference Series: Materials Science and Engineering, 569:052092, aug 2019.
3. Priyanshu Srivastava and Rizwan Khan. A review paper on cloud computing International Journal of Advanced Research in Computer Science and Software Engineering, 8:17, 06 2018.
4. Sean Taylor and Benjamin Letham. Forecasting at scale. The American Statistician, 72, 09 2017.
5. P. M. Swamidass, editor. Forecasting models, Holt's HOLT'S FORECASTING MODEL, pages 274{274. Springer US, Boston, MA, 2000.
6. Md. Habibur Rahman, Umma Salma, Md Hossain, and Md Tareq Ferdous Khan. Revenue forecasting using holt{winters exponential smoothing. Research Reviews: Journal of Statistics, 2016.
7. Andrew Tanenbaum and Maarten van Steen. Chapter 1 of Distributed Systems Principles and Paradigms. 03 2016.
8. Fahd Al-Haidari, Mohammed Sqalli, and Khaled Salah. Impact of cpu utilization thresholds and scaling size on autoscaling cloud resources.

Thank You