



Exception Based  
Modeling and Forecasting

Igor Trubin, PhD

## Introduction:

### ■ Questions

- How often does the need arise for modeling and forecasting?
- Should it be done manually by ad-hoc, by project requests or automatically?
- What tools and techniques are best for that?

### ■ Answers

- The capacity management system should automatically provide a small list of resources that needs to be modeled or forecasted; then a simple spreadsheet tool can be used for forecasting.
- This method is already implemented on the author's environment with thousands of servers.

# Introduction: pre-experience

## ■ Developing a forecasting system for a Stock Brokerage company:

- sometimes the management did not need accuracy but just nice pictures and graphs to persuade the customers to buy something...

## ■ First precise forecast

- simple model: based on common benchmarks (TPM) to recalculate UNIX box CPU usage for a few possible upgrade scenarios.

## ■ “Beautiful” Trend-Forecast Chart

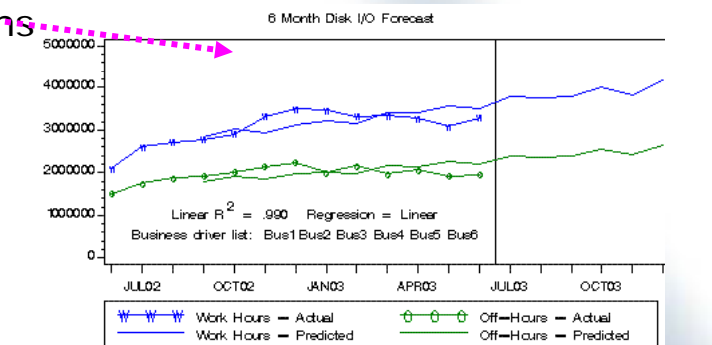
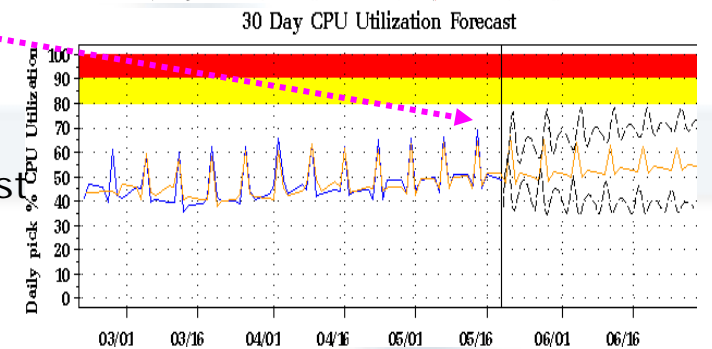
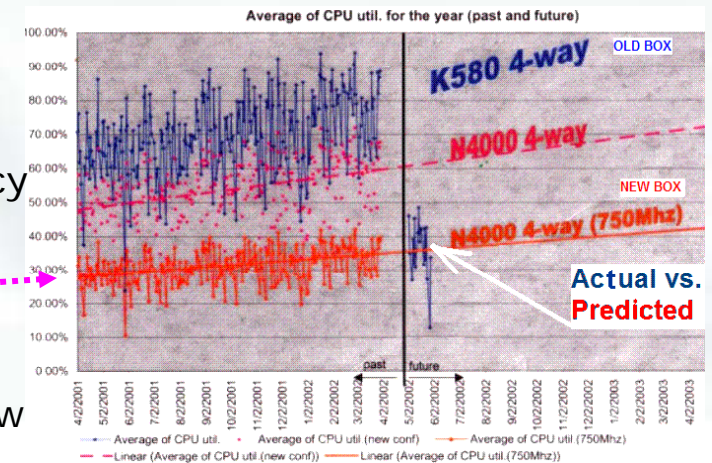
- automatically updated trend forecast charts for every server for several metrics ... in most cases were useless! **Why?** - In a good environment most of the production boxes are not trending much.

## ■ Other CMG papers:

- “..to automate the production of resource projections from business driver inputs; projections are stored in a database; automatic “actual vs. projected” comparisons was displayed...”

## ■ ALL in ALL it appeared that

- The maintenance and usage of the “bulk” forecasting with or without business driver correlations **becomes a nightmare!!!**

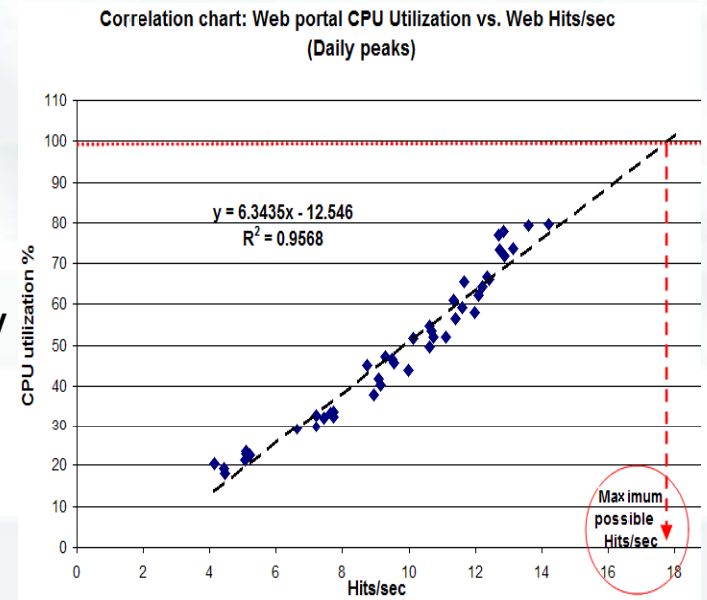


## Basic Forecasting Rules

- **RULE A:** "Right Summarization". Historical data should have the right summarization, including data for correlation (e.g. business drivers).
- **RULE B:** "Do not mix shifts". Do not mix shifts: forecasts should be done separately for working days and/or hours or off-shifts.
- **RULE C:** "Statistical models influence". The result depends on the statistical model chosen.
- **RULE D:** "Significant Events". The starting historical time point should take into account the significant events such as hardware upgrades; virtual machines, databases and application migrations; LPARs reconfigurations and so on.
- **RULE E:** "Outliers". "Bad" data points should be excluded from historical sample as "Outliers".

## Basic Forecasting Rules. **RULE A:** "Right Summarization"

- Collector data granularity: 10 sec -15 min at least.
- Should be summarized by hour, day, week or month.
- Do not forecast based on raw data – only after summarization.
- Do not forecast based on hourly or daily snap-shots type of data.
- Independent and dependent variables should be normalized by the same interval.
- Less granular the summarized data is – the better the correlation that can be found.

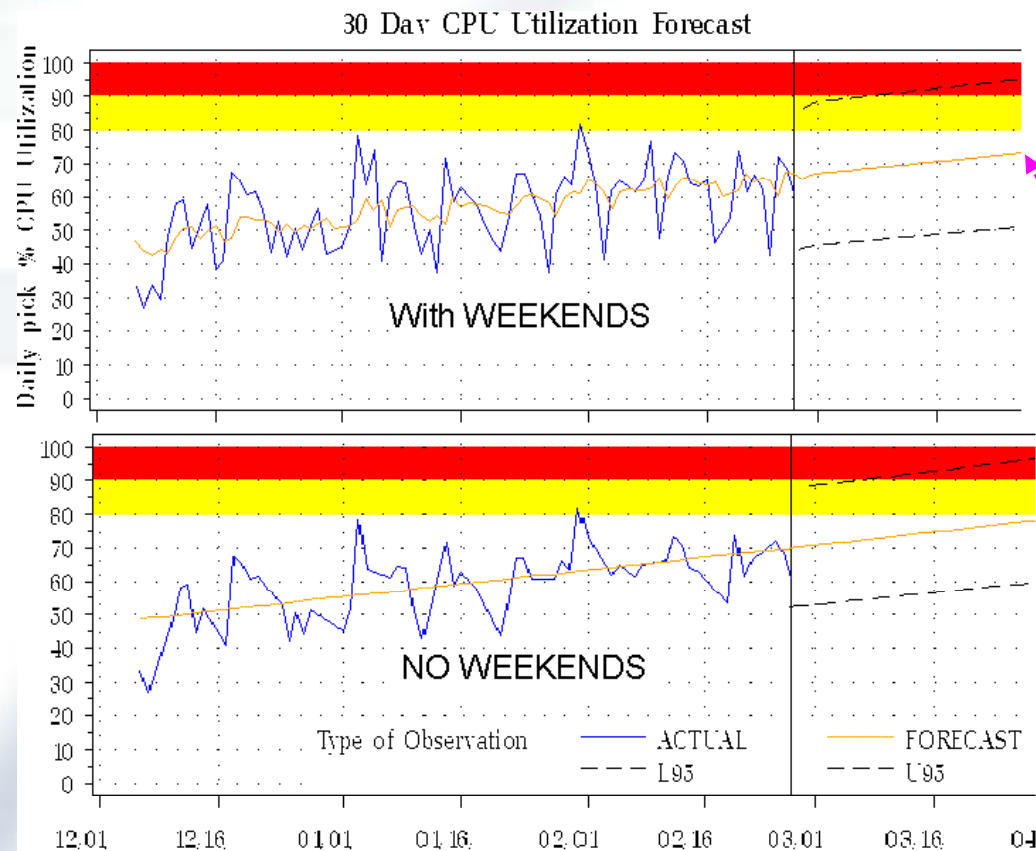


A good example of correlation analysis of CPU usage vs. Web hits  
(5 min. interval data summarized by days)

[Link To SPREADSHEET](#)

## Basic Forecasting Rules. **RULE B: "Do not mix shifts"**

- This is a pretty obvious rule; here is the example to show the difference between trend forecast made with including and excluding weekends.



The “no-weekends”  
chart forecast reaches  
the yellow zone sooner!

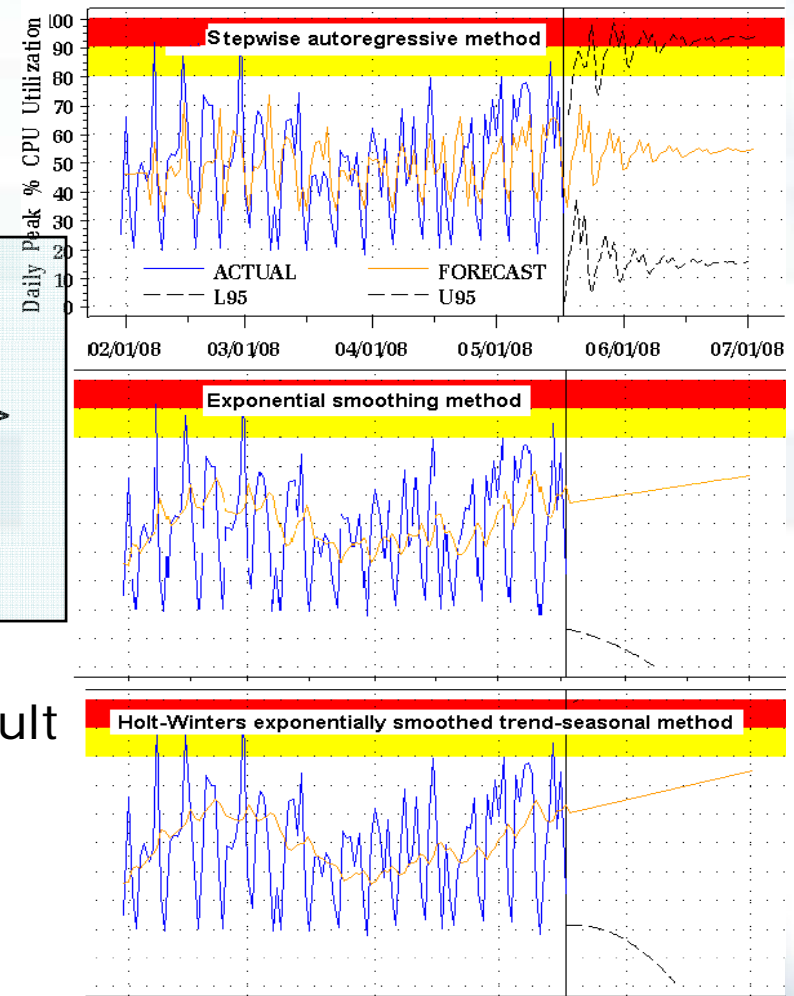
## Basic Forecasting Rules. **RULE C:** "Statistical models influence"

- Start with linear trend but play with other algorithms if they are available and really necessary.
- Using SAS language:

```
proc forecast  
  data=<actual_dataset>  
  interval=day lead=30  
  method=<STEPAR|EXPO|WINTERS>  
  trend=<1|2|3>  
  out=<actual&predicted_dataset>;  
run;
```

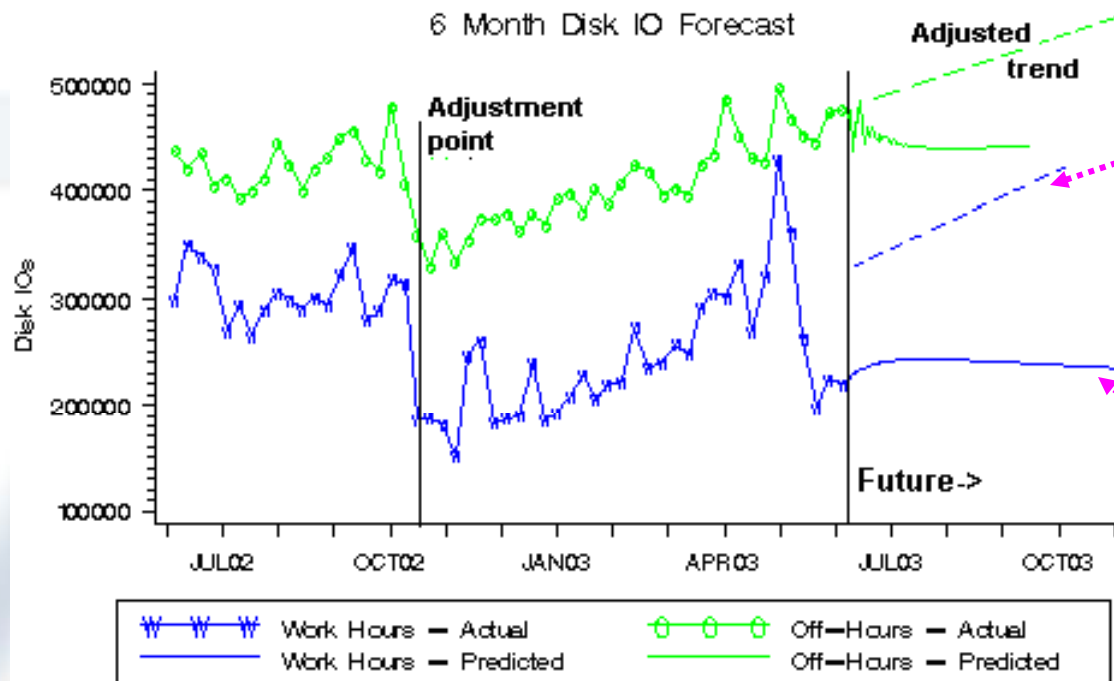
(The **stepar** and **trend=2** are default values and they mean "stepwise autoregressive" with the "linear trend model")

- Using : [SPREADSHEET](#)



## Basic Forecasting Rules. **RULE D: "Significant Events"**

- The historical data might consist of phases with different patterns (upgrades, workload shifts or consolidations, LPAR moving/reconfiguring).
- The forecasting method should be adjustable to take in consideration only the latest phase with a consistent pattern.

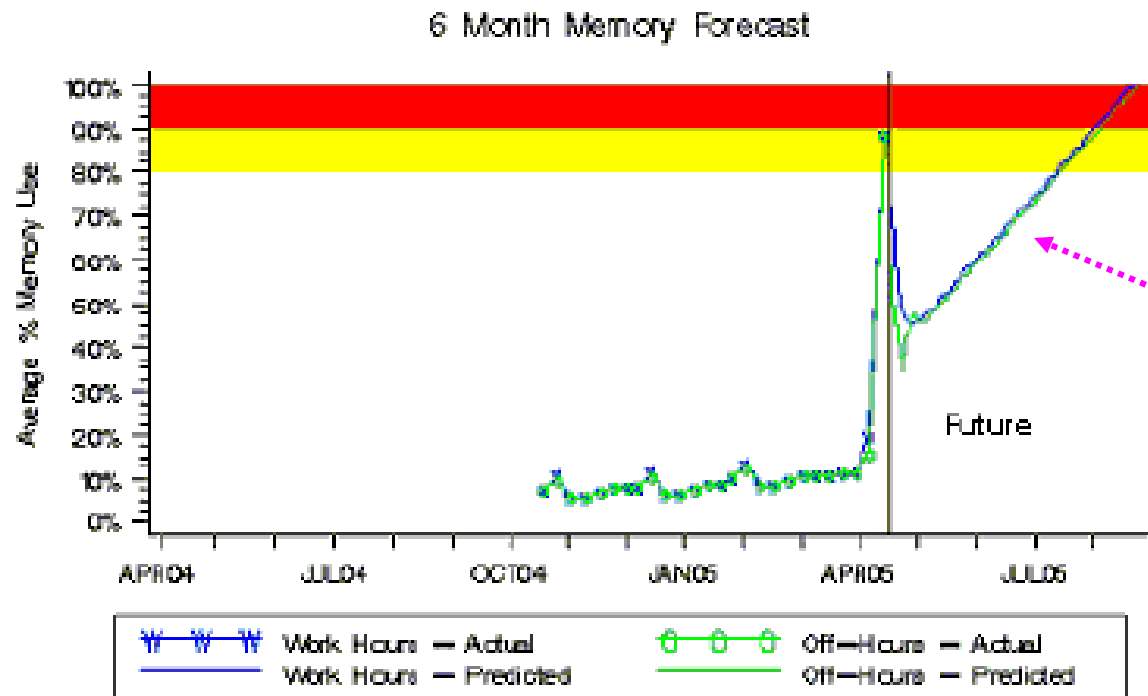


Forecast based on history started after adjustment point (**more realistic!**)

Forecast based on whole history (**wrong forecast!**)

## Basic Forecasting Rules. **RULE E: "Outliers"**

- The historical data with pathologies (run-away processes, memory leak situations and so on) causes inaccurate prediction
- That types of events along with some unexpected outages should be removed from the history as outliers



Memory leak issue spoiled the forecast

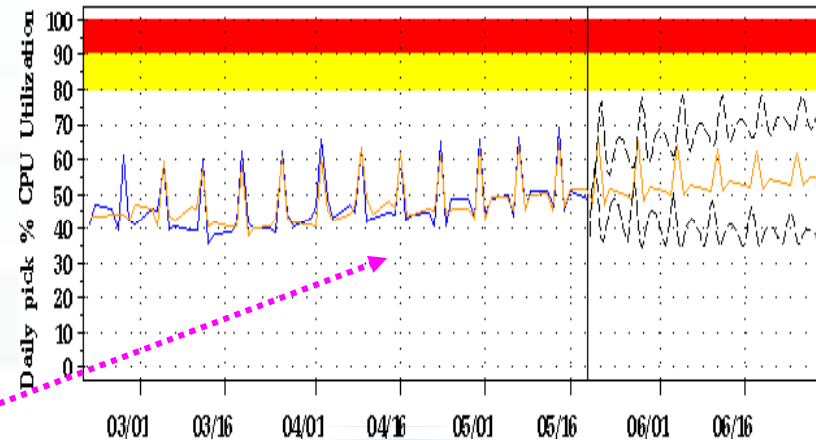
# Forecasting vs. Exception Detecting

■ Traditional way of Forecasting System development:

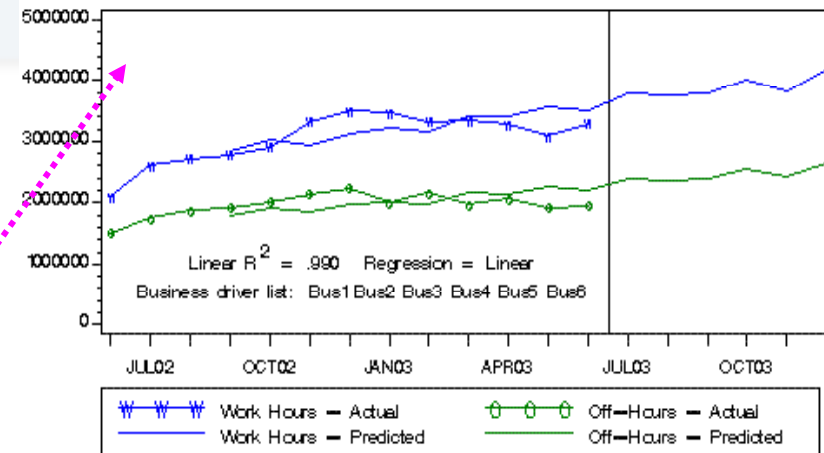
○ To produce trend-forecast and/or business correlated forecast charts for every server – **to many (and mostly boring) charts!**

○ To forecast when the future trend intersects with the threshold ( e.g. when the database used space metric intersects current allocation) – **Does not work when the threshold is unknown!**

30 Day CPU Utilization Forecast

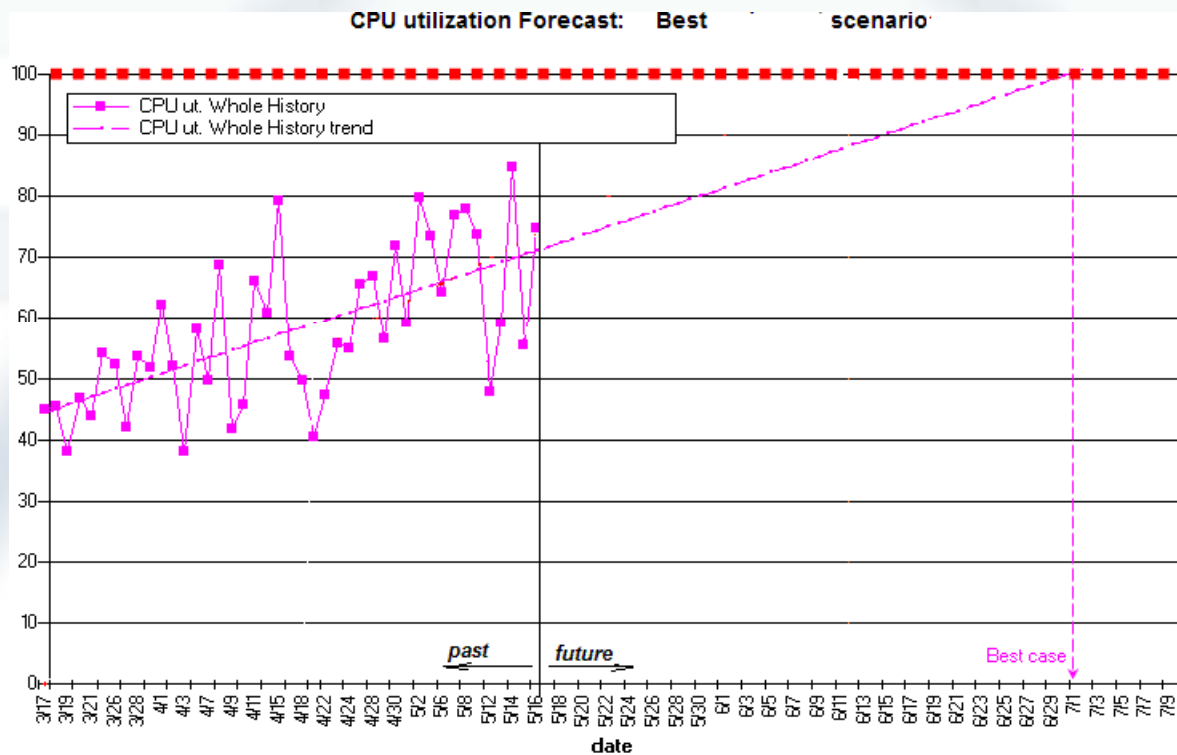


6 Month Disk I/O Forecast



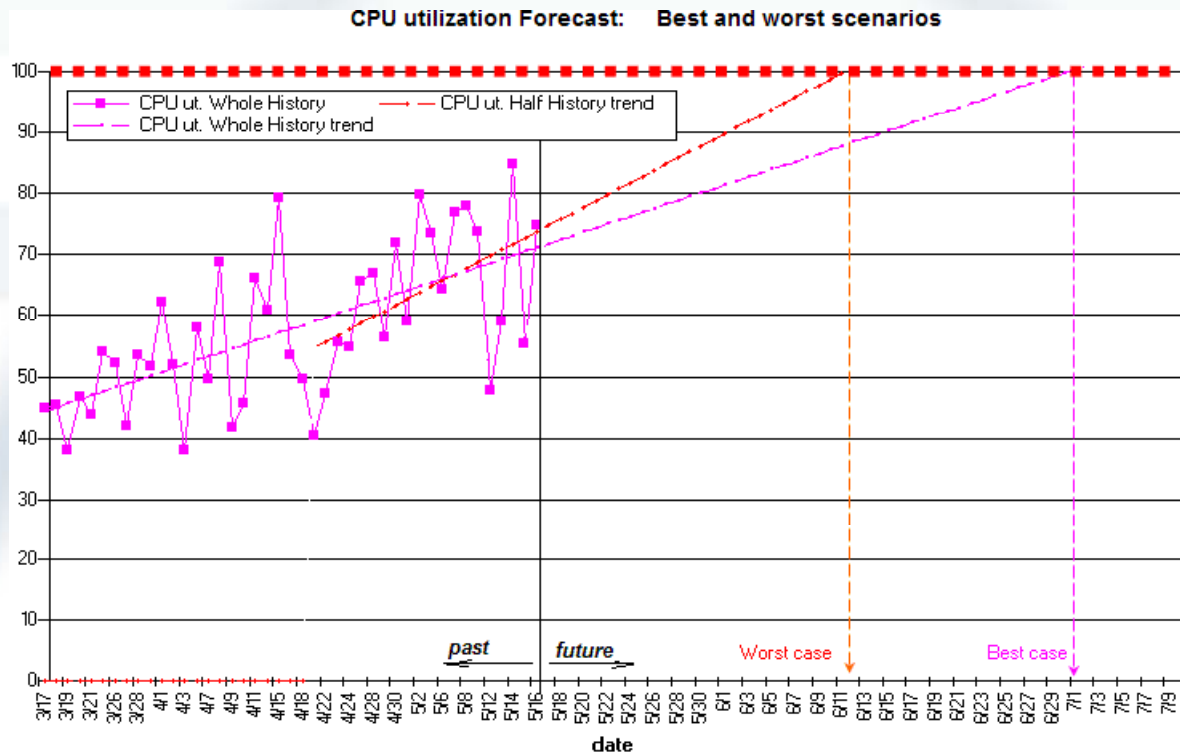
# Forecasting vs. Exception Detecting

- Improved way to do that is forecasting 2-3 times for short (,medium ) and long history sample data to show **the best scenario**



# Forecasting vs. Exception Detecting

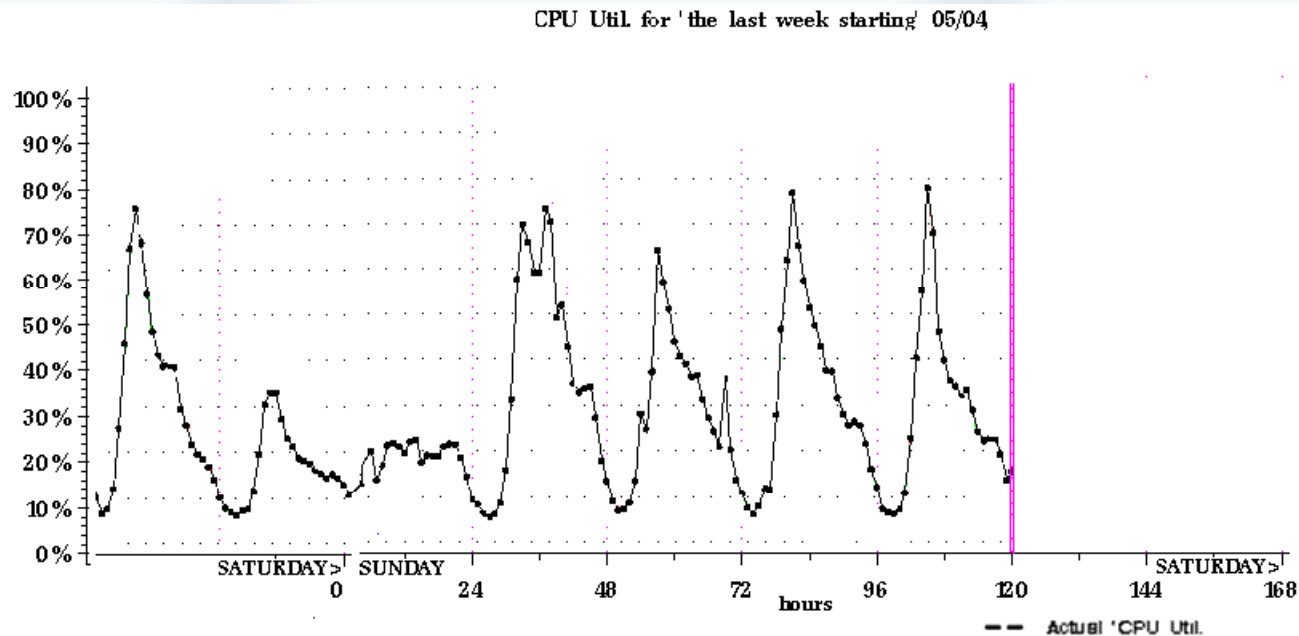
- Improved way to do that is forecasting 2-3 times for short (,medium ) and long history sample data to show **the best scenario** and **worst scenario**



[Link to spreadsheet](#)

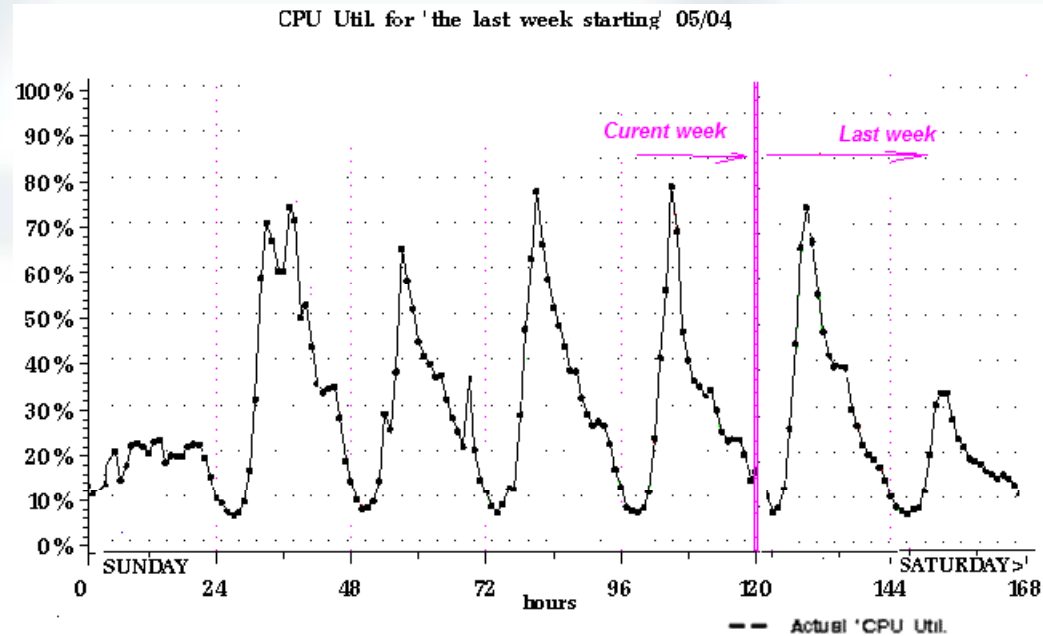
## Forecasting vs. Exception Detecting

- The most efficient way to do that is doing that on exception basis by using **Statistical Exception Detection System (SEDS)** and their main tool – **CONTROL CHART**:
  - Take one week of recent data



## Forecasting vs. Exception Detecting

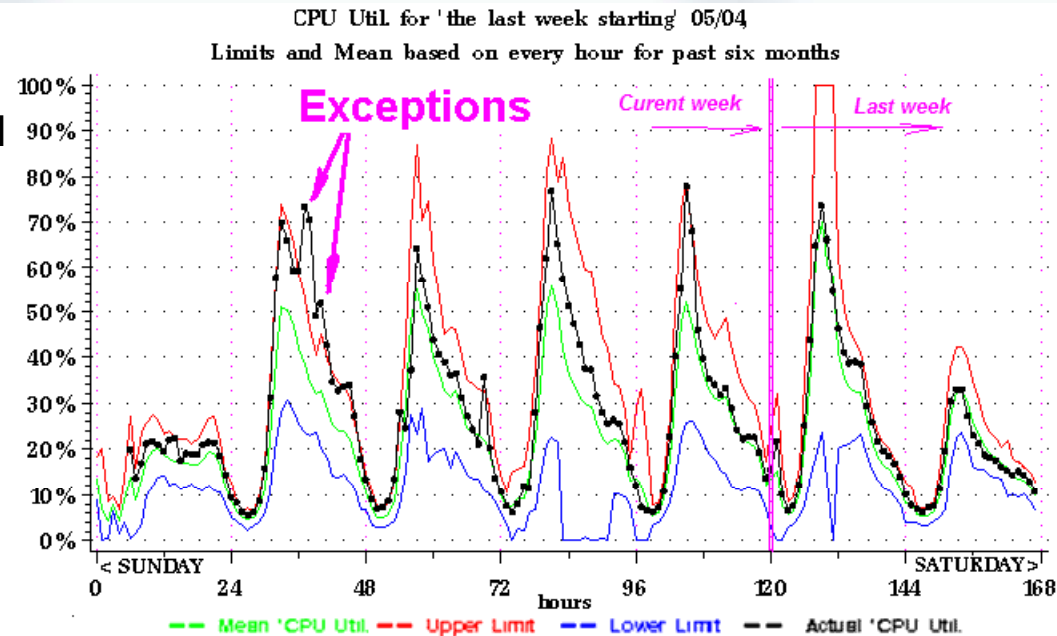
- The most efficient way to do that is doing that on exception basis by using **Statistical Exception Detection System (SEDS)** and their main tool – **CONTROL CHART**:
  - Take one week of recent data and put that in weekly profile format



## Forecasting vs. Exception Detecting

- The most efficient way to do that is doing that on exception basis by using **Statistical Exception Detection System (SEDS)** and their main tool – **CONTROL CHART**:
  - Take one week of recent data and put that in weekly profile form;
  - **Take** some representative historical reference data; set it as a baseline and then compare it with the most recent actual data.

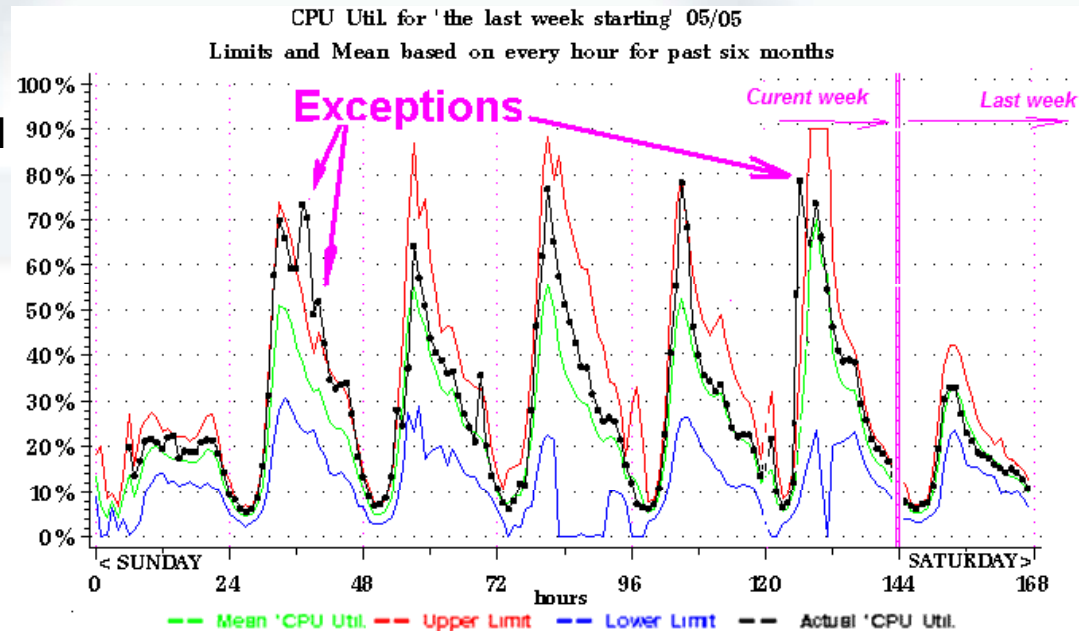
- If the actual data exceeds some statistical thresholds, (e.g. **Upper** (UCL) and **Lower** (LCL) Control Limits are **mean** plus/minus 3 standard deviations),
- generate an exception (alert via e-mail) and build a **control chart** .



## Forecasting vs. Exception Detecting

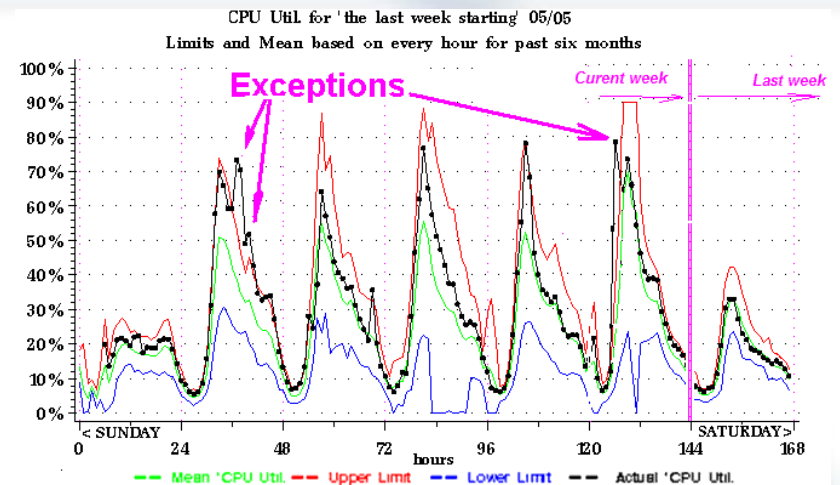
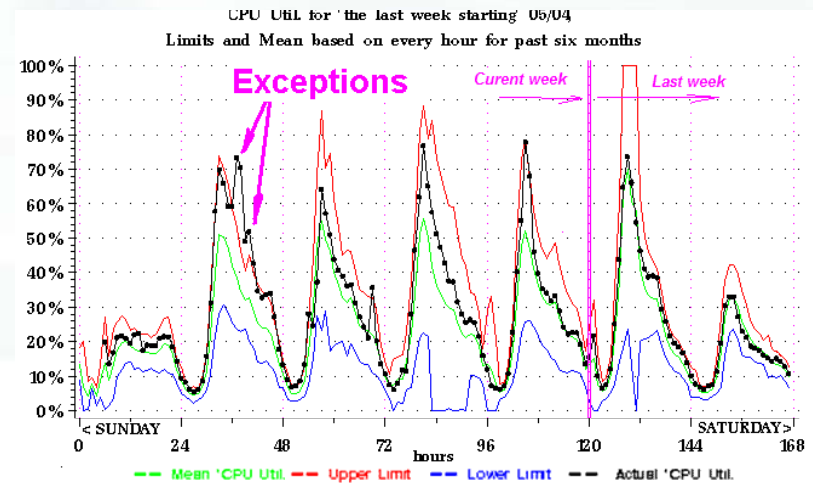
- The most efficient way to do that is doing that on exception basis by using **Statistical Exception Detection System (SEDS)** and their main tool – **CONTROL CHART**:
  - Take one week of recent data and put that in weekly profile form;
  - **Take** some representative historical reference data; set it as a baseline and then compare it with the most recent actual data.

- If the actual data exceeds some statistical thresholds, (e.g. **Upper** (UCL) and **Lower** (LCL) Control Limits are **mean** plus/minus 3 standard deviations),
- generate an exception (alert via e-mail) and build a **control chart** .



# Forecasting vs. Exception Detecting

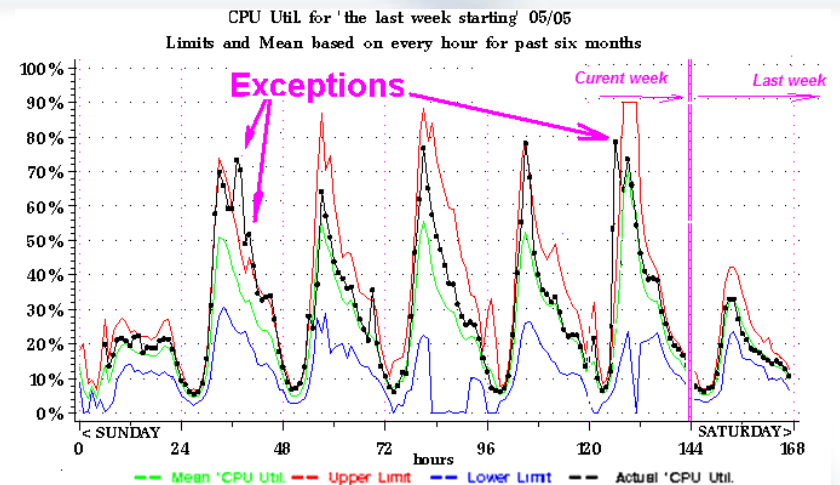
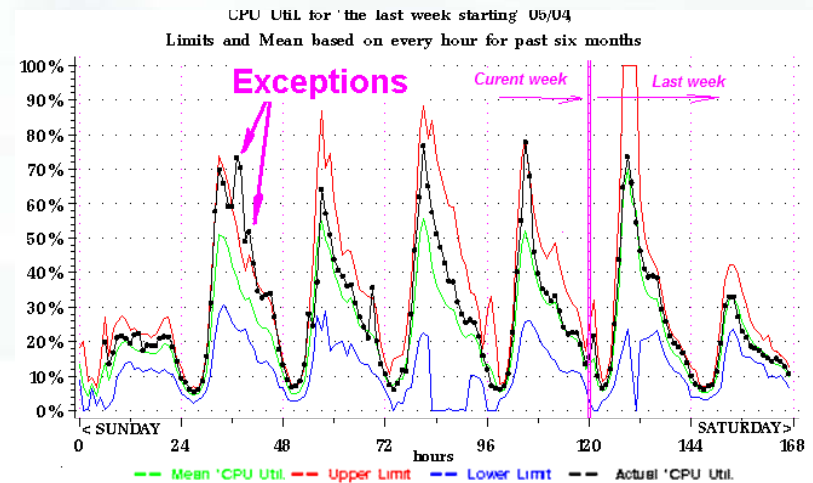
- Weekly Control Chart meets all basic forecasting rules:
  - **RULE A: "Summarization"** the Exception Detector summarizes data (6-8 month history of hourly data).
  - That allows you to see where system performance and business driver metrics correlate simply by analyzing control charts.



# Forecasting vs. Exception Detecting

■ Weekly Control Chart meets all basic forecasting rules:

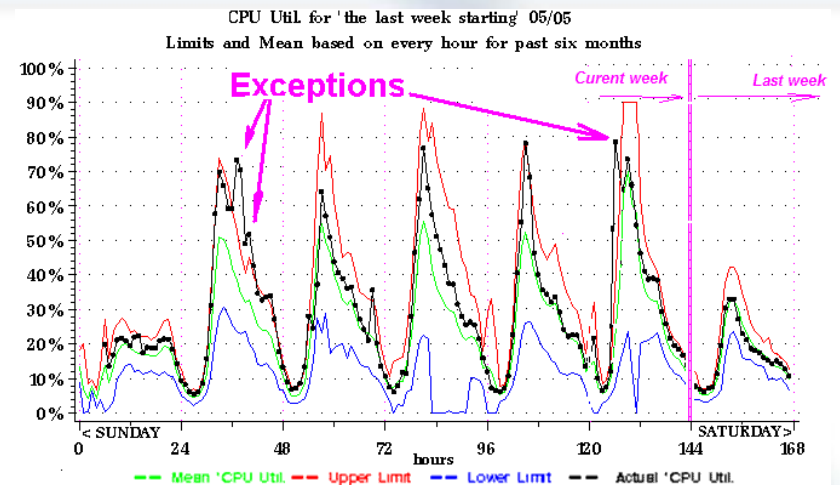
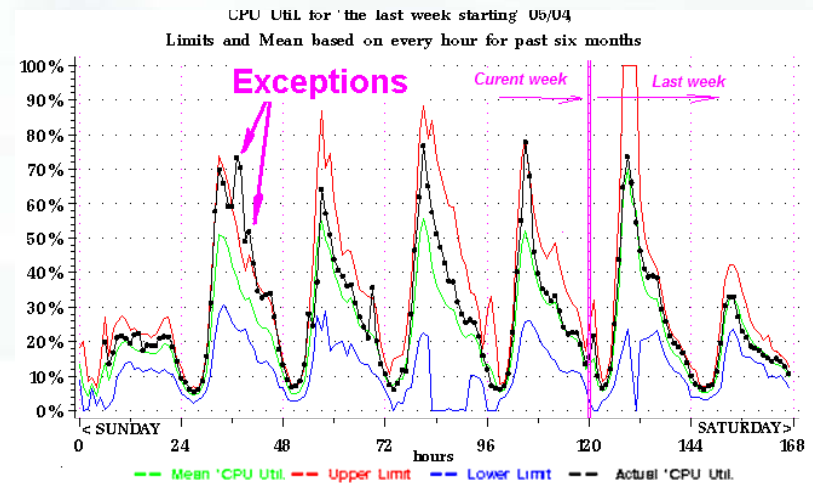
- **RULE B: “Do Not Mix Shifts”** Control chart by nature visualizes the separation of work or peak time and off time.
- **RULE C: “Statistical Model Choice”** means playing with different statistical limits (e.g. 1 st. dev. vs. 3 or more st. dev.) to tune the system and reduce the rate of false positives.



# Forecasting vs. Exception Detecting

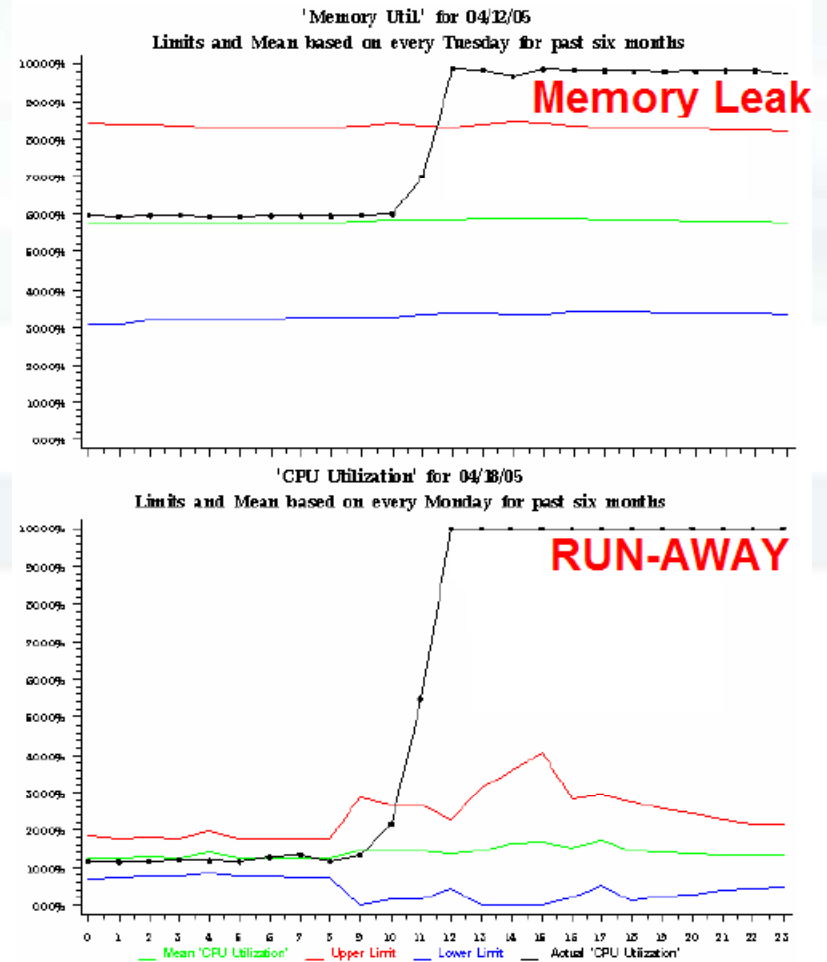
■ Weekly Control Chart meets all basic forecasting rules:

- **RULE D: “Significant Events”** is used to determine the depth of the reference set.
- The Exception Detector has the ability to adjust itself statistically to some events because the historical period follows the actual data and every event will occasionally be older than the oldest day in the reference set.



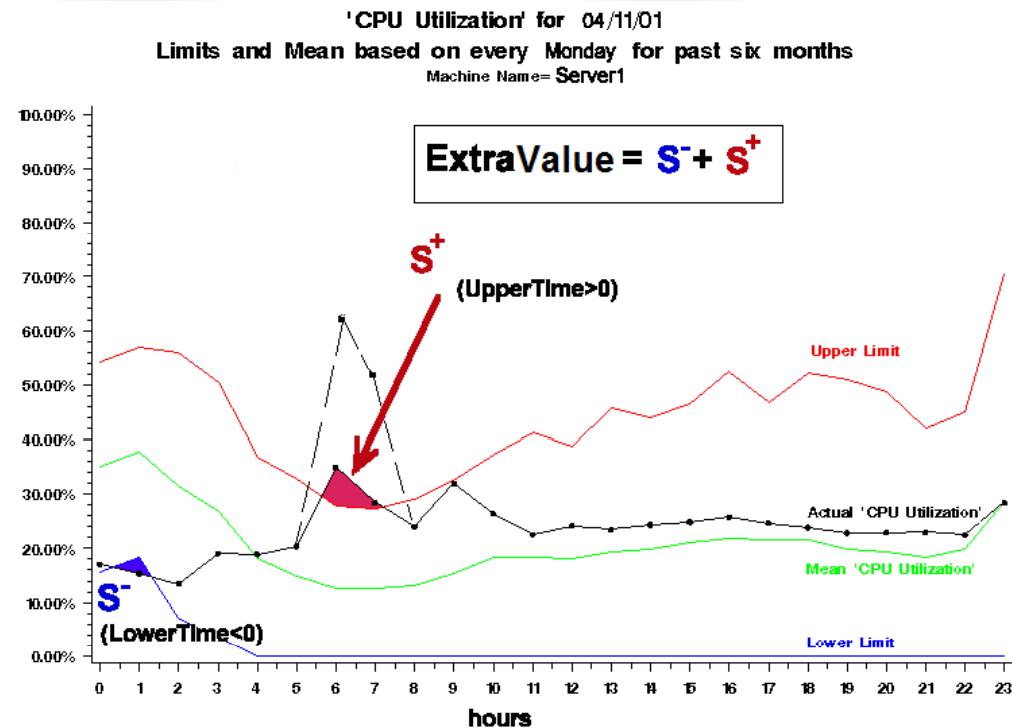
## Forecasting vs. Exception Detecting

- Weekly Control Chart meets all basic forecasting rules:
  - **RULE E: “Outliers”** All workload pathologies are statistically unusual. By adding some (non-statistical) filters to the system, the most severe of these pathologies should be excluded from the history to keep the reference set free from outliers.



## "ExtraValue" (EV) - a magnitude of exception

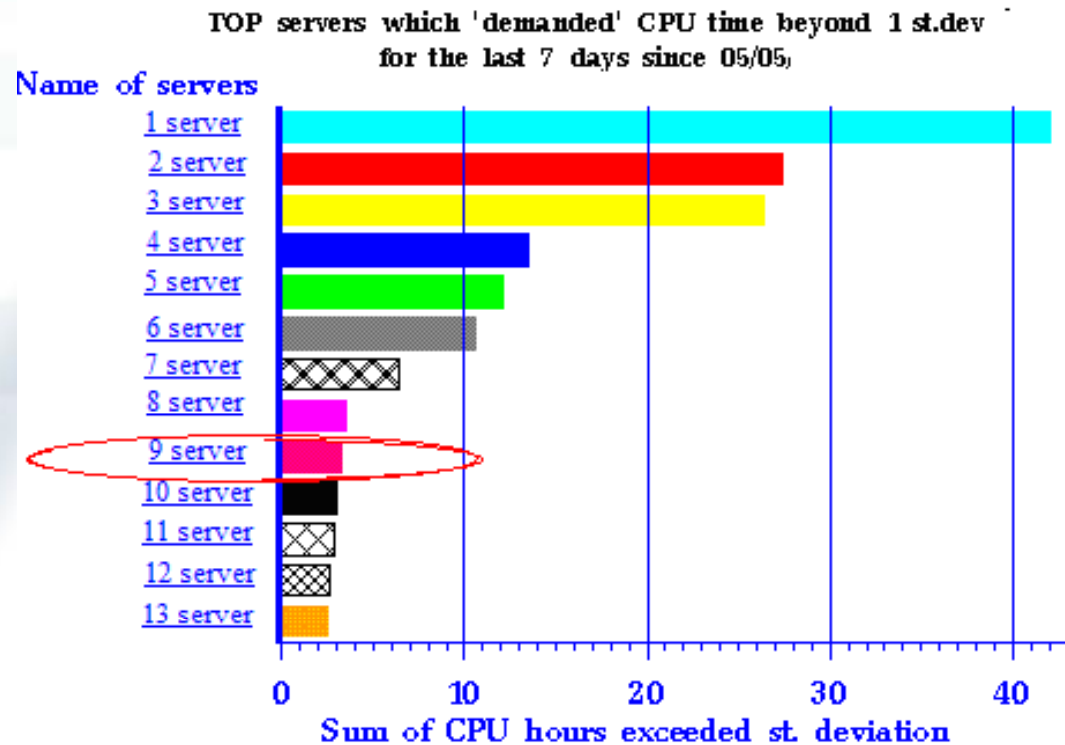
- To increase the accuracy of the Exception Detector and to reduce the number of the false positive situations (false alerting), a new meta-metric was added - "**ExtraValue**" (EV) of exception.



- It is the area between the actual data curve (black line on a control chart) and the statistical limit curves (red and yellow lines on the control chart)

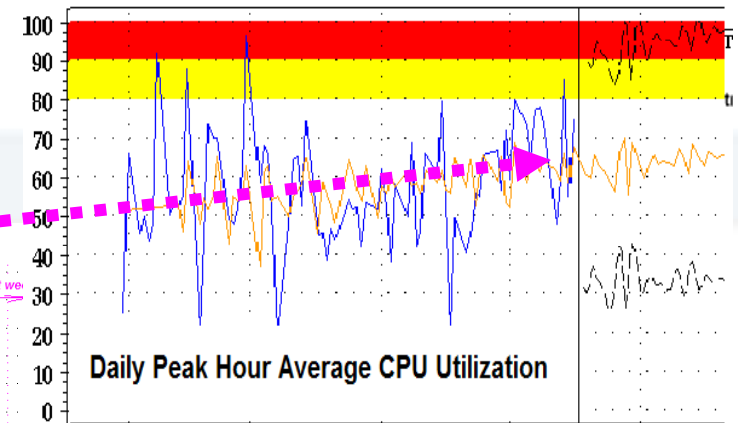
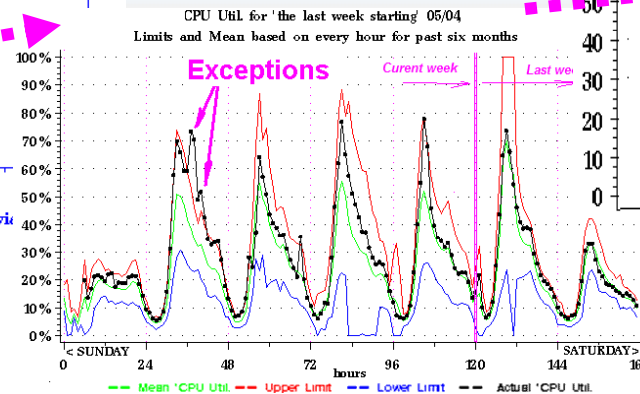
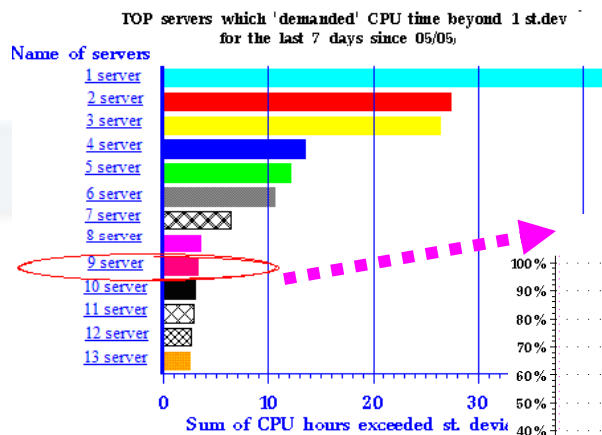
## "ExtraValue" (EV) - a magnitude of exception

- Publishing this top list in some way along with links to control charts significantly reduces the number of servers that require the focus of Capacity Planning or Performance Management analysts.



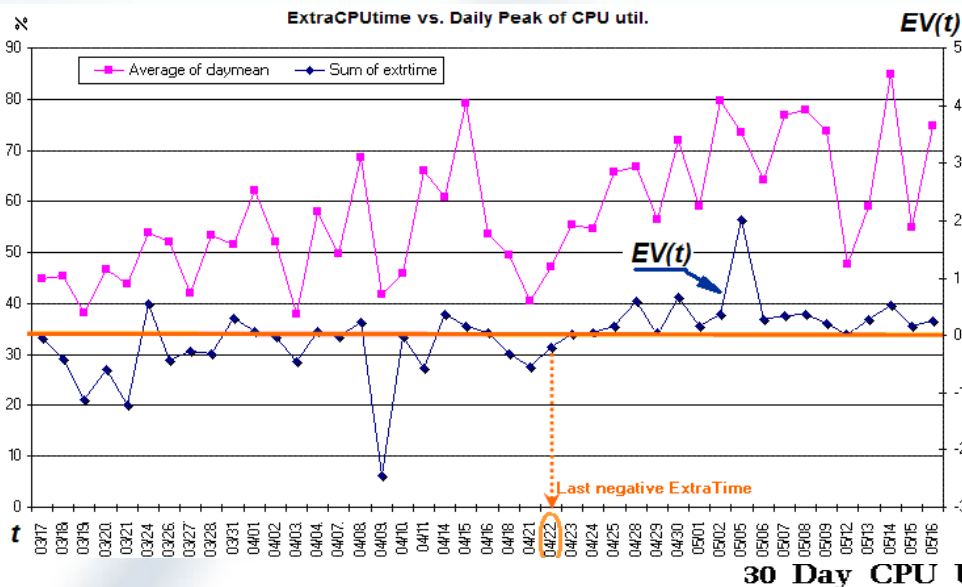
## Exception Based Forecasting method

- The data for the SEDS and forecasting system should be the same.
- The trend-forecast charts should be generated only for resources listed in the SEDS outputs.
- The data for trending analysis should be freed up from outliers based on Exception Detector pathology filters (e.g., free from run-away and memory leak days or hours).



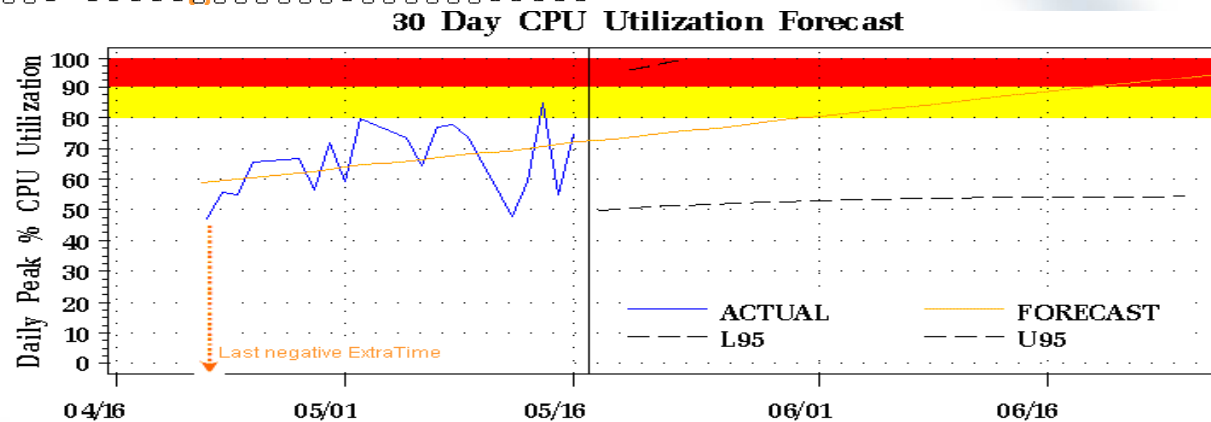
## Exception Based Forecasting method

- The starting time point (s) in the historical data for trending analysis can be found based on exception database data with "**ExtraValue**" metric records, as the most recent negative value of this metric indicates time when the data actually started trending up.



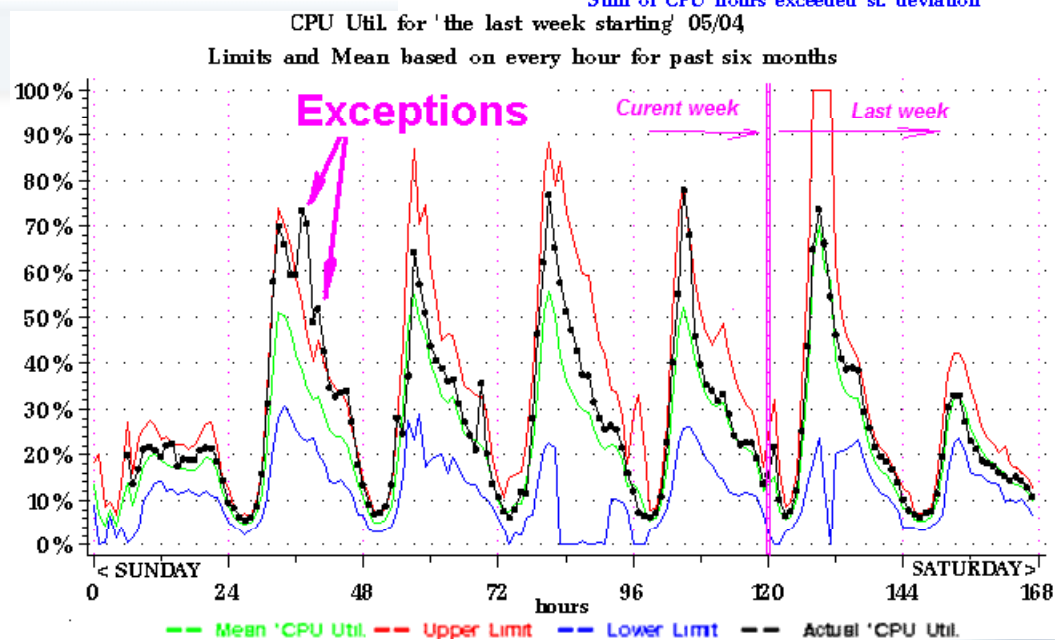
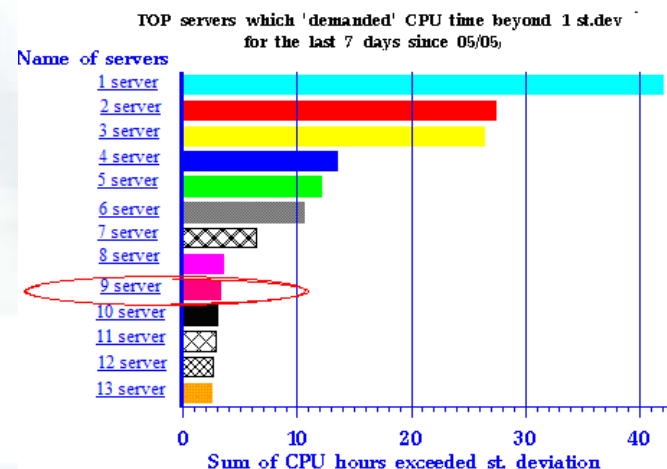
Basically, the starting point for analyzing the history is one of the roots in the following equation:

$$EV(t)=0$$



## Exception Based Forecasting: CASE STUDY

- One day, **server 9** has hit the exception list
- Some hourly exceptions occurred on Monday.
- During the entire previous week the actual CPU utilization was slightly higher than average (**green mean curve**).
- On Friday the upper limit (**red curve**) reached the 100% threshold for a few hours, which indicates that in the past the actual data might be at 100% level on other Fridays;
- ... it is a good idea to look at the historical trend !!!

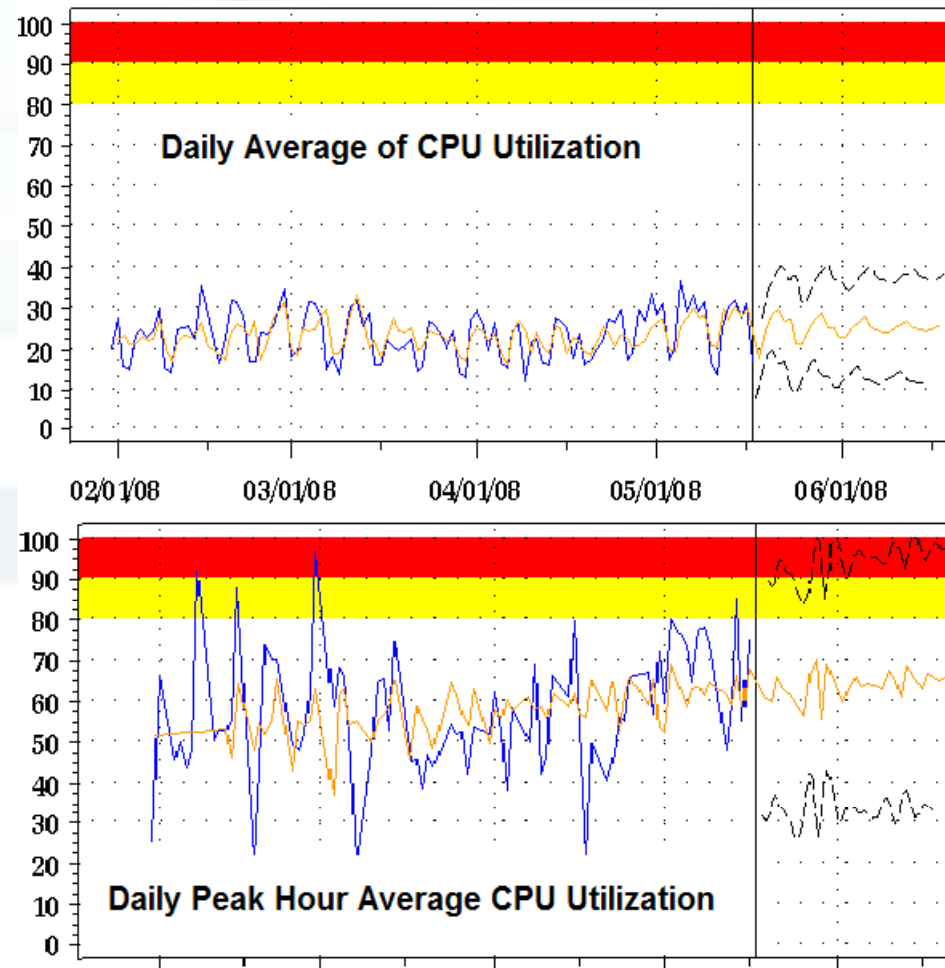


## CASE STUDY: Trend Forecast of Daily Average vs. Daily Peak

- Which presentation is better? It depends...

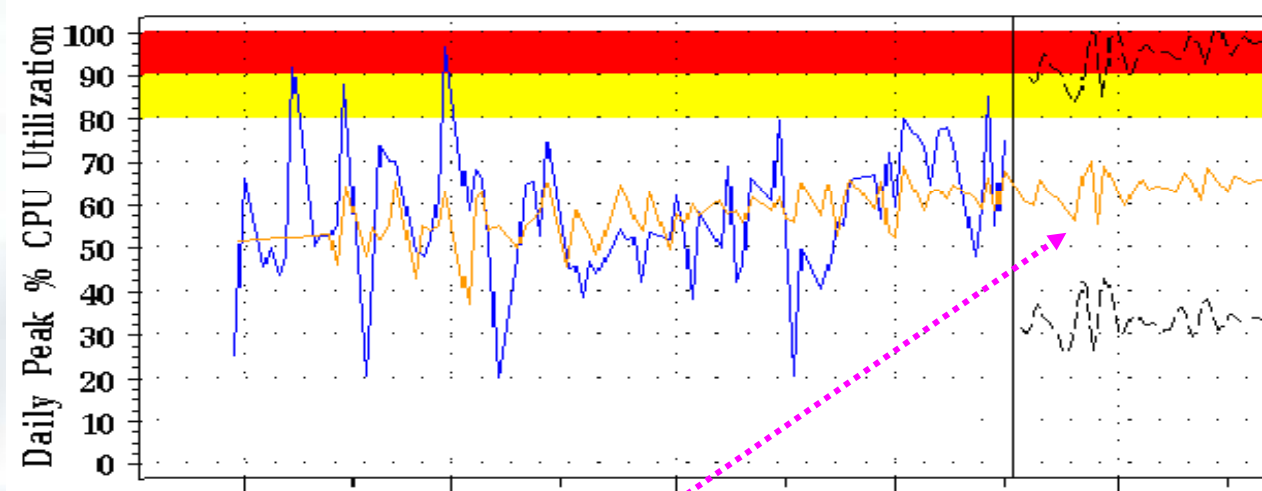
### **Recommendations are**

- Use the daily peak for OLTP or web application servers.
- Use daily average for back-up and other batch oriented application servers.

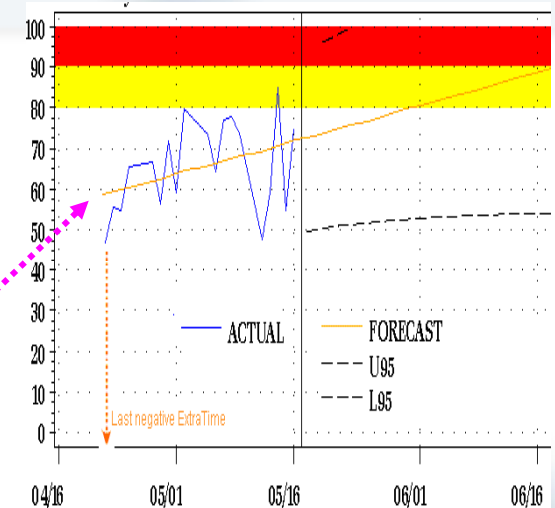


## CASE STUDY: Trend Forecast of Daily Average vs. Daily Peak

- Let's use Daily Peak as it is a web application server

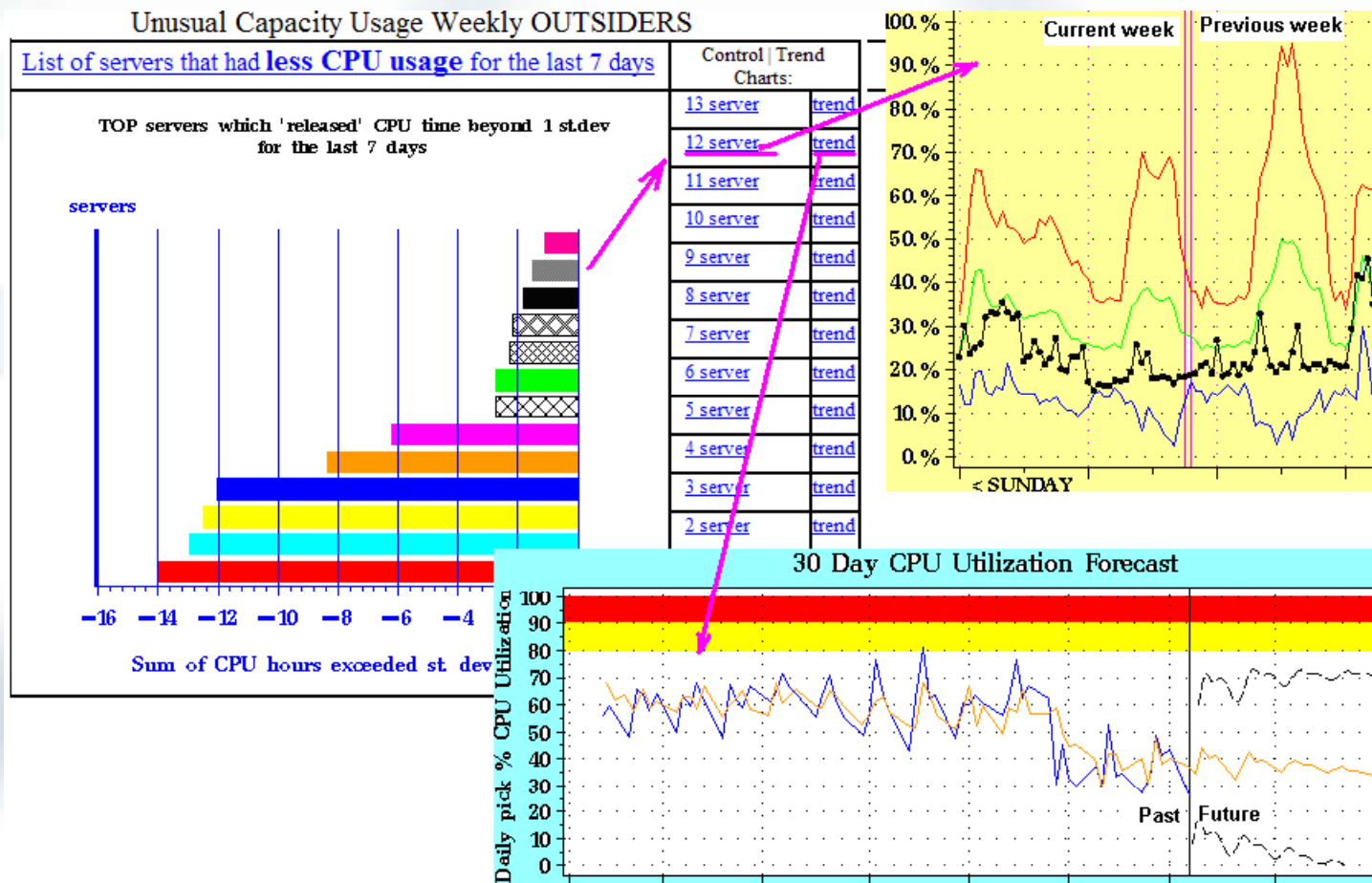


- Trend forecast, the future looks good. Why? Because **RULE D is not applied** and the entire history was used for forecasting.
- Taking in account only recent history the forecast looks more realistic



## Opposite side of the exceptions list

- Exceptions Detector (SEDS) publishes that as well and it makes perfect sense to build control and trend-forecast charts for each resource from that list:

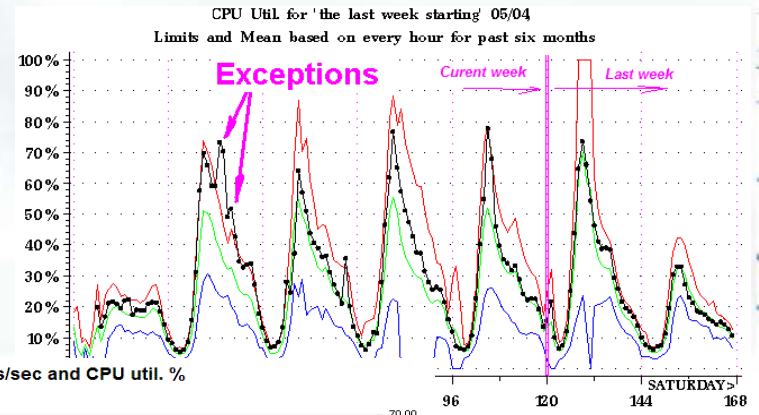


## Exception Based Modeling and Forecasting

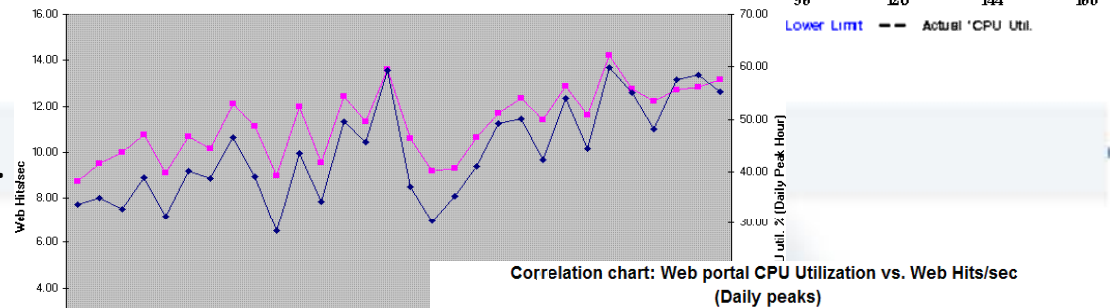
- How often do we need to perform modeling?
  - On **Exception basis**, when
    - there is a **project to upgrade** hardware or
    - to **consolidate resources** (applications, servers, VMs, LPARs or databases),
    - SEDS detected a capacity exception.
- What tools are good for that?
  - A **queuing theory** based analytical tool and/or **SAS** (expensive!)
  - What about to use just a **spreadsheet!**
    - ***Scattered*** charts
    - ***Add Trend line*** wizard
    - ***Forecast(...)*** formula or just **dragging** down a data range...

# Exception Based Modeling and Forecasting: Case Study

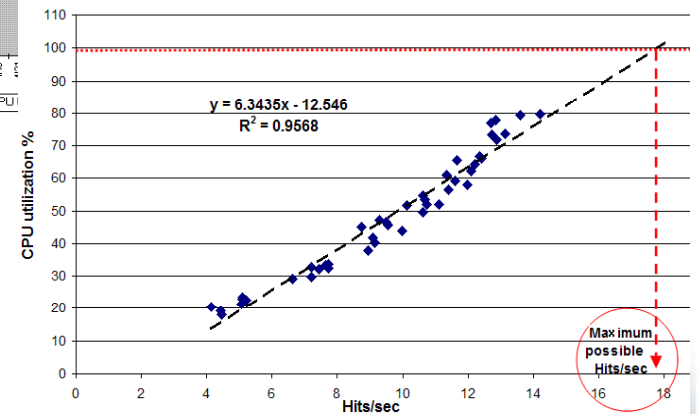
■ Other day, the Exception Detector notified that some web application had an unusual number of web hits producing the hits rate control chart similar to CPU utilization one



■ CPU and Web hits data were downloaded to a Spreadsheet for modeling.



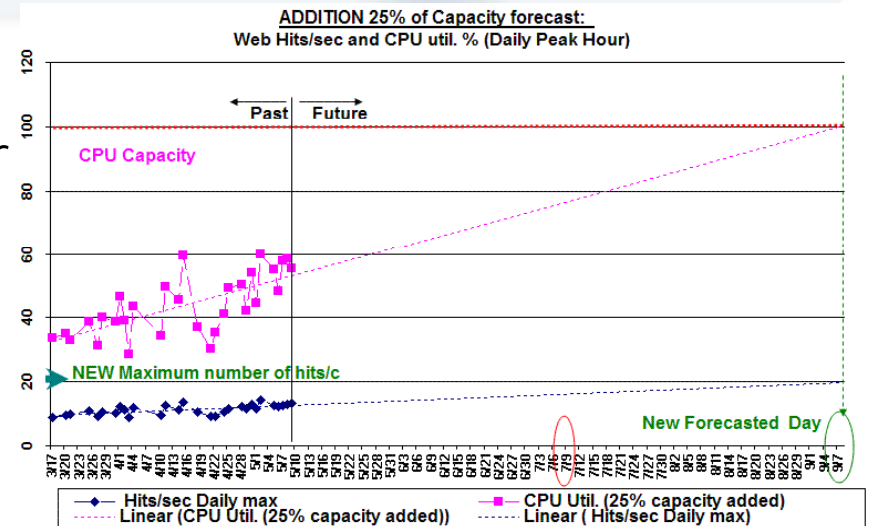
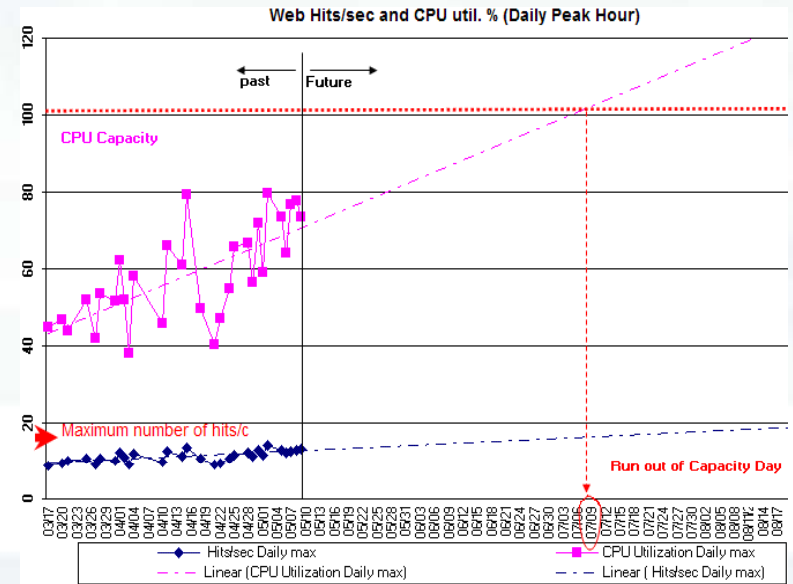
■ Combining those metrics in one scattered chart shows excellent correlation with  $R^2=0.96$ .



[Link To SPREADSHEET](#)

# Exception Based Modeling and Forecasting: Case Study Result

- The model shows the maximum number of **hits/Sec** that this server can handle is **18**.
- In two month the server **will be at capacity !!**
- If there is plan to get **more hits/Sec**, this server will need more processing capacity.
- Adding **25% more** processing power gives the ability to handle **20 hits/sec** and that capacity will be reached in a period **twice as long**.



[Link To SPREADSHEET](#)

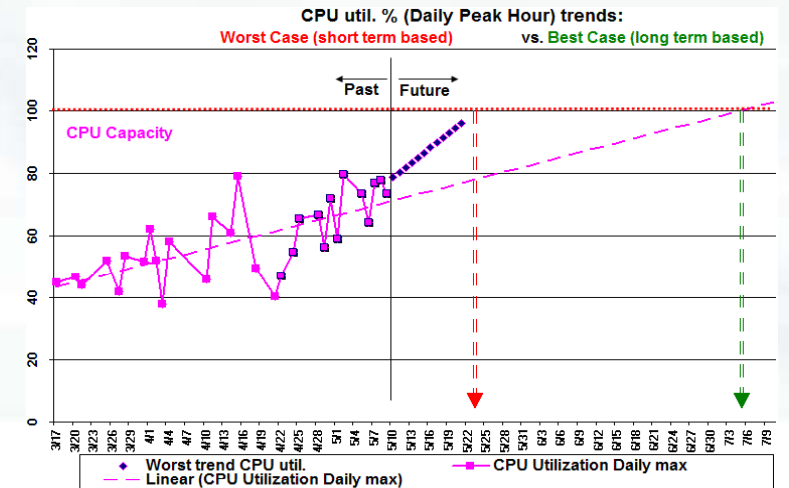
# Exception Based Modeling and Forecasting: Worst Case

- The most recent trend might be dangerous.
- Using Only Spreadsheet both trends could be built

Only spreadsheet features were used:

- The "**XY (scattered)**" standard chart.
- The "**Add trendline**" wizard with the coefficient =0.75 to reflect the proposed capacity increase.

- The future data were populated by just dragging down the selected range of historical data



	A	B	C	D
	date	Prime Shift CPU Utilization Daily max	Worst trend CPU util.	
1				
2	4/24	54.79	54.79	
3	4/25	65.66	65.66	
4	4/28	66.86	66.86	
5	4/29	56.40	56.40	
6	4/30	71.92	71.92	
7	5/1	59.17	59.17	
8	5/2	79.73	79.73	
9	5/5	73.45	73.45	
10	5/6	64.30	64.30	
11	5/7	76.86	76.86	
12	5/8	77.90	77.90	
13	5/9	73.66	73.66	
14	5/10		78.67	
15	5/11		80.25	
16	5/12		81.84	
17	5/13		83.42	
18	5/14		85.00	
19	5/15		86.58	
20	5/16		88.16	
21	5/17		89.74	
22	5/18		91.33	
23	5/19		92.91	
24	5/20		94.49	
25	5/21		96.07	
26	5/22		97.65	
27	5/23		99.24	
28	5/24		100.82	

[Link To SPREADSHEET](#)

# ExtraValue" (EV) meta-metric calculation technique

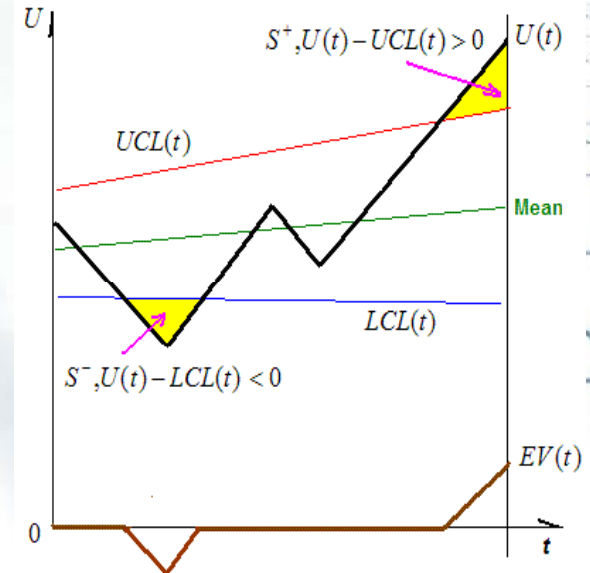
■ **ExtraValue" (EV)** is a magnitude of exception that occurred at a particular time with some metric

○ 2D model:

$$EV(t) = \begin{cases} S^+, U(t) - UCL(t) > 0 \\ S^-, U(t) - LCL(t) < 0 \\ 0, UCL(t) \leq U(t) \leq LCL(t) \end{cases}$$

where

$$S^+ = U(t) - UCL(t) \text{ and } S^- = U(t) - LCL(t)$$



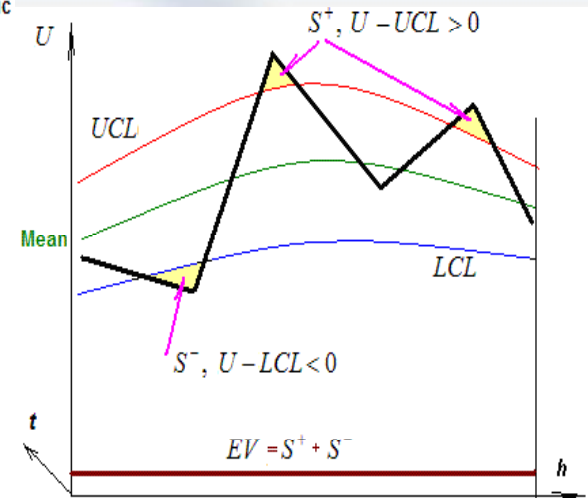
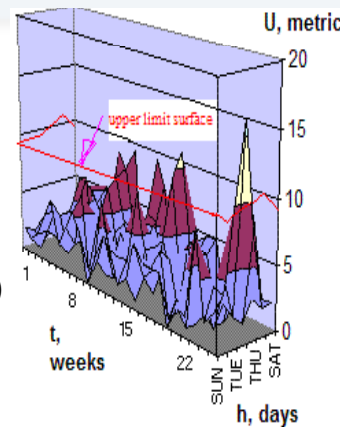
○ 3D model:

$$EV(t) = S^+ + S^-$$

where

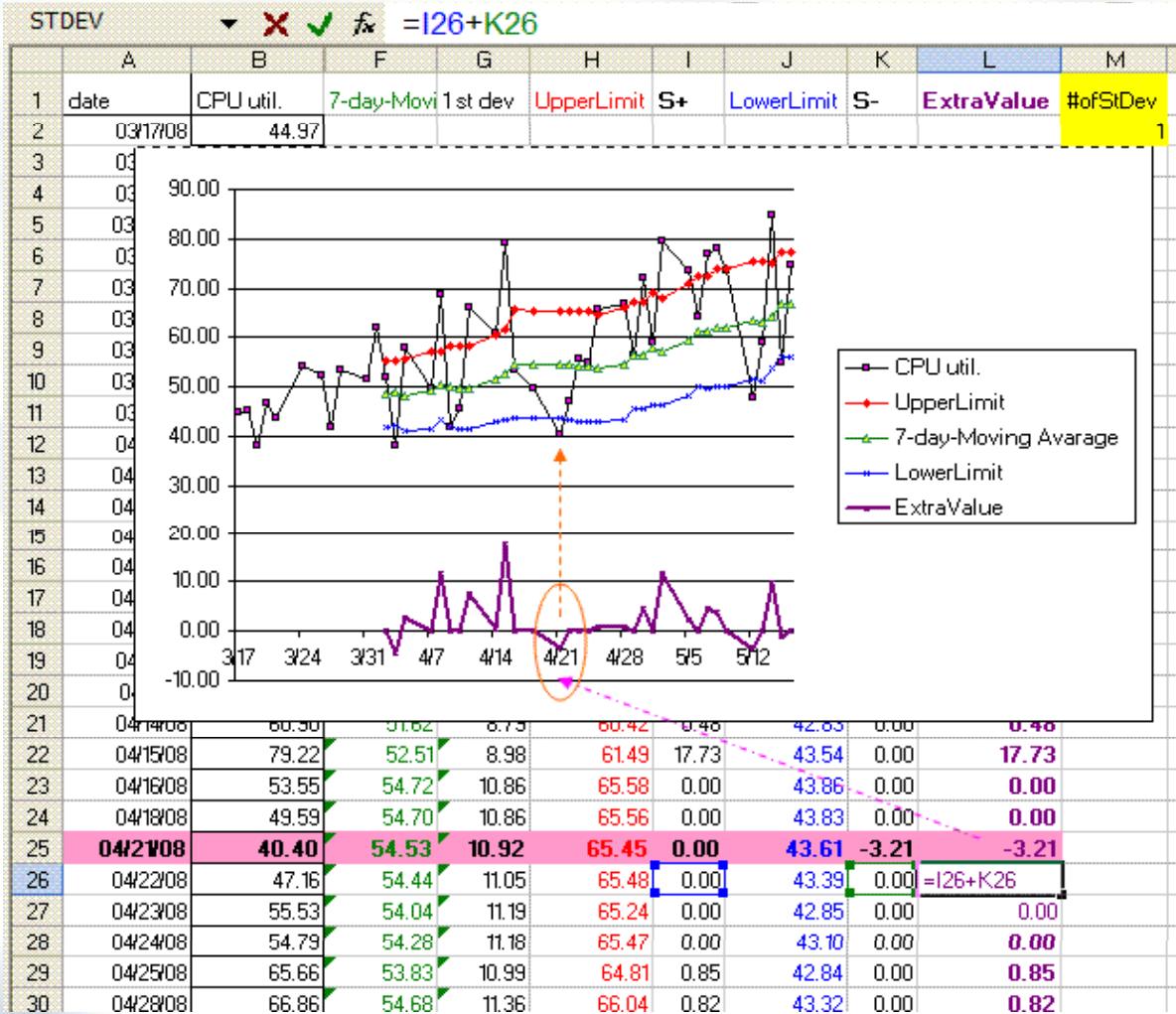
$$S^+ = \begin{cases} \int (U(h,t) - UCL(h,t)) dh, U - UCL > 0 \\ 0, U - UCL \leq 0 \end{cases}$$

$$S^- = \begin{cases} \int (U(h,t) - LCL(h,t)) dh, U - LCL < 0 \\ 0, U - LCL \geq 0 \end{cases}$$



# ExtraValue" (EV) meta-metric calculation technique

- EV and control chart could be built using spreadsheet:



**UpperLimit** = F + M\$2\*G = H

**LowerLimit** = F - M\$2\*G = J

**7-day-Moving Average**  
= AVERAGE(B:B+10) = F

**1 st. dev** = STDEV(B:B+10)

**S+** = IF(B-H < 0, 0, B-H) = I

**S-** = IF(B-J > 0, 0, B-J) = K

**EV = ExtraValue = I + K**

[Link To SPREADSHEET](#)

## Summary

- Capacity management in a large IT environment should perform forecasting and modeling only **when it is really needed**
- Exception Detection (**SEDS**) **techniques could be used** to automate the decision making process with regard to what needs to be modeled/forecasted and when.
- **Control Charts** have the ability to uncover some trends showing actual data deviations from an historical baseline.
- The most recent negative **ExtraValue** is an indicator of the moment of time when it is good to start the trending analysis of an historical sample.
- The **SEDS** approach helps to produce the trending analysis necessary for **cases without obvious thresholds**
- SEDS provides data (dates and hours) **to exclude outliers** for forecast improvement.
- Application data vs. Server performance data **correlation analysis** gives a priceless opportunity to add some meaning to forecasting/modeling studies and that analysis can be done using **standard spreadsheet tools**.

Questions?

Thank you!

Igor Trubin

Technical Blog:

<http://itrubin.blogspot.com/>