Performance Anomaly Detection & Forecasting Model (PADFM) for eRetailer Web application

Ramya Ramalinga Moorthy
EliteSouls Consulting Services LLP, Bangalore, India.
ramya.moorthy@elitesouls.in

With high performance becoming a mandate, the need for sophisticated performance management is realized by every e-business. Robust performance anomaly detection and forecasting solution is in demand to detect anomalies in production environment and to provide forecasts on server resource demand to support server sizing. This paper deals with the implementation of PADFM for an online retailer using statistical modeling & machine learning techniques that has yielded multi-fold benefit to the business.
Keywords: Performance Anomaly detection, Forecasting, K-NN, K-Means, Exponential smoothing, ARIMA.

1. Introduction

With the revolution brought by digital transformation, performance management for digital products is no more a luxury. If Amazon, experiences 1% decrease in sales for every additional 100 ms delay in response time, we can imagine the severity of performance impact on every high traffic e-business.

Though Application Performance Management (APM) tools are widely used and has brought down the performance problem diagnosis time to a great extend, these tools don't actually help in detecting the anomalies / outliers in the production environment at this point of time. As deploying robust performance management solutions became one of the compliance factors for digital products, the demand for quick & effective performance anomaly detection & forecasting solution is increasing steadily. The secret sauce for building such solution is the right choice of statistical modelling & machine learning techniques as they provide a strong basis for detecting system performance anomalies and for making forecasts.

With several techniques available in the domain of statistics & machine learning, choosing the right technique(s) that has the capability to detect maximum possible anomalies meeting high accuracy requirement of business is a great challenge. The choice of the technique(s) purely depends on the type of data and its characteristics.

A performance anomaly is an abnormal application behavior that could be due to some unrelated event or an unexpected application behavior that adversely affects the application performance. The performance data used for detecting anomalies & for making forecasts come from the measurements of two different classes of performance metrics namely application metrics (Example: server hits, # users, application response time, etc) that provides the current state / health of an application and system metrics (Example: CPU / memory / disk utilization, queue length, etc) that provides the current state of the underlying system.

Providing forecasts on the server resource requirements supports in application capacity planning & infrastructure sizing. Usage of filtered server resource utilization statistical data, free from anomalies & outliers, can result in providing accurate forecasts that can be used for server sizing analysis.

This paper deals with our solution, Performance Anomaly Detection and Forecasting Model (PADFM v1.0) developed for addressing the requirements of e-retailer business application. The paper is organized in eight sections. Section 2 provides the overview of the e-retailer problem space. Section 3 provides overview of the PADFM model & its objectives. Section 4 details various performance anomaly detection techniques employed by our model through illustrations. Section 5 details various performance forecasting techniques employed by our model through illustrations. Section 6 provides model recommendations & business benefits. Section 7 provides the limitations & future plans. Section 8 concludes the paper along with the references.

2. E-Retailer Context / Problem Space Overview

A US based retailer, who operates its women apparels business both in brick-n-mortar shop & online during last 7 years, plans to expand its online business portfolio in subsequent years. The business is interested in historical data investigation to gain confidence on what kind of analysis can be done to improve their current performance management practices. Also, the business is interested to know whether any recommendations on server resource demands can be made for justifying the need for server capacity upgrade for the year 2016.

The online ecommerce application developed using J2EE technologies is hosted on JBoss & Oracle server at their private data center is configured with IBM Tivoli to monitor the production environment health metrics at 5 minutes interval from Jan 2010 till Dec 2015 (though not much analysis is currently done on the collected data). Server traffic monitoring logs are available from Jan 2013 till Dec 2015 (though not analyzed).

Business demanded a Proof-Of-Concept (POC) model that can use the historical data for providing recommendations to improve current performance & capacity management practices.

3. PADFM Solution Overview

Our solution, Performance Anomaly Detection & Forecasting Model (PADFM v1.0) designed for POC, helps in identifying the performance anomalies from the available historical data & for forecasting the hardware demand, particularly CPU requirements of web, application & database server for next one year.

Our model is implemented on R statistical platform and uses various statistical & machine learning techniques. The choice of the techniques is based on our study of data characteristics and evaluation of accuracy of various techniques. Our model provides two key solutions - Performance anomaly detection & forecasting in offline mode.

For anomaly detection, our solution uses below algorithms.

Twitter's Anomaly detection algorithm & Breakout detection algorithm is used for detecting anomalies in the user traffic trend.

Twitter's Breakout detection is used for identifying breakouts & shift in distribution trends of server resource utilization data.

K-means machine learning algorithm is used to cluster the server monitoring data of web, application & database servers based on similarity, to group suspicious data points on the outer most clusters to quickly detect anomalies.

K-NN machine learning algorithm is used to store representative server utilization data points samples to segregate new data points into well defined groups based on majority vote of its k neighbours to quickly detect anomalies.

Pearson correlation analysis is used to correlate all the metrics considered in the model, at every, point anomalies detected by the model to help in performing first level of root cause analysis.

For forecasting, our solution uses three time series analysis techniques, moving average, exponential smoothing (single, double or triple) and ARIMA. The historical server utilization data of web, application & database server are fed to these models to forecast the demand for next year. The best forecasts are recommended based on evaluation of various forecast accuracy measures.

4. Performance Anomaly Detection Solution

The performance anomaly detection techniques are grouped by the type of underlying model they assume for nominal data and the discrimination function used to identify data that doesn't agree with the model.

Our performance anomaly detection solution is designed to offer below two type of analysis:

1) Visualize the performance counters over time, classifies them to provide easy view of anomalies or outliers and reports any deviations from the set thresholds.

2) Support in root-cause analysis by correlating several performance counters & report counters with high correlation coefficients.

The techniques used by our model are discussed below:

4.1 Anomaly Detection Model using Twitter's AD algorithm

Twitter's anomaly detection algorithm is based on S-H-ESD (**S**easonal **H**ybrid **E**xtreme **S**tudentized **D**eviates) algorithm, which is an extension of a generalized ESD method that helps to detect both global & local anomalies better than generalized ESD method.

The web server access log that has the server hit details for last 3 years (Jan 2013 till Dec 2015) is used for this analysis. By using this anomaly detection

algorithm, both global & local anomalies are detected as shown in the figure1.

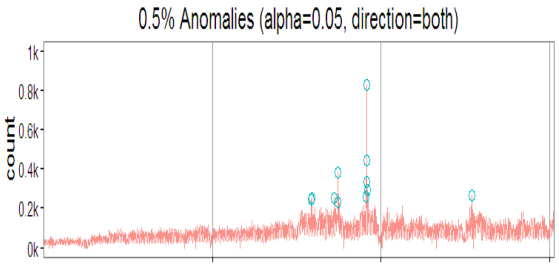

Figure 1: Anomalies detected using Twitter's AD

Drill down analysis for a specific investigation period reported large number of anomalies where upon verification, it was found that 82% of anomalies detected during the investigation period seem to correlate with the days when performance tests were conducted on production environment for a while, due to infrastructure issues on the test environment.

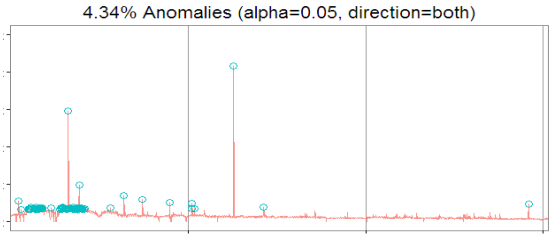

Figure 2: Drill down analysis using Twitter's AD

The anomalies detected along with its percentage deviation are reported by the model as in figure 3.

> anomaly_table

	date	timestan	anoms	expected_value	perc_diff
1	2/3/2014	8:32:00	12576	4237	-197
2	2/13/2014	16:50:00	2190	3498	-106
3	3/13/2014	2:22:00	7415	6794	-9
4	3/15/2014	9:29:00	7561	7666	1
5	3/16/2014	8:52:00	7953	7845	-1
6	3/17/2014	17:30:00	8079	7840	-3
7	3/18/2014	14:21:00	7608	7889	4
8	3/19/2014	6:27:00	7649	7713	1
9	3/20/2014	5:30:00	7634	7343	-4
10	3/22/2014	11:42:00	8015	7803	-3

Figure 3: Anomalies detected using Twitter's AD

This algorithm is chosen as it is very popular in detecting both global & local anomalies. Our solution is able to rightly detect, one point sudden increases, unusual noise, sudden growth on seasonal pattern, etc on the server hits trend. But there are two limitations, where the algorithm is not able to detect 1) linearly increasing growth anomaly & 2) negative anomaly. These two issues are represented in the figure 4.

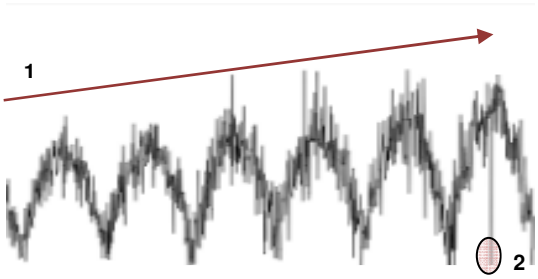

Figure 4: Anomalies not detected using Twitter's AD

4.2 Anomaly Detection in data distribution trend using Twitter's BD algorithm

Twitter's Breakout Detection algorithm is based on EDM (E-Divisive Median) which is comparatively better in detecting breakouts & identifying changes in the distribution trend quickly than E-Divisive, Moving Average & Moving Median algorithms on our input data.

Our model is able to detect two major types of anomalies, sudden increase / decrease and gradual increase / decrease in the time series trend of the resource utilization metrics.

A sudden increase in the DB server CPU utilization trend noticed on a specific day (28th Nov 2014) where the CPU utilization increased from average of 40% to 60% detected by the algorithm is shown in figure 5.

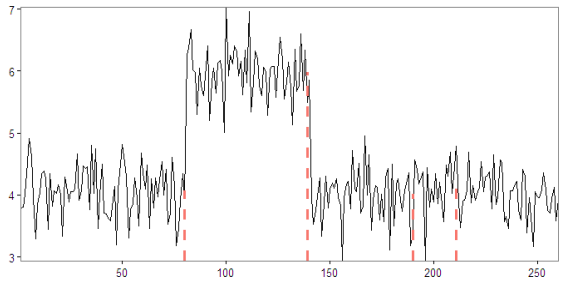

Figure 5: Sudden increase detected using Twitter's BD

A slow increase in web server disk utilization trend noticed on a specific day (14th Sep 2015) where disk utilization increased from average of 10% to 40%, detected by the algorithm is shown in figure 6.

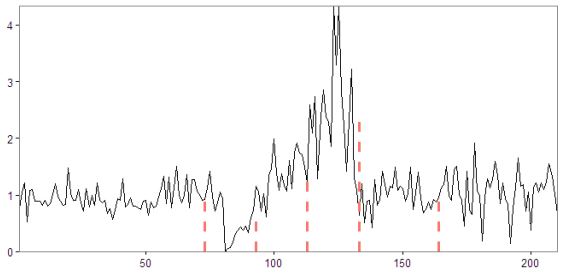

Figure 6: Slow increase detected using Twitter's BD

Though above types of breakouts are easily detected & very useful, sometimes, the algorithm doesn't detect changes in distribution shift from Poisson to Uniform.

4.3 Anomaly Detection Model using K-Nearest Neighbours Algorithm

K-NN algorithm is a simple instance based supervised learning algorithm that stores all the representative data and classifies new data input by a majority vote of its k neighbours. This algorithm segregates unlabeled data points into well defined groups. Choosing the number of nearest neighbours i.e. determining the value of k plays a significant role in determining the efficacy of the model.

The data fed to our model has % CPU & % Disk utilization values with type labels (sample shown in figure 7), collected from application server during peak hour statistics observed during 2015. The % CPU utilization threshold value (configurable) used for classifying the data as invalid (H-CPU) is about 70% & % Disk utilization threshold value (configurable) to consider the data as invalid (H-DISK) is about 20%

Data ID	% CPU	% Disk	Type
16	56	11	Valid
17	72	9	H-CPU
18	57	23	H-DISK
19	73	22	H-CPUnDISK

Figure 7: Input Data Sample used for K-NN analysis

The input data points are labeled using four classes, VALID (where both % CPU & % Disk utilization values are less than set thresholds), H-CPU (where % CPU utilization value is greater than set threshold), H-DISK (where % Disk utilization value is greater than set threshold) and H-CPUnDISK (where both % CPU & % Disk utilization values are greater than set thresholds).

Figure 8: Input data classification used for K-NN analysis

Out of 100 input data points fed to the model, 65% of the provided data is configured to be used for training purpose & remaining 35% data is used for testing purpose. The pictorial view of input data classification is shown in figure 8.

Upon applying this algorithm, the prediction accuracy provided by our model for the test data values is shown in figure 9.

Total Observations in Table: 35

SampleDataforT.testLabels	SampleDataforT_pred H-CPU	H-CPUnDISK	H-DISK	Valid	Row Total
H-CPU	5	0	0	0	5
	25.714	0.857	1.000	2.429	
	1.000	0.000	0.000	0.000	0.143
	1.000	0.000	0.000	0.000	
	0.143	0.000	0.000	0.000	
H-CPUnDISK	0	6	0	0	6
	0.857	24.029	1.200	2.914	
	0.000	1.000	0.000	0.000	0.171
	0.000	1.000	0.000	0.000	
	0.000	0.171	0.000	0.000	
H-DISK	0	0	7	0	7
	1.000	1.200	22.400	3.400	
	0.000	0.000	1.000	0.000	0.200
	0.000	0.000	1.000	0.000	
	0.000	0.000	0.200	0.000	
Valid	0	0	0	17	17
	2.429	2.914	3.400	9.257	
	0.000	0.000	0.000	1.000	0.486
	0.000	0.000	0.000	1.000	
	0.000	0.000	0.000	0.486	
Column Total	5	6	7	17	35
	0.143	0.171	0.200	0.486	

Figure 9: Resultant classification accuracy on test data

It is evident from the results that 100% of the predicted values on the tested data (35% of input data) are valid. Thus developed model can be used to classify the new unlabelled input data & the model is able to classify the input data samples (with 10000+ observations) into 4 provided input classes, with an average accuracy of about 92%.

Total Observations in Table: 119

	Prediction Data H-CPU	H-CPUnDISK	H-DISK	Valid	Row Total
H-CPU	23	0	0	1	24
	68.130	3.227	3.429	10.584	
	0.958	0.000	0.000	0.042	0.202
	0.958	0.000	0.000	0.016	
	0.193	0.000	0.000	0.008	
H-CPUnDISK	0	15	0	0	15
	3.025	83.579	2.143	7.815	
	0.000	1.000	0.000	0.000	0.126
	0.000	0.938	0.000	0.000	
	0.000	0.126	0.000	0.000	
H-DISK	0	1	17	2	20
	4.034	1.061	70.007	6.804	
	0.000	0.050	0.850	0.100	0.168
	0.000	0.062	1.000	0.032	
	0.000	0.008	0.143	0.017	
Valid	1	0	0	59	60
	10.183	8.067	8.571	24.615	
	0.017	0.000	0.000	0.983	0.504
	0.042	0.000	0.000	0.952	
	0.008	0.000	0.000	0.496	
Column Total	24	16	17	62	119
	0.202	0.134	0.143	0.521	

Figure 10: Resultant classification on new input data sample

A new unlabelled input data, with about 120 data points fed to the model has yielded 96% classification accuracy as shown in figure 10.

4.4 Anomaly Classification Model using K-Means Clustering algorithm

K-Means clustering is an unsupervised learning algorithm that tries to cluster data based on their similarity. The underlying idea of the algorithm is that a good cluster is the one which contains the smallest possible within-cluster variation of all observations in relation to each other. The variation is calculated

using squared Euclidean distance between the data points.

The data fed to our model has 100 unlabelled data points of % CPU & % Disk utilization values (sample shown in figure 11) collected from application server during peak hour statistics observed during 2015.

Data ID	% CPU	% Disk
13	73	32
14	74	9
15	34	10
16	78	9

Figure 11: Input data sample in K-means

Upon entering the # of clusters input value as four, the resultant clustered view of our model is shown in figure 12. Using this cluster representation, the anomalies can be automatically identified by analyzing cluster 4 (high CPU & Disk utilization cluster) data points.

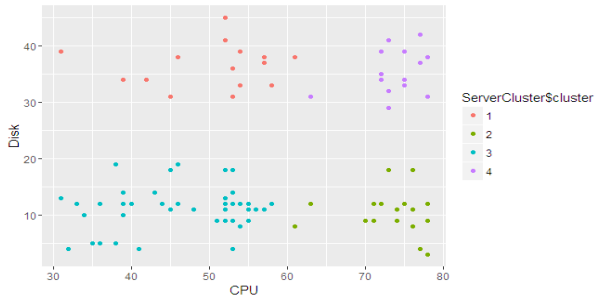

Figure 12: Resultant cluster view of input data

There are minor errors noticed in the clustering results. Upon comparing the clustered results of this data with the labeled input data (shown in figure 8 used for K-NN analysis), there are 5 deviations noticed as represented in figure 13.

```
> table(SampleData$Class,ServerCluster$cluster)

            1  2  3  4
H-CPU       0 20  1  0
H-CPUnDISK  0  0  0 14
H-DISK     15  0  0  1
Valid       0  3 50  0
```

Figure 13: Labeled data versus Cluster results comparison

The optimal number of clusters will be calculated using Elbow method. By plotting the total within sum of squares, for various selected K values, the elbow point can be identified. The graph shown in figure 14, justifies, why K value is chosen as 4 for clustering the input data sample in the discussed case. From the graph, it is clear that, the graph shows a bend & stop making big gains after this point.

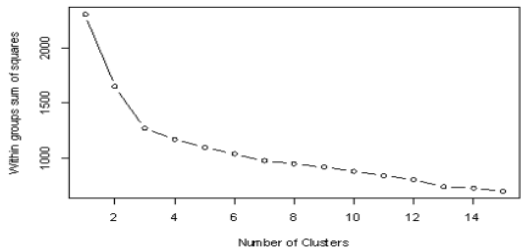

Figure 14: Cluster K value analysis

Another labeled data input fed to our model has 75 labeled data points of % CPU & % Disk utilization data from web, application & database server collected from peak traffic hours (samples format shown in figure 15). K-means clustering result for this input data is shown in figure 16.

Data ID	% CPU	% Disk	Type
1 to 25	...	...	Web Server
26 to 50	...	...	Application Server
51 to 75	...	...	DB Server

Figure 15: Input data sample with server labels

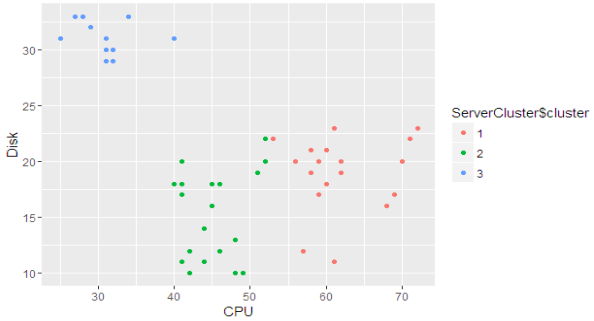

Figure 16: Clustering results of Web, App & DB server data

Comparison of resultant data clusters with the input data values showing minor erros is represented in figure 17. 2 data points of application server data class (cluster 1) is wrongly clustered into cluster 2 and incase of DB server data class (cluster 3), 3 data pointss are wrongly clustered into cluster 2.

```
table(SampleData1$Class,ServerCluster$cluster)

            1  2  3
App Server 23  2  0
DB Server   0  3 22
Web Server  0 25  0
```

Figure 17: K-Means Clustering results validation

By tuning the right k value, K-means clustering analysis can be easily used for classifying data both in online & offline modes for analyzing the production monitoring data or user traffic data for easy outlier detection.

4.5 Anomaly Root Cause analysis using Pearson Correlation algorithm

Our root-cause analysis solution is based on Pearson coefficient correlation, which assumes that the two populations have bell-shaped (normal) distributions and that the relationship between them, if any, is approximately a straight line. The correlation between 2 metrics is accepted as real & significant, if p value calculated from the correlation test is less than 0.05.

Our model takes into account 10 performance metrics (shown in figure 18), for performing correlation analysis on the server metrics. For every threshold deviation reported on CPU utilization, below correlation analysis can be performed by our model. In an instance, high CPU utilization confirmed the presence of memory leak in the application. During correlation analysis, 4 key counters, CPU, Memory (calculated from Available Mbytes counter), and Disk % utilization & processor queue length are correlated & reported as in figure 19. Additionally any counters of interest from the list can be added to do the correlation, if required.

Performance Object	Counter
Processor	% Processor Time
	% User Time
Memory	Available MBytes
	Page Faults/sec
Physical Disk	% Disk Time
	Avg disk queue length
	Avg disk sec/transfer
Network	Bytes Sent/sec
	Bytes received/sec
System	Processor Queue Length

Figure 18: Performance Metrics (#10) used by PAD model

The correlation trend analysis graph shows the previous 6 hours data trend (configurable value) from the threshold deviation point.

The Pearson correlation coefficient analysis between % CPU Utilization & % Available Memory metric is reported as -0.83 with the p-value of 2.2e-16 (<0.05).

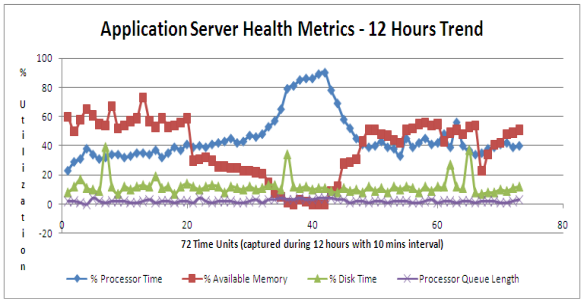

Figure 19: Server resource trend from our model showing before & after 6 hours of anomaly point resulted in exception

The scatter plot is used to visually report the correlations across all counters, which confirms existence of correlation between CPU & memory Utilization counter.

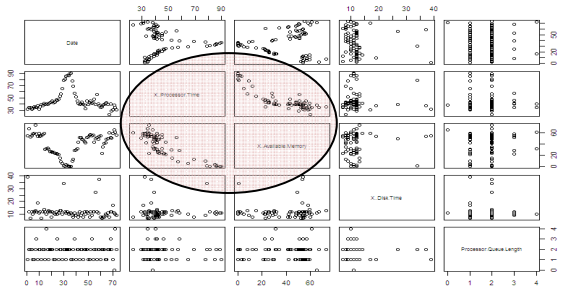

Figure 20: Scatter plot representing correlations

The correlation coefficient matrix shown in figure 21 is used to report the severity of the correlation between counters used in analysis.

```
> ServerMetricsCorrelationMatrix
                P.CPU P.Avail.Mem      P.Disk          QL
P.CPU       1.00000000 -0.83164192  0.01238987  0.58917414
P.Avail.Mem -0.83164192  1.00000000  0.03789708 -0.54151370
P.Disk       0.01238987  0.03789708  1.00000000 -0.04558604
QL           0.58917414 -0.54151370 -0.04558604  1.00000000
```

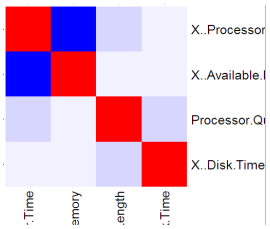

Figure 21: Correlation Coefficient Matrix & Heap map view

The heat map view shows the correlation values of all selected counters used in the analysis. The dark blue boxes in the figure, represents metrics with significant negative correlation value, as %.CPU & % Memory utilization values have inverse proportionality.

Various rules are included in the model to conclude on the root-causes though significant correlation can be seen between multiple counters. These results & graphs reported by our model help in quickly detecting the anomalies & performing root cause analysis.

5. Performance Forecasting Solution

Our forecasting model uses three univariate time series analysis techniques to forecast the future behavior of a performance metric according to its past values. Our model deploys Moving Average, Exponential Smoothing (Single or Double or Triple) and ARIMA techniques to forecast the CPU requirement of web, application & database servers and highlights the best forecasts based on forecast accuracy measures, RMSE & MAPE values.

Our forecasting model used in PADFM version 1.0 uses 95th percentile value of past 72 months (Jan 2010 till Dec 2015) to make the forecasts for next year. The % CPU utilization data reported by Tivoli at every 5 minutes interval is used as input for the forecasting model, after removal of anomalies (using the techniques discussed in section 3). The 95th percentile value of every hour is used to calculate the 95th percentile value for every day & this value is then used to calculate the 95th percentile value for every month for the last 6 years.

Though this is not the right approach, this technique is followed since the objective of the model during POC is to show how various forecasting techniques can be applied on the available statistical data to provide recommendations for server sizing. Thus, the data derived from this data aggregation analysis, approved by the business team & infrastructure team is used in forecasting models.

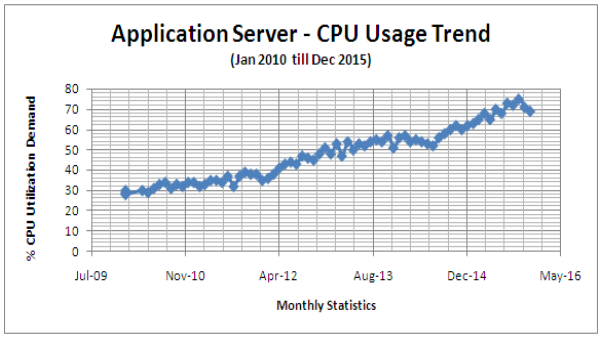

Figure 22: Time Series data input used for forecasting

Figure 22 provides the view of year on year increase in the Application Server % CPU utilization during last 6 years (Jan 2010 till Dec 2015). The forecasts made using this time series data input by applying 3 statistical modelling techniques will be discussed in this section to make forecasts for the next year, 2016.

Our forecasting model facilitates making forecasts either by using the all the data points or for the user selected timeframe period (that are considered relevant for sizing).

5.1 Moving Average (MA) Model

This method uses the average of the most recent k data values in the time series as the forecast for the next period. In this method, every time a new observation becomes available for the time series, the oldest observation in the equation is replaced and a new average is computed. For most recent time series values, a small value of k is preferred whereas for past values, a larger value of k is preferred.

Forecasted trend reported by our solution is shown in figure 23.

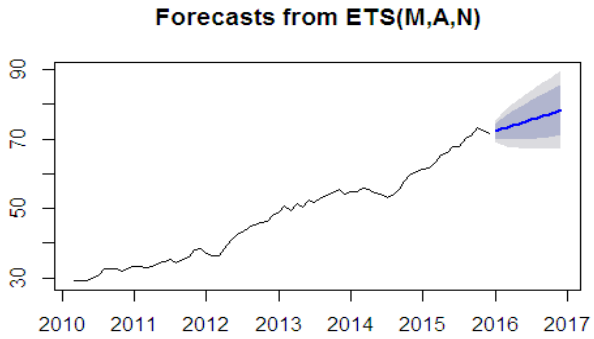

Figure 23: Forecasted Trend using MA model with k=3

The forecast accuracy measure values are calculated as RMSE is 1.03 & MAPE is 1.79.

5.2 Exponential Smoothing (ES) Model (Single, Double & Triple techniques)

This method uses a "smoothing constant" to determine how much weight to assign to the actual values. The smoothing parameter should fall into the interval between 0 and 1, so as to produce the smallest sums for the residuals.

Our solution performs data decomposition analysis to study the type of time series data before performing this analysis. During decomposition analysis, the input time series data is analyzed for the presence of 3 components - level, trend & seasonality. Accordingly, our solution applies one of the below mentioned smoothing technique to make forecasts.

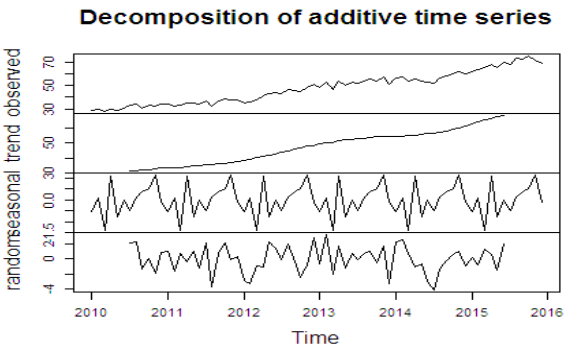

Figure 24: Decomposed view of input time series

5.2.1 Simple Exponential Smoothing

This method is applicable for time series data with only level and no trend or seasonality. The fitted trend & forecast trend reported by our model are provided in figure 25 & 26.

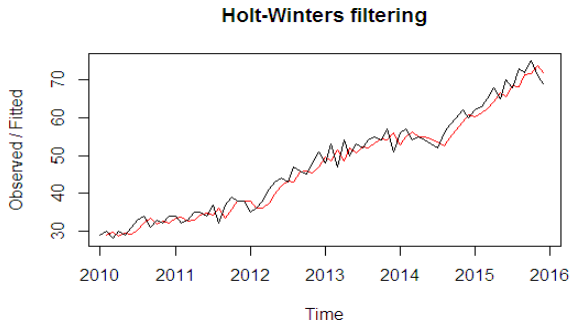

Figure 25: Observed vs Fitted Trend using Simple ES

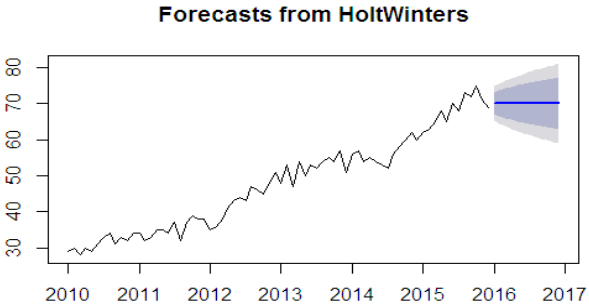

Figure 26: Forecasted Trend with alpha=0.64

The forecast accuracy measure values are calculated as RMSE is 2.54 & MAPE is 4.65.

5.2.2 Holt's Exponential Smoothing

This method is used when time series data have level and trend and no seasonality. Forecasts are made where smoothing is controlled by two parameters, alpha, for the estimate of the level at the current time point, and beta for the estimate of the slope b of the trend component.

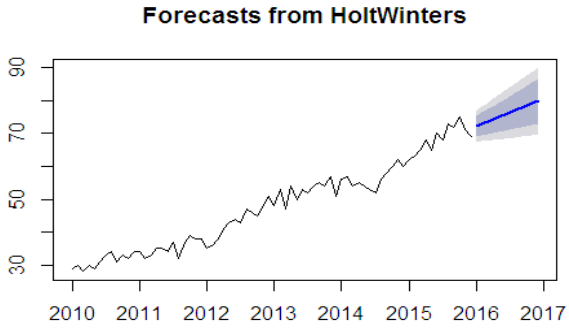

Figure 27: Forecasted trend with alpha = 0.48 & Beta = 0.03

The forecasted trend & residual trend are shown in figure 27 & 28. The residual plot shows forecast errors seem to have roughly constant variance over time.

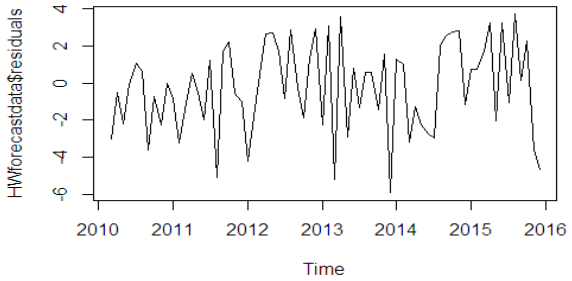

Figure 28: Residuals Trend using Holt's ES

The forecast accuracy measure values are calculated as RMSE is 2.4 & MAPE is 4.4.

5.2.3 Holt-Winter's Exponential Smoothing

This method is used when time series data have level, trend and seasonality. This ES technique is majorly used, as server resource utilization time series data had level, trend & seasonality components. Forecasts are made where smoothing is controlled by three parameters: alpha, beta, and gamma, for the estimates of the level, slope b of the trend component, and the seasonal component, at current time point.

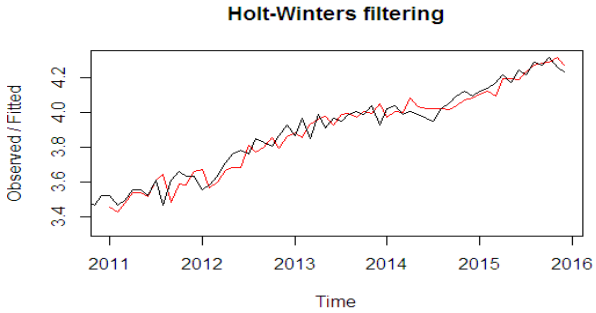

Figure 29: Observed vs Fitted Trend using Holt-Winter's ES

The fitted trend & forecast trend reported by our model are shown in figure 29 & figure 30.

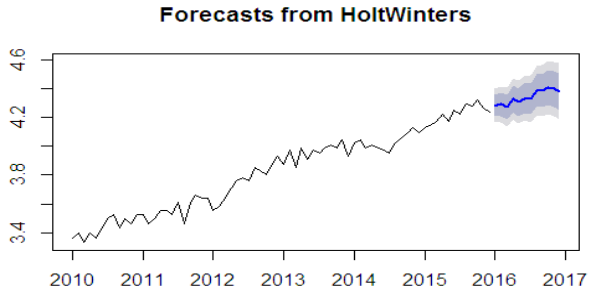

Figure 30: Forecasted Trend using Holt-Winter's ES model with alpha = 0.39, beta =0.006 & gamma =0.46

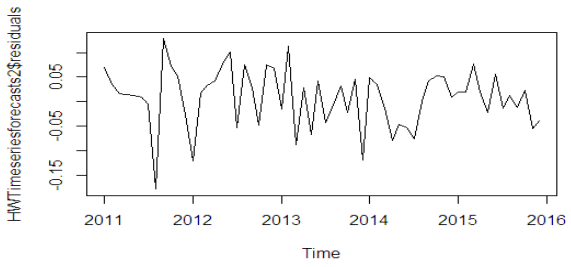

Figure 31: Residuals Trend using Holt Winter's ES model

The residual plot provided by our model shows forecast errors seem to have roughly constant variance over time. As per the Ljung-Box test, p-value is 0.55 (>0.05), so there is little evidence of non-zero autocorrelations in the in-sample forecast errors at lags 1-20 confirms the appropriate fit of time series model.

To be sure that the predictive model cannot be improved upon, our model provides a histogram representation of forecast errors. This helps to check whether the forecast errors are normally distributed with mean zero and constant variance. The forecast errors histogram plot with overlaid normal curve shows that the distribution of forecast errors is perfectly centered on zero and shows a normal curve.

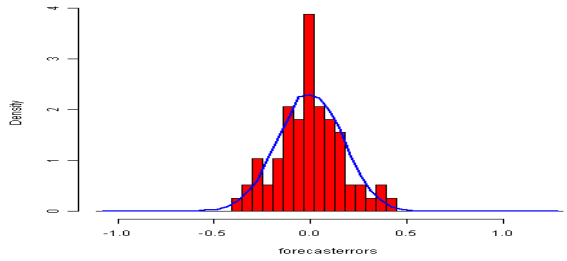

Figure 32: Forecasted Errors Histogram

The forecast accuracy measure values are calculated as RMSE is 0.06 & MAPE is 1.29.

5.3 ARIMA Model

AutoRegressive Integrated Moving Average (ARIMA) model is the most popular approach to forecast both stationary & non-stationary time series data. The model summarized as ARIMA (p,d,q), comprises of three parameters: autoregressive parameter (p), the number of differencing passes (d), and moving average parameter (q). The method operates in three phases namely Transformation, Parameter Estimation and Fitting & Diagnostic Checking.

Our model uses transformation functions to convert the input data to stationary data and then calculates the parameter values to fit the best ARIMA model to make forecasts.

The non-stationary input data shown in figure 33, is differenced until it appear to be stationary in mean and variance, i.e., the level & variance of the series should stay roughly constant over time, as shown in figure 34.

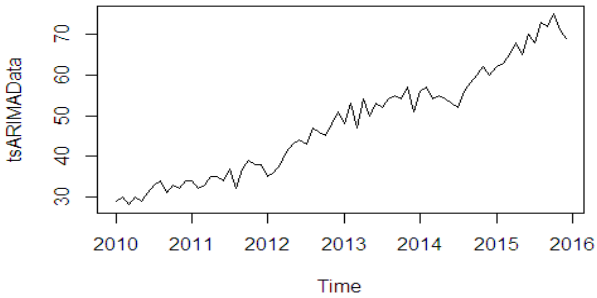

Figure 33: Non-stationary input data

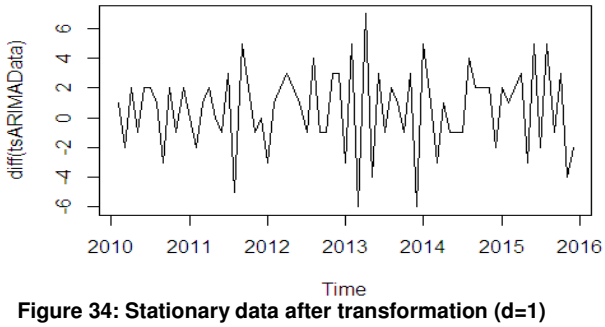

Figure 34: Stationary data after transformation (d=1)

Unit root test is performed to test the stationary of the time series data. The Correlogram (ACF) and Partial Correlogram Analysis (PACF) analysis carried out to calculate the value of ARIMA model parameters p & q is shown in figure 35 & figure 36.

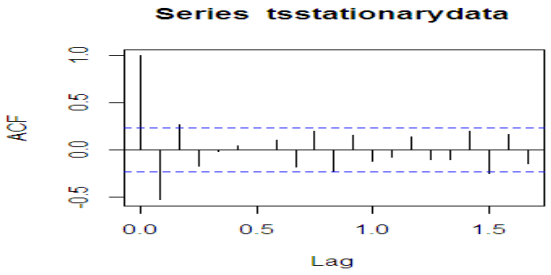

Figure 35: ACF Correlogram View

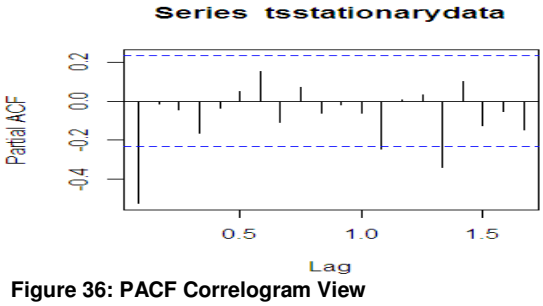

Figure 36: PACF Correlogram View

Our model suggests the best ARIMA fit with least AIC and BIC value. The forecast trend reported by our solution using the recommended ARIMA model (1,0,0) (1,0,0) is provided in figure 37.

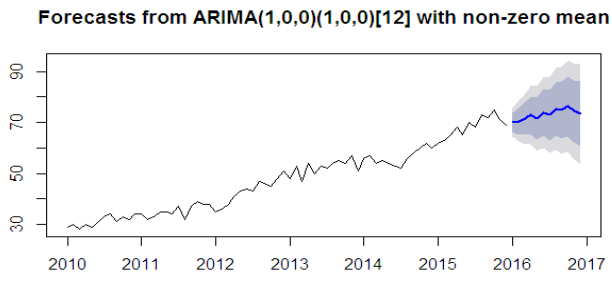

Figure 37: Forecasted Trend of ARIMA model

The Ljung-Box test & Normal Q-Q Plot are performed to validate there is no pattern in the residuals. The p-values for the Ljung-Box test are well above 0.05, indicating "non-significance". The normal Q-Q plot values are normal as they rest on a line and aren't all over the place.

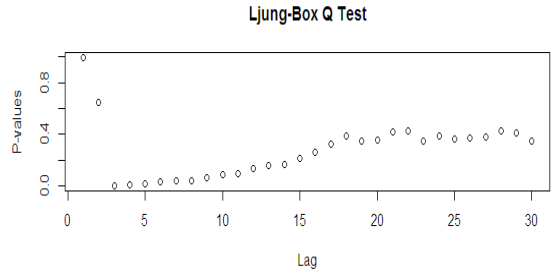

Figure 38: Ljung-Box Test Plot

The forecast accuracy measure values are calculated as RMSE is 2.89 & MAPE is 4.95.

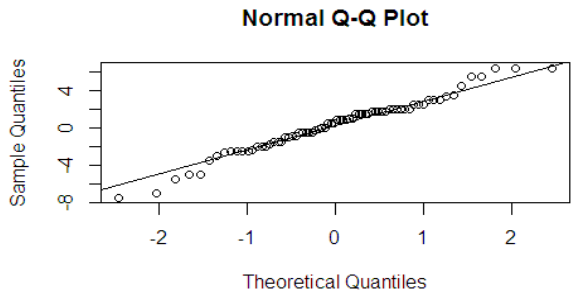

Figure 39: Normal Q-Q Plot

5.4 Forecasting Model Accuracy Analysis Summary

Our model performs the below three types of analysis to confirm the best fit.

Ljung Box test (p value >0.05) is used to check for non-significance of autocorrelations on the residuals.

Histogram plot of forecast errors is used to check whether the residuals are normally distributed with mean zero and constant variance.

Various forecast accuracy measures (ME, RMSE, MAE, MPE, MAPE, MASE & ACF1) are used to compare the forecast accuracy of employed techniques. Forecast model recommendation is based on lowest RMSE & MAPE value shown in figure 40.

Forecast Accuracy Measures Comparision Report:

Best Recommended Forecast Model : ES

Measures	ES	MA	ARIMA
ME	0.01	0.07	0.33
RMSE	0.06	1.03	2.89
MAE	0.05	0.83	2.28
MPE	0.22	0.06	0.58
MAPE	1.29	1.79	4.95
MASE	0.31	0.11	0.3
ACF1	-0.09	0.05	-0.53

Figure 40: Forecasting Model Accuracy Measures

As shown in the above recommendation, Exponential Smoothing technique seems to have better forecast accuracy with relatively low RMSE value (0.06) & MAPE value (1.29).

5.5 Forecasting Model Results

Though our forecasting model recommends the best forecasting technique, it provides the forecasted trend of all the models for referential purpose. The forecasted data trend for the application server CPU utilization for the year 2016 reported by our model is shown in figure 41.

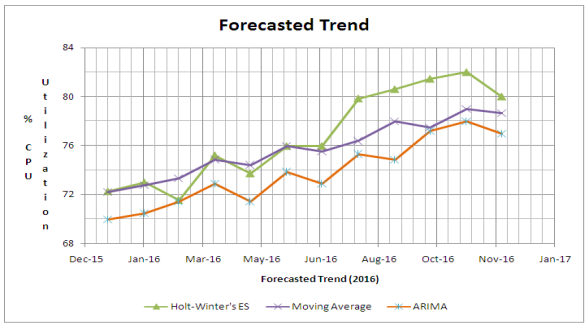

Figure 41: Forecasted Trend based on 3 techniques

As per the forecast, the application server maximum % CPU utilization will fluctuate between 80% to 85% range during 2016. As the forecast values are above the application server % CPU utilization threshold (configured as 70%), the model suggests for capacity upgrade. This justifies the need for application server capacity upgrade for the next year, 2016. This forecast assumes, no sudden increase in traffic trend is expected, due to new promotional deals or

marketing campaign activities, etc. As these factors are not considered in the model, extra capacity room needs to be buffered to accommodate the additional requirement for the forecasted year.

Server Capacity Upgrade Recommendation Report:

Configured Servers	Server Upgrade Required?
Web Server	NO
Application Server	YES
DB Server	NO

Figure 42: Server Capacity Upgrade Recommendation Report

A similar analysis performed for web server & database server confirmed, there is no need for server capacity upgrade for the year 2016. As per the forecasts reported by the model, the web server maximum % CPU utilization will fluctuate between 60% to 65% range (threshold: 80%) and the database server maximum % CPU utilization will fluctuate between 45% to 50% range (threshold: 75%).

6. PADFM Results & Business Benefits

The anomalies detected in the user traffic trend & server resource utilization trend using various techniques requires manual confirmation to include or exclude the reported anomalies for forecasting analysis. For a reported point anomaly, 12 various performance metrics (# server hits & # users details from user traffic data and 10 server resource monitoring metrics shown in figure 18) can be graphically viewed & Pearson correlation analysis can be performed to identify the root cause for the anomalous behavior.

Based on the root cause analysis results, server configuration tuning or application code tuning activities can be suggested. This manual analysis of point anomalies can result in any of the 2 actions namely; anomaly data point can be included for sizing analysis, considering it is a false positive detection or can be removed considering it as an outlier / unrelated event for the sizing analysis.

This refinement made on the time series data after filtering of possible anomalies and outliers is then used by the forecasting models to make forecasts for the next year. The best forecasted model results are used as an indication for justifying the need for server capacity upgrade for the forecasted period.

The business benefits achieved through our model include 1) Improvement in performance SLA management 2) Quick correlation analysis reports helps business to understand the system load & utilization behavior during BAU & peak seasons due to dynamisms in business factors like deals & promotions. 3) Anomalous distribution trends reported helps in providing feedback to performance testing team for refining the test workload by considering realistic usage patterns. 4) Point anomalies indicated on server resource trends helps in bringing immediate focus on performance tuning and optimization activities based on first level of root cause analysis results. 5) Better forecast accuracy & server sizing, due to the usage of filtered (anomalies / outliers removed) input data. 6) Added intelligence on production monitoring to detect anomalies & to perform first level of root cause analysis helps in reducing the L1 and/or L2 level infrastructure support team on production environment.

7. Model Limitations & Future Plans

Our PADFM version 1.0 model has several limitations in handling large data volumes, processing speed, rich graphical representation & automatic workload versus server performance correlation analysis. Also, forecasts are purely based only on server utilization data without consideration of various business factors like promotional events/deals history, such impact to be assumed for the forecasted period and Server specifications (like CPU model, type & speed, memory, etc). The impact due to these factors business & application factors need to be correlated with server resource utilization patterns for making forecasts.

Our next version of the model, PADFM v2.0 under progress, focuses on handling the above limitations and includes additional techniques like NetFlix RAD algorithm (based on Robust Principle Component Analysis) & LOF (Local Outlier Factor) for detecting additional anomaly patterns missed by version 1.0 model.

Additional features introduced to capture workload fluctuations along with server utilizations including JVM statistics to feed the supervised machine learning techniques (K-NN algorithm) can help in quickly classifying input data for detecting anomalies related to workload fluctuations & patterns.

Forecasting model will include server sizing recommendations based on the several additional business & application factors along with industry benchmark data from TPC/SPEC.

To overcome the limitations related to data handling & speed, we have used InfluxDB, an open source database for storing real-time time series data from production servers & we have used Grafana for better visualization of time series data for representing the anomaly detection graphs.

8. Conclusion

Unidentified performance anomalies or unexpected capacity issues might lead to potential system failure leading to immediate impact on customer experience resulting in business loss. The real success of performance anomaly detection & forecasting solution lies in choosing the right combination of techniques (with fine tuning) that can complement each other and accounting the impact due to several other factors to yield better accuracy. The techniques employed by our model for retailer client has yielded great benefits for the business in detecting anomalies in production environment & for providing recommendations for server sizing.

References

[1] Ludmila Cherkasova, Kivanc Ozonat, Ningfang Mi, Julie Symons, and Evgenia Smirni. Anomaly? application change? or workload change? towards automated detection of application performance anomaly and change.

[2]Joao Paulo Magalhaes and Luis Moura Silva. Detection of performance anomalies in web-based applications.

[3] Sandip Agarwala, Fernando Alegre, Karsten Schwan, and Jegannathan Mehalingham. E2eprof: Automated end-to-end performance management for enterprise systems

[4] Manjula Peiris, James H Hill, Jorgen Thelin, Sergey Bykov, Gabriel Kliot, and Christian Konig. Pad: Performance anomaly detection in multi-server distributed systems.

[5] Tao Wang, Jun Wei, Wenbo Zhang, Hua Zhong, and Tao Huang. Workload-aware anomaly detection for web applications.

[6] Joao Paulo Magalhaes and Luis Moura Silva. Root-cause analysis of performance anomalies in web-based applications.

[7] Frank M Bereznay and Kaiser Permanente. Did something change? using statistical techniques to interpret service and resource metrics.

[8] Daniel Joseph Dean, Hiep Nguyen, and Xiaohui Gu. Ubl: unsupervised behavior learning for predicting performance anomalies in virtualized cloud systems.

[9] Terence Kelly. Detecting performance anomalies in global applications.

[10] Tian Huang, Yan Zhu, Qiannan Zhang, Yongxin Zhu, Dongyang Wang, Meikang Qiu, and Lei Liu. An lof-based adaptive anomaly detection scheme for cloud-computing.