Paper 281-25

Using SAS® Software for Exception Analysis of Sybase Performance

Joseph Mirabal
America Online Inc.
12100 Sunrise Valley Dr.
Reston, VA 20191

Zhengyu Wang
Sybase, Inc.
6550 Rock Spring Drive
Bethesda, MD 20817

Abstract: Analyzing data that accounts for system behaviors is often required to define the technical limits, tuning parameters and resource capacity of database applications in an enterprise client-server computing environment. In a distributed environment exception analysis becomes a key instrument to determine corporate system resource management strategy. The approach presented in this paper is an information delivery of the dynamic corporate system requirements as they evolve. Sybase performance exception analysis is used as an experimental subject for data analysis and problem resolution of common concerns in business processing needs. The system lends itself to simple parametric analysis of any information base. We use three primary SAS products, Base SAS, SAS/STAT and SAS/Graph with the Access to Sybase and the Web publishing tools to build this application.

Introduction

Converting technical data into information is increasingly becoming a critical task for data centers. A huge amount of performance data can be collected via utilities, programs and third party tools. However, without appropriate data analysis, the voluminous data gets in the way of an effective review of activities, which leaves system administrators with surprises when problems arise. Much too often the problem symptoms can be identified long before their occurrence. Some examples include, concurrent user connections approaching the maximum number configured for the database server, CPU utilization inching into the 90th percent, physical I/Os not balanced across devices for the database and not enough cache memory to deliver the transaction response at user acceptable levels. Database storage capacity is another concern since disk space is a limited resource too. Whatever the situation, it is always a grueling experience for database or operating system administrators when they have to fix problems in a limited time frame without much preparation. We believe that exception analysis is able to help predict and issue warnings that potential problems may be on the way. Therefore, administrators and managers are able to fix the problems before they even happen. No matter whether the solution is changing a configuration parameter or contacting hardware vendors to increase resource capacity, exception analysis report provides evidence to procure resources or make adjustments on database servers or operating systems. This is becoming more and more critical in today's 24x7 e-business application systems where minimal downtime is required. Exception analysis also requires some degrees of sophistication and knowledge so that it is tested and proven in order not to mislead users and cause actions that may be costly with little or no beneficial results.

For the exception analysis to be useful, it must by necessity provide measurements of concern that have impacted or will soon impact performance of analytical elements, such as resources, applications or database objects. In the case of Sybase relational database management system the issues of concern relate to the resources it consumes and its efficiency of operation. An example would be estimating the number of days it takes to reach full storage capacity for certain databases. This can be accomplished by performing data analysis on historical database storage capacity information and testing the accuracy of calculation. Certain assumptions must also be made when using an analytical paradigm on different servers. For instance, some servers seem to be less affected by nearing capacity since their database growth is stagnant. However, on some other servers close capacity monitoring is necessary because of its rapid database growth. Therefore, the selection of parameters that define exception becomes our first task. Secondly, we study the data and determine the thresholds of acceptable performance. This can also include service level agreements where they exist. In a nutshell this activity involves the definition of service level objectives, which can be determined by studying the specific performance concerns. Usually they are intrinsically tied to the user experiences.

Data Collection

The key to initiating exception performance analysis is gathering proper data in an effective way. It can be accomplished by capturing Sybase performance data from multiple perspectives and storing it for evaluation and assessment. Sybase SQL Monitor Server, combined with Sybase Historical Server, provides Sybase SQL Server performance data in today's enterprise-wide client-server computing environment. Sybase SQL Server saves performance data in a shared memory area where Monitor Server reads in a manner that doesn't impact the server performance. Historical Server communicates with Monitor Server in sample intervals defined by users, captures the performance information and saves it in text files.

In our environment, since performance data from hundreds of servers are being collected, it is almost impractical to analyze the data saved in multiple flat files. The file size can also grow rapidly due to the volume of data collected and saved every day. Additionally, this data is normally of interest on a chronological or ad-hoc basis. In flat files, we have to read the entire file to access the data of interest, in an RDBMS you don't. In order to effectively manage data and facilitate efficient data analysis, it is necessary to consolidate these performance data into a central data store. In our case, a performance data warehouse was built on a Sybase SQL Server. Data collected by Historical Server is being copied into a performance database after sanity checks.

Several tables are defined in the performance database by combining related data items. Though Historical Server provides 100+ performance data metrics, we are particularly interested in CPU utilization, process status, server status, device activity, cache activity, object activity and procedure execution. Each of these performance tables is composed of multiple related data items. For instance, a "Device Activity" table consists of data items such as server name, time stamp, device name, number of hits and misses, number of I/O operations, and number of reads and writes. Performance data on all servers is consolidated into tables according to performance criteria, not server name. For instance, CPU utilization table includes CPU related performance data across all servers. This makes server-to-server performance comparison easy to accomplish in a SAS program.

Though Historical Server collects most of the performance data required by exception analysis, there are some data elements that are not defined in Historical Server. Database storage capacity is a good example. Instead of relying solely on Historical Server as our performance data collecting engine, several programs were developed to communicate with Sybase SQL Server directly and capture performance information. These programs consume resources on the server whenever they run. Therefore, we keep the number of these programs to a minimum and also have them run less frequently than Historical Server.

Data Analysis

Exception performance analysis can be easily employed in areas like resource exception. However, it encounters increasing complexity when the subject becomes applications or database objects. The root of all statistical research uses data analytic techniques to highlight problematic areas and focuses on immediate concerns. There are a number of SAS procedures that help us accomplish this. One of them is PROC UNIVARIATE. Other procedures include PROC SUMMARY and PROC MEANS that accomplish the same results. Each provides a different set of outputs from a data analysis standpoint. We prefer PROC UNIVARIATE because of the rich amount of information that can be received from its output. The Stem and Leaf Plots gives us an indication of how normally distributed the data is and provides a mechanism for defining the contingencies to focus on.

A sample of PROC UNIVARIATE output is shown below:

Variable=PCTCPBSY Percent CPU Busy

Moments

Mean	91.45409	Sum	4389.796
Std Dev	25.70145	Variance	660.5643
Skewness	-2.851	Kurtosis	2.436359
USS	43511.4	CSS	31046.52
CV	28.10311	Std Mean	3.709684
T:Mean=0	24.10311	Pr>|T|	0.0001
Num ^= 0	48	Num > 0	8
M(Sign)	24	Pr>=|M|	0.0001
Sgn Rank	588	Pr>=|S|	0.0001

Quantiles(Def=5)

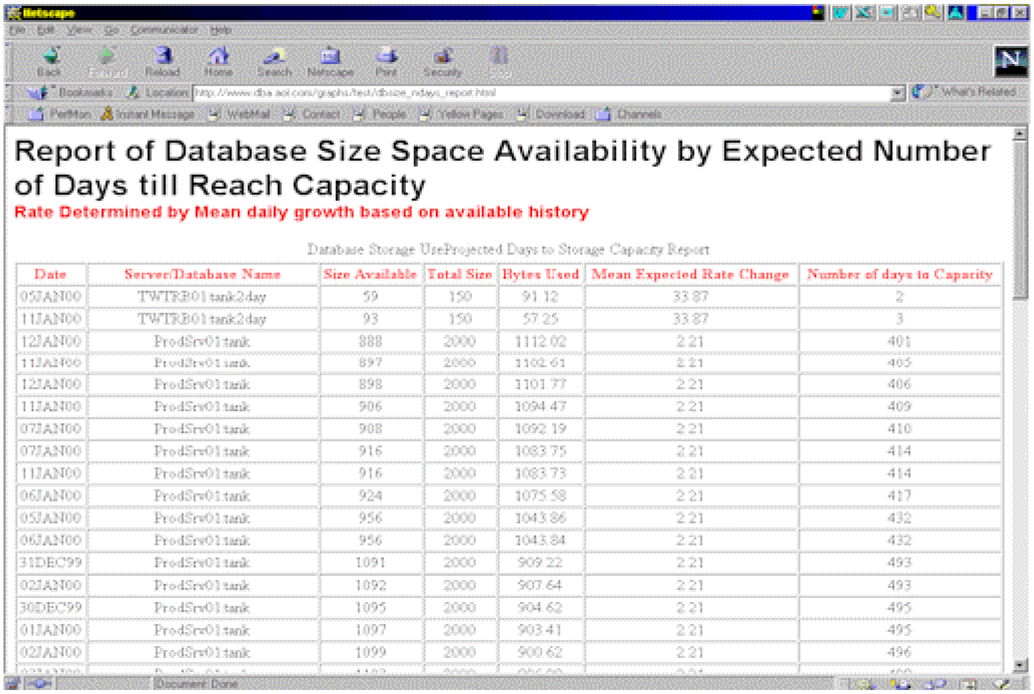
100% Max	100	99%	100
75% Q3	100	95%	100
50% Med	100	90%	100
25% Q1	99.86001	10%	49.62791
0% Min	6.644981	5%	8.77193
		1%	6.644981
Range	93.35502	Q3-Q1	0.139991
Mode	100		

Extremes

Lowest	Obs	Highest	Obs
6.644981(	48)	100(	37)
7.628866(	45)	100(	38)
8.77193 (	46)	100(	39)
19.72789(	47)	100(	42)
49.62791(	44)	100(	43)

The above example demonstrates the comprehensive analysis PROC UNIVARIATE can provide, especially when analyzing the data in a short time interval, such as within an hour or a day. In our example, PROC UNIVARIATE was applied on CPU utilization performance data collected within a specific day. Clearly these statistics show the gravity of capacity exception. Mean and standard deviation provide the classical analytical result to the distribution of processor utilization, which is typically linear. However, analyzing quantiles is more useful, since it provides inter-quartile ranges and key dectile (percentage distribution breakdown at 10% intervals) statistics that allow us to break down the distribution. An increasingly narrowing inter-quartile range at high usage levels indicates sustained consumption levels. These are not merely identified by simply checking the mean and maximum values. Without inter-quartile analysis, the analyst must diligently measure the delta between mean and the maximum values, since peak values of 100% processor utilization might be normal in some instances, however can mask capacity constraints.

Our performance exception analysis also includes analyzing daily database storage usage on Sybase servers in order to keep up with the database growth. It requires measurements of central tendency to define the usage pattern. Average database storage usage shows the database growth trend on a daily basis. Since adding and deleting records in the database causes constant ebb and flow of storage usage, measuring central tendency becomes a key indicator. In order to allow a sufficient grace period before the database runs out of space, we determine the average usage over the course of last five days and project the number of days until database space saturation. Database administrators are notified before the grace period so that they have sufficient time to take actions before database space reaches capacity. These actions include increasing database sizes or adding raw devices in order to support database growth. The result of database space exception analysis is specified in table format and presented to database administrators, as shown in Figure 1.



Report of Database Size Space Availability by Expected Number of Days till Reach Capacity

Rate Determined by Mean daily growth based on available history

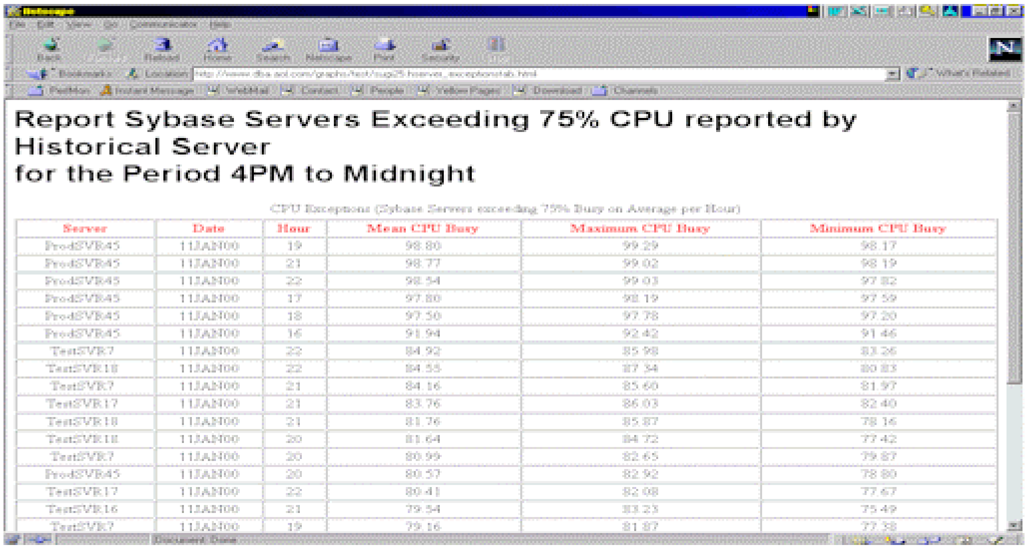
Database Storage UseProjected Days to Storage Capacity Report

Date	Server/Database Name	Size Available	Total Size	Bytes Used	Mean Expected Rate Change	Number of days to Capacity
05JAN00	TWTRB01tank2day	59	150	91.12	33.87	2
11JAN00	TWTRB01tank2day	93	150	57.25	33.87	3
12JAN00	ProdSrv01tank	888	2000	1112.02	2.21	401
11JAN00	ProdSrv01tank	897	2000	1102.61	2.21	405
12JAN00	ProdSrv01tank	898	2000	1101.77	2.21	406
11JAN00	ProdSrv01tank	906	2000	1094.47	2.21	409
07JAN00	ProdSrv01tank	908	2000	1092.19	2.21	410
07JAN00	ProdSrv01tank	916	2000	1083.75	2.21	414
11JAN00	ProdSrv01tank	916	2000	1083.73	2.21	414
06JAN00	ProdSrv01tank	924	2000	1075.58	2.21	417
05JAN00	ProdSrv01tank	956	2000	1043.86	2.21	432
06JAN00	ProdSrv01tank	956	2000	1043.84	2.21	432
31DEC99	ProdSrv01tank	1091	2000	909.22	2.21	493
02JAN00	ProdSrv01tank	1092	2000	907.64	2.21	493
30DEC99	ProdSrv01tank	1095	2000	904.62	2.21	495
01JAN00	ProdSrv01tank	1097	2000	903.41	2.21	495
02JAN00	ProdSrv01tank	1099	2000	900.62	2.21	496

Fig.1 Sybase Database Space Exception Analysis

We can apply a similar approach to analyze other key resources such as processor or memory usage. Usually processor usage is defined by CPU busy percentage. When CPU reaches 100% utilization, all transactions begin queuing, therefore system response slows down significantly. Rules of thumb have been developed to define the processing capacity that the system can sustain. As discussed above, adding storage capacity is a simple action of increasing space to the database or distributing the data space. However, in order to increase processor capacity, we have to add more CPUs, change existing CPUs to more powerful ones or migrating the database/application to a more scalable host. This is more complicated than adding storage capacity. Therefore an appropriate threshold defining near capacity level depends on the growth rate over the recent past. A less CPU-intensive system may experience the processing bottleneck when CPU capacity is near 80%. Some systems are able to process all transactions without slow response time even when CPU capacity is at the 95% level. In essence, the exception threshold is determined by several factors, among them are operating system, processor scalability, applications, service level objectives and/or key volume indicators for database applications. Whenever service level agreements are in place, the exception criteria is pre-defined. When we analyze multiple systems or components on the exception criteria, the exception threshold bar is lowered. The system's sensitivity to the exception may be greater for one system versus another. System administrators have to check the reported above-threshold CPU usage data and decide their corresponding actions based on their understanding of the applications. Therefore, if the tolerance level to performance impact is 95% CPU busy, the exception criteria would be lowered by 5 to 25 percent to accommodate reaction time for fully configured systems.

Figure 2 shows the table highlighting all servers whose CPU busy percentage is above 75%. Due to variety of applications in our environment, it is difficult to determine an exception percentage that is applicable across the board. System administrators have to make adjustment at their own discretion. Peak and non-peak hour CPU utilization is differentiated in the table so that decisions can be made in a business-oriented environment.


Report Sybase Servers Exceeding 75% CPU reported by Historical Server
for the Period 4PM to Midnight

CPU Exceptions (Sybase Servers exceeding 75% Busy on Average per Hour)

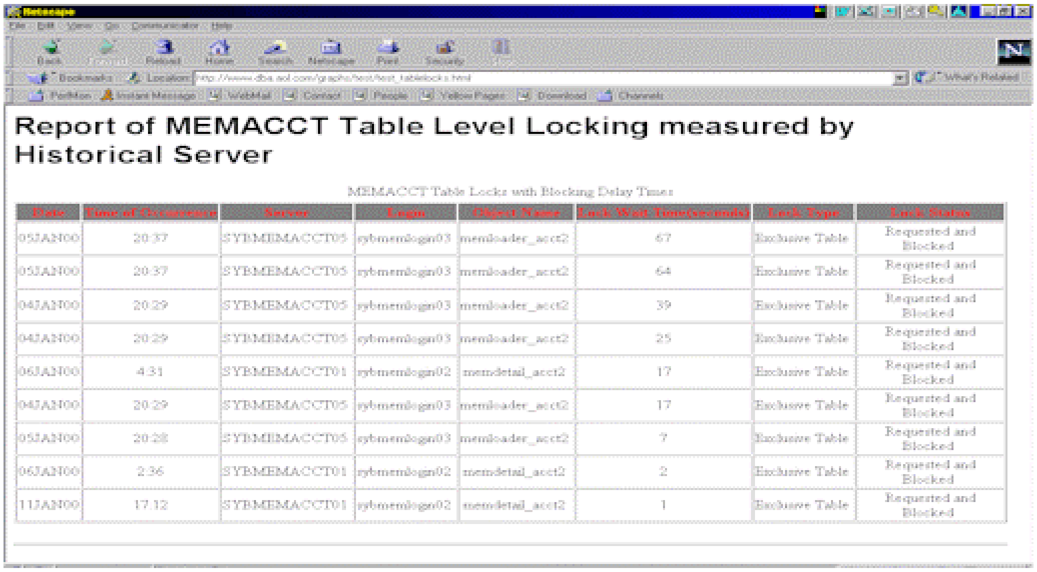
Server	Date	Hour	Mean CPU Busy	Maximum CPU Busy	Minimum CPU Busy
ProdSVR45	11JAN00	19	98.80	99.29	98.17
ProdSVR45	11JAN00	21	98.77	99.02	98.19
ProdSVR45	11JAN00	22	98.54	99.03	97.82
ProdSVR45	11JAN00	17	97.80	98.19	97.59
ProdSVR45	11JAN00	18	97.50	97.78	97.20
ProdSVR45	11JAN00	16	91.94	92.42	91.46
TestSVR7	11JAN00	22	84.92	85.98	83.26
TestSVR18	11JAN00	22	84.55	87.34	80.83
TestSVR7	11JAN00	21	84.16	85.60	81.97
TestSVR17	11JAN00	21	83.76	86.03	82.40
TestSVR18	11JAN00	21	81.76	85.87	78.16
TestSVR18	11JAN00	20	81.64	84.72	77.42
TestSVR7	11JAN00	20	80.99	82.65	79.87
ProdSVR45	11JAN00	20	80.57	82.92	78.80
TestSVR17	11JAN00	22	80.41	82.08	77.67
TestSVR16	11JAN00	21	79.54	83.23	75.49
TestSVR7	11JAN00	19	79.16	81.87	77.38

Fig.2 Sybase CPU Utilization Exception Analysis

Memory usage in a database system is identified by cache hit ratio. Cache hit ratio under 95% usually indicates memory constraints. Having ample memory reduces disk I/O, which improves performance, since memory access is much faster than disk access. While evaluating cache hit ratio at a database system level, it is beneficial to take into account memory usage at the operating system level. The actions to improve memory usage in a Sybase server include adding more memory, and other Sybase-specific tuning mechanisms, which is not the focus of this article. In general, memory usage is a very important aspect to evaluate system performance. Poor response time is usually the result of low cache hit ratios.

Locking activity is another area which exception analysis can benefit Sybase performance tuning. In a multi-user environment locking prevents several users from modifying the same data at the same time, so processes may be blocked until the lock is released. This significantly reduces concurrency, hence slows overall system throughput. Reducing blocking on locks requires analysis of the application code and the system architecture. Therefore, locking-related exception analysis focuses on blocking of process inside Sybase servers. Figure 3 shows the delay of processes waiting for lock releases.

After exception analysis highlights problematic areas, it is system administrator's task to take corresponding actions to correct the problems before the disaster happens. Sometimes it involves increasing resource capacity, such as adding disks, CPU and memory. Using SAS to analyze history through trends and project long-term growth provides substantial evidence for decision making.



Report of MEMACCT Table Level Locking measured by Historical Server

MEMACCT Table Locks with Blocking Delay Times

Date	Time of Occurrence	Server	Login	Object Name	Lock Wait Time(seconds)	Lock Type	Lock Status
05JAN00	20:37	SYBMEMACCT05	sybmemlogin03	memloader_acct2	67	Exclusive Table	Requested and Blocked
05JAN00	20:37	SYBMEMACCT05	sybmemlogin03	memloader_acct2	64	Exclusive Table	Requested and Blocked
04JAN00	20:29	SYBMEMACCT05	sybmemlogin03	memloader_acct2	39	Exclusive Table	Requested and Blocked
04JAN00	20:29	SYBMEMACCT05	sybmemlogin03	memloader_acct2	25	Exclusive Table	Requested and Blocked
06JAN00	4:31	SYBMEMACCT01	sybmemlogin02	memdetail_acct2	17	Exclusive Table	Requested and Blocked
04JAN00	20:29	SYBMEMACCT05	sybmemlogin03	memloader_acct2	17	Exclusive Table	Requested and Blocked
05JAN00	20:28	SYBMEMACCT05	sybmemlogin03	memloader_acct2	7	Exclusive Table	Requested and Blocked
06JAN00	2:36	SYBMEMACCT01	sybmemlogin02	memdetail_acct2	2	Exclusive Table	Requested and Blocked
11JAN00	17:12	SYBMEMACCT01	sybmemlogin02	memdetail_acct2	1	Exclusive Table	Requested and Blocked

Fig.3 Sybase Locking Activity Exception Analysis

Presentation

The best medium to deliver exception analysis results to users is the corporate intranet. The web browser grants users easy access to information on PC or UNIX workstations without installing additional software. Security is a concern during transmitting critical business or operational information (exception analysis results in our case), but with firewall technology and network security tools, the means to protect the data is within reach. The main objective here is to make the data easier to access. Consequently, we use frames in presenting web pages so that multiple exception analysis can be accessed by simple mouse clicks.

An example of a web page is shown in Figure 4.

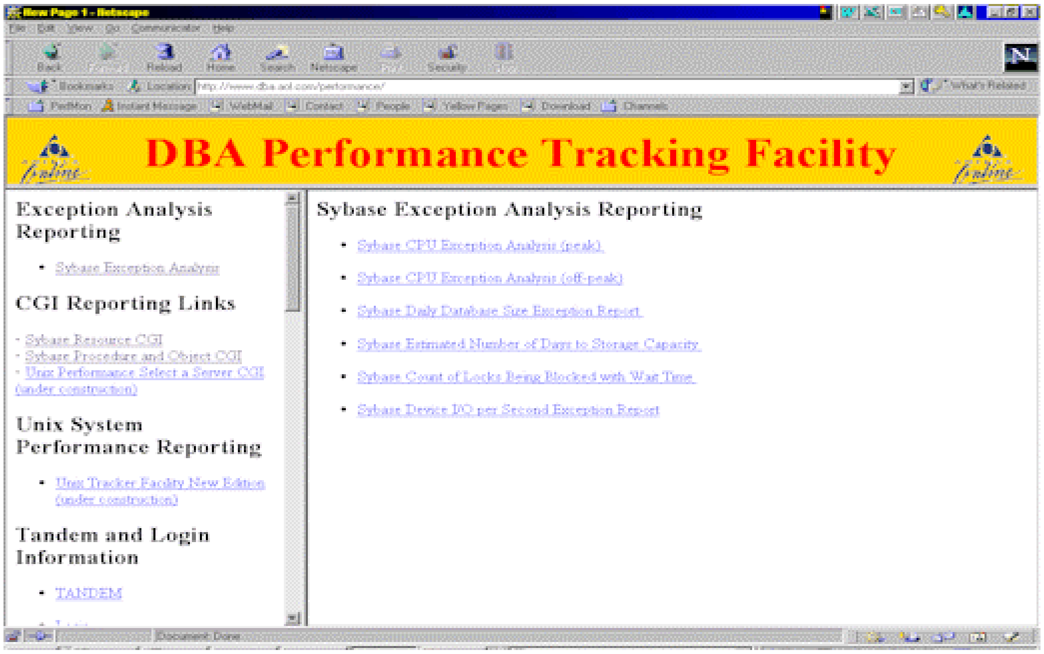


Fig.4 Sybase Exception Analysis Web Page

In this page we present information on all applications as well as Sybase exception analysis. This provides us the ability to view standard performance information and highlight problematic areas so that related causal factors can be checked simultaneously. The left page frame includes links to static and dynamic exception analysis information. The dynamic facility is based on CGI (Common Gateway Interface) to allow users to input the analytical criteria, such as time of day and parameter of interest. This gives us an avenue to explore the causes of the exception. For example, to answer questions like "Is the surge in resource use volume related to the exception?", a detailed analysis is required. The static pages are often defined by already proven analysis. These areas that have been studied provide intelligence about events encountered in the exception report. For example, CPU usage ceilings lead us to evaluate the trend and determine if this is an isolated incident or a testimony to a saturated processing environment. Finally, we look at a model of the system (a regression analysis of CPU usage over time) to determine the expected value of the system at some point in the future, as shown in Figure 5.

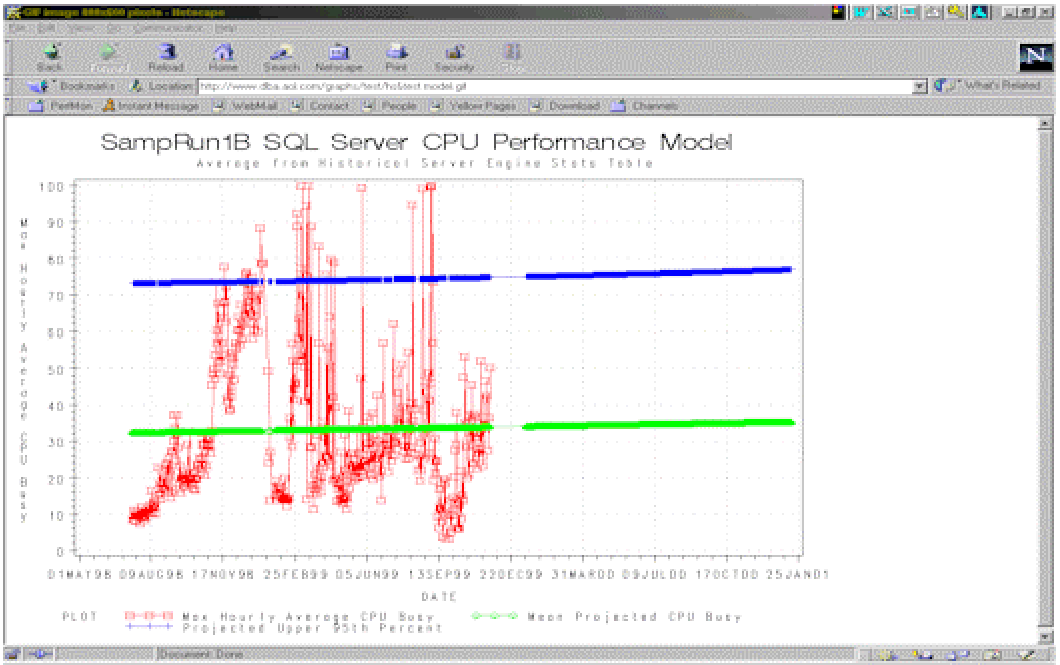


Fig.5 Sybase CPU Utilization Trend Analysis

Conclusion

SAS tools are optimal for analyzing quantitative data. It provides a quick reporting facility for data stored in database or flat files on most platforms. This flexibility becomes highly utilitarian in a multi-platform environment. The procedures presented in this paper provide a significant amount of value added to the operations of database-oriented application systems. The web is an optimal environment to present the exception analysis result so users at multiple locations can easily access the data.

With the performance exception analysis, technical personnel become better aware of performance issues and can discuss the potential solution without having to spend time identifying the problems. The availability of readily-developed analysis on resource usage and visibility of the path to solving problems translate the activity from identifying the problem to driving optimal solutions. One of the areas lacking in today's high stress computing centers is a solutions based performance metrics system rather than a problem based system. We hope that our paper presents approaches to that direction.

Contact Information

The authors can be contacted at

Joseph Mirabal
America Online, Inc.
12100 Sunrise Valley Drive
Reston, VA 20191
(703)265-4478
FAX: (703)265-4020
Email: jmmirabal@aol.com

Zhengyu Wang
Sybase Inc.
6550 Rock Spring Drive
Bethesda, MD 20817
(301)896-1127
FAX: (301)896-1602
Email: zwang@sybase.com