



Performance Testing Software Systems: Analyzing Performance Test Data

Derived from:

Microsoft patterns & practices

Performance Testing Guidance for Web Applications

By: J.D. Meier, Carlos Farre, Prashant Bansode, Scott Barber, Dennis Rea

© 2007 Microsoft Corporation. All rights reserved.

<http://www.codeplex.com/PerfTestingGuide>

Scott Barber

Chief Technologist

PerfTestPlus, Inc.



Performance Testing Software Systems



Scott Barber, CTO, PerfTestPlus, Inc.

sbarber@perftestplus.com

www.perftestplus.com



Co-Founder,
Workshop On Performance and Reliability

www.performance-workshop.org

Executive Director,
Association for Software Testing

www.associationforsoftwaretesting.org



Co-Author,

Performance Testing Guidance for Web Applications

www.codeplex.com/PerfTestingGuide

www.amazon.com/gp/product/0735625700





Credits

Some of this material was developed for, or inspired by, *Performance Testing Guidance for Web Applications*, a Microsoft patterns & practices book by J.D. Meier, Scott Barber, Carlos Farre, Prashant Bansode, and Dennis Rea.

Many ideas in this course were inspired or enhanced by colleagues including Alberto Savoia, Roland Stens, Richard Leeke, Mike Kelly, Nate White, Rob Sabourin, Chris Loosley, Ross Collard, Jon Bach, James Bach, Jerry Weinberg, Cem Kaner, Dawn Haynes, Karen Johnson, and the entire WOPR community.

Most of the concepts in this presentation are derived from publications, presentations, and research written and/or conducted by Scott Barber.

Many ideas were improved by students who took previous versions of this course, back to 2001.

This course has been heavily influenced by:

Rapid Software Testing (James Bach & Michael Bolton, ©1995-2007 Satisfice, Inc.)

Just-In-Time Testing (Robert Sabourin, ©1998-2007 Amibug, Inc.)





I Assume That You:

Test software performance or manage someone(s) who does.

Have at least some control over the design of your tests and some time to create new tests.

Have at least some influence over your test environment.

Are worried that your test process is spending too much time and resources on things that aren't important AND/OR

Are worried that your test process doesn't leave enough time and resources to determine what IS important.

Believe that good testing requires thinking.

Test under uncertainty, resource limitations and time pressure.

Have a major goal to find important problems quickly.

Want to get very good at testing software performance.





*“There is no such thing as a
‘junior performance tester’...*

*but there are people who are new
to performance testing.”*

--Scott Barber





Instructional Methods That I Use

The Class Presents My Editorial Opinions: I do not make appeals to authority; I speak only from my experiences, and I appeal to your experience and intelligence.

Not All Slides Will be Discussed: There is *much* more material here than I can cover in detail, so I may skip some of it. (If you want me to go back to something that I skipped, just ask.)

I Need to Hear from You: You control what you think and do, so I encourage your *questions about* and *challenges to* the lecture. (Talk to me during the break, too.)

If You Want Specifics, Bring Specifics: I invite you to bring real examples of testing problems and test documents to class. (I am happy to show you how I would work through them.)

The Exercises are the Most Important Part: I sometimes use *immersive socratic exercises* that are designed to fool you if you don't ask questions. I usually do not provide all the information you need. *Asking questions is a fundamental testing skill!*

Slide Adapted from *Rapid Software Testing* by James Bach & Michael Bolton, © 1995-2007, Satisfice, Inc.





Instructional Methods That I Use

I am likely to push you

If I call on you,
and you don't want to be put on the spot,
just say **"Pass!"**

But you can push back

Slide Adapted from *Rapid Software Testing* by James Bach & Michael Bolton, © 1995-2007, Satisfice, Inc.





What Not to Expect From Me

Untested theory.

Marketing fluff.

Pulled punches to protect the guilty.

The “One True Answer” to anything.

Every concept to apply, precisely as presented, to every context.

Over simplifications without acknowledgement.

A disimpassioned, boring instructor!





Primary Goal of this Course

To teach you how to
think about, *organize*, *manage*, and *report*
performance test data effectively,
under time and resource constraints.





*“Let’s face the truth, performance testing
IS rocket science.”*

--Dawn Haynes





Performance Testing Principles

Context

Project context is central to successful performance testing.

Criteria

Business, project, system, & user success criteria.

Design

Identify system usage, and key metrics; plan and design tests.

Install

Install and prepare environment, tools, & resource monitors.

Script

Script the performance tests as designed.

Execute

Run and monitor tests. Validate tests, test data, and results.

Analyze

Analyze the data individually and as a cross-functional team.

Report

Consolidate and share results, customized by audience.

Iterate

"Lather, rinse, repeat" as necessary.





Context





Context

Do you know your performance testing mission?

Do you know the “Commander’s Intent”?

Can you find out?

Might COPE in PUBS help?

Example from my days as a U.S. Army LT:

Mission: Secure hilltop 42 NLT 0545 tomorrow.

Commander’s Intent: It is my intent that the supply convoy safely cross the bridge spanning the gorge between hilltop 42 and hilltop 57 between 0553 and 0558 tomorrow.





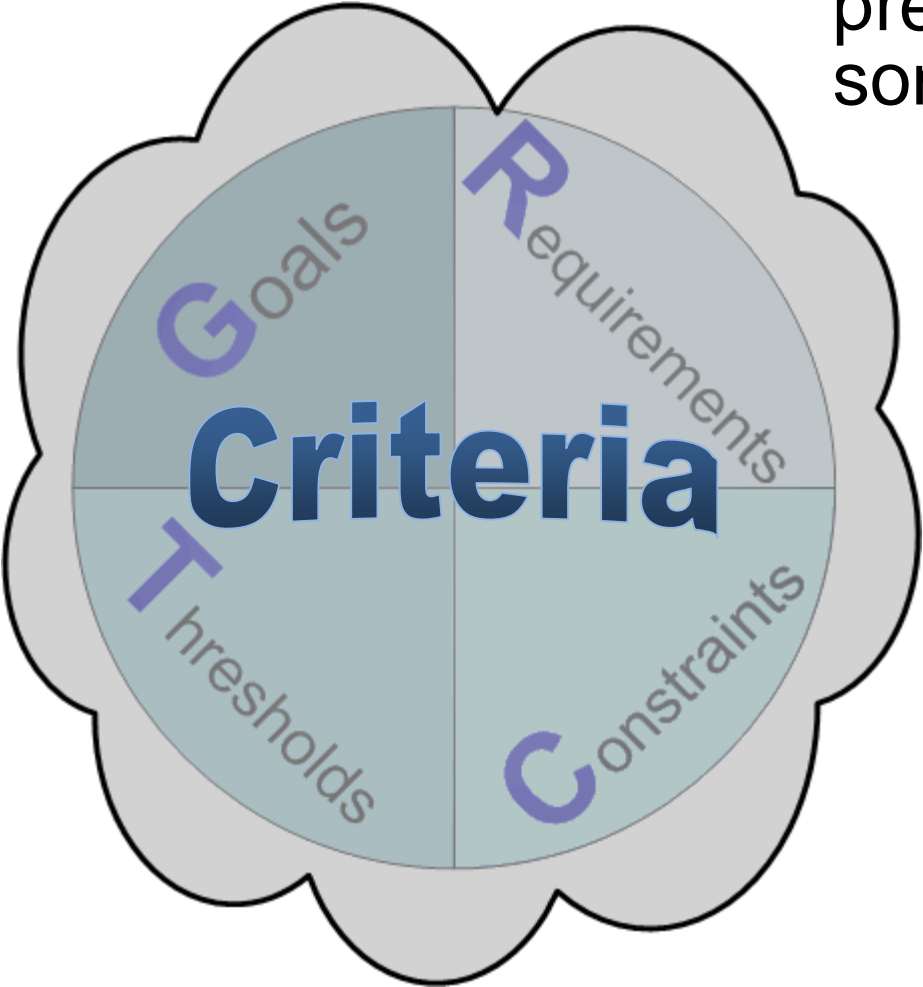
Criteria





Criteria

Performance Criteria are *boundaries* dictated or presumed by someone or something that matters.



Goals: Soft Boundaries
(User Satisfaction)

Requirements: Firm Boundaries
(Business or Legal)

Thresholds: Hard Boundaries
(Laws of Physics)

Constraints: Arbitrary Boundaries
(Budget or Timeline)





Criteria

Context

+

Criteria

=>

Performance Testing Objectives





Performance Testing Objectives

What we actually hope to gain by testing performance

Are sometimes completely unrelated to stated requirements, goals, thresholds, or constraints

Should be the main drivers behind performance test design and planning

Usually indicate the performance-related priorities of project stakeholders

Will frequently override goals in “go-live” decisions

How do we know if we're meeting our objectives?





Design





Design

Sorry, this slide is busy now. Try a different course by Scott Barber.

For now, remember to design tests that are provide information about your objectives.

--SlideMaster





Instrument





Instrument

Sorry, this slide is busy now. Come back for a different course by Scott Barber.

For now, remember to instrument your environment to collect information about your objectives.

--SlideMaster





Script





“MacGyver is a super-hero,

not

a career path.”

--Scott Barber





Script

Yes, this slide is busy as well. It is also available in other courses by Scott Barber.

For now, remember that for your data to be useful, the scripts must accurately represent your design.

--SlideMaster





Execute





Execute

You guessed it, this slide is busy too. It is having a party with its' friends, but is happy to display itself during other courses by Scott Barber.

For now, remember to validate that your tests did what you intended and that you execute enough times to have confidence in your data.

--SlideMaster





Analyze





*“With an order of magnitude fewer variables
performance testing could be a science,
but for now,*

*performance testing is at best
a scientific art.”*

--Scott Barber





Analyze

	Sample Size	Minimum	Maximum	Average	Median	Normal	Mode	95th Percentile	Standard Deviation
Data Set A	100	1	7	4	4	4	4	6	1.5
Data Set B	100	1	16	4	1	3	1	16	6.0
Data Set C	100	0	8	4	4	1	3	8	2.6

All three have an average of 4.

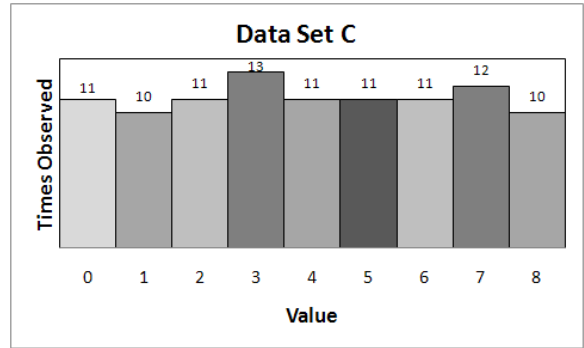
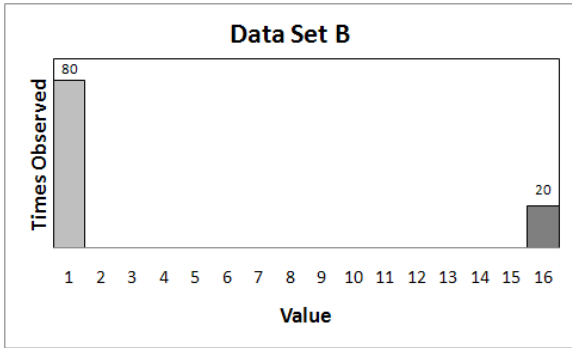
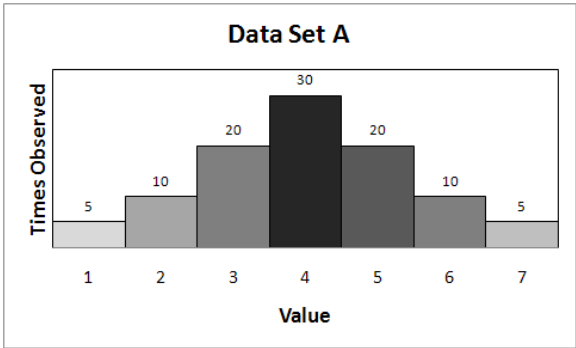
Which has the “best” performance”?

How do you know?





Analyze



	Sample Size	Minimum	Maximum	Average	Median	Normal	Mode	95th Percentile	Standard Deviation
Data Set A	100	1	7	4	4	4	4	6	1.5
Data Set B	100	1	16	4	1	3	1	16	6.0
Data Set C	100	0	8	4	4	1	3	8	2.6

All three have an average of 4.

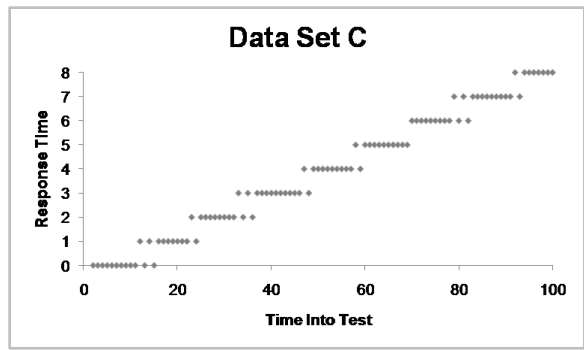
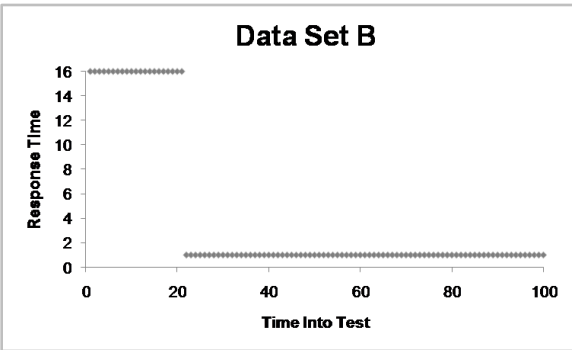
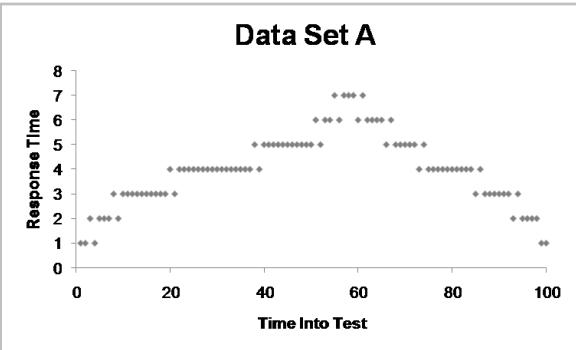
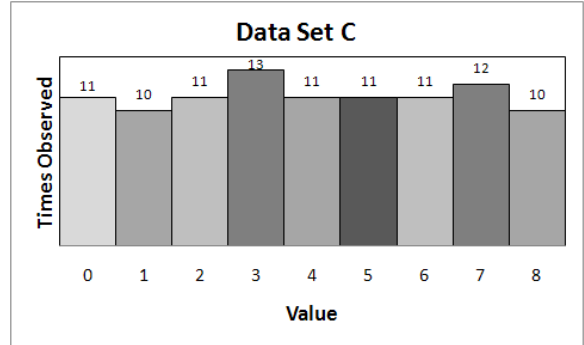
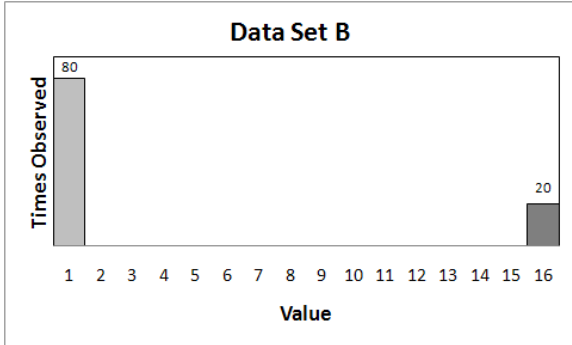
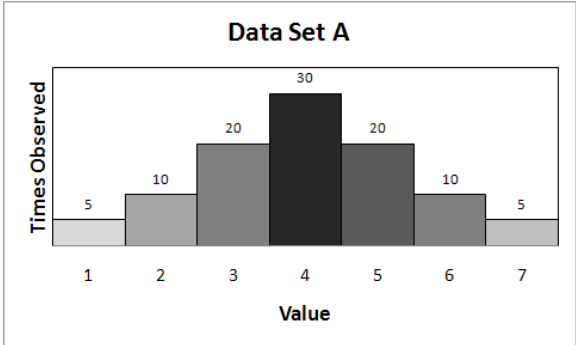
Which has the “best” performance”?

How do you know?





Analyze



	Sample Size	Minimum	Maximum	Average	Median	Normal	Mode	95th Percentile	Standard Deviation
Data Set A	100	1	7	4	4	4	4	6	1.5
Data Set B	100	1	16	4	1	3	1	16	6.0
Data Set C	100	0	8	4	4	1	3	8	2.6

Now which has the “best” performance”?





Analyze

Configurations

Results are meaningless without technical context.

Significance & Repeatability

Don't over-trust results until you can repeat them.

Trends

Within the test run, across tests, across data, etc.

Outliers

If you can repeat it or it's >1%, it's not an outlier.

Patterns

Graph, blink, overlay, compare, and contrast.

Compliance

If it can get you sued, check it every time.

Accuracy

How well do the results represent reality.

Resources & Times

This is where users care and symptoms are found.

Errors & Functionality

If it's broken, performance doesn't matter.





Analyze

Methods:

- ✍ Blink
- ✍ De-Focus & Re-Focus
- ✍ Overlay
- ✍ Plot
- ✍ Bucket
- ✍ Look for Odd
- ✍ Be Derivative
- ✍ Ditch the Digits
- ✍ Un-average Averages
- ✍ Manual





Analyze

Facts:

- ✦ Analysis is a team sport.
- ✦ We cannot **prove** anything.
- ✦ Focus on patterns, trends, and feelings.
- ✦ Numbers are meaningless out of context.
- ✦ Qualitative feedback is **at least** as relevant as quantitative feedback.





Analyze

Instructions:

Reassemble into your group.

Pay attention, I'm going to explain this group of exercises orally.

Be prepared to describe your findings with the class.





Analyze

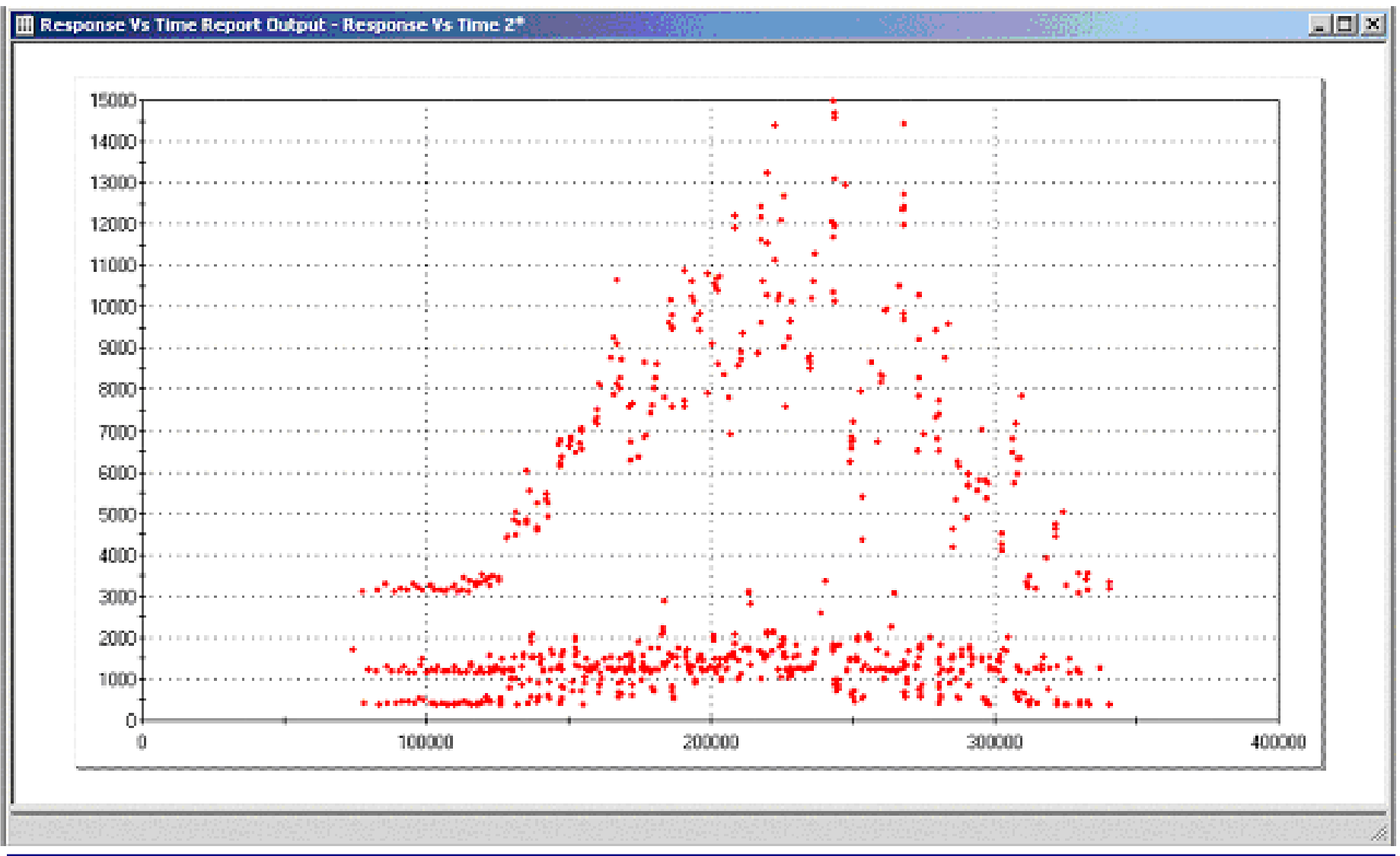
Let's play a game that I call...

**If Dr. House was a
Performance Tester**



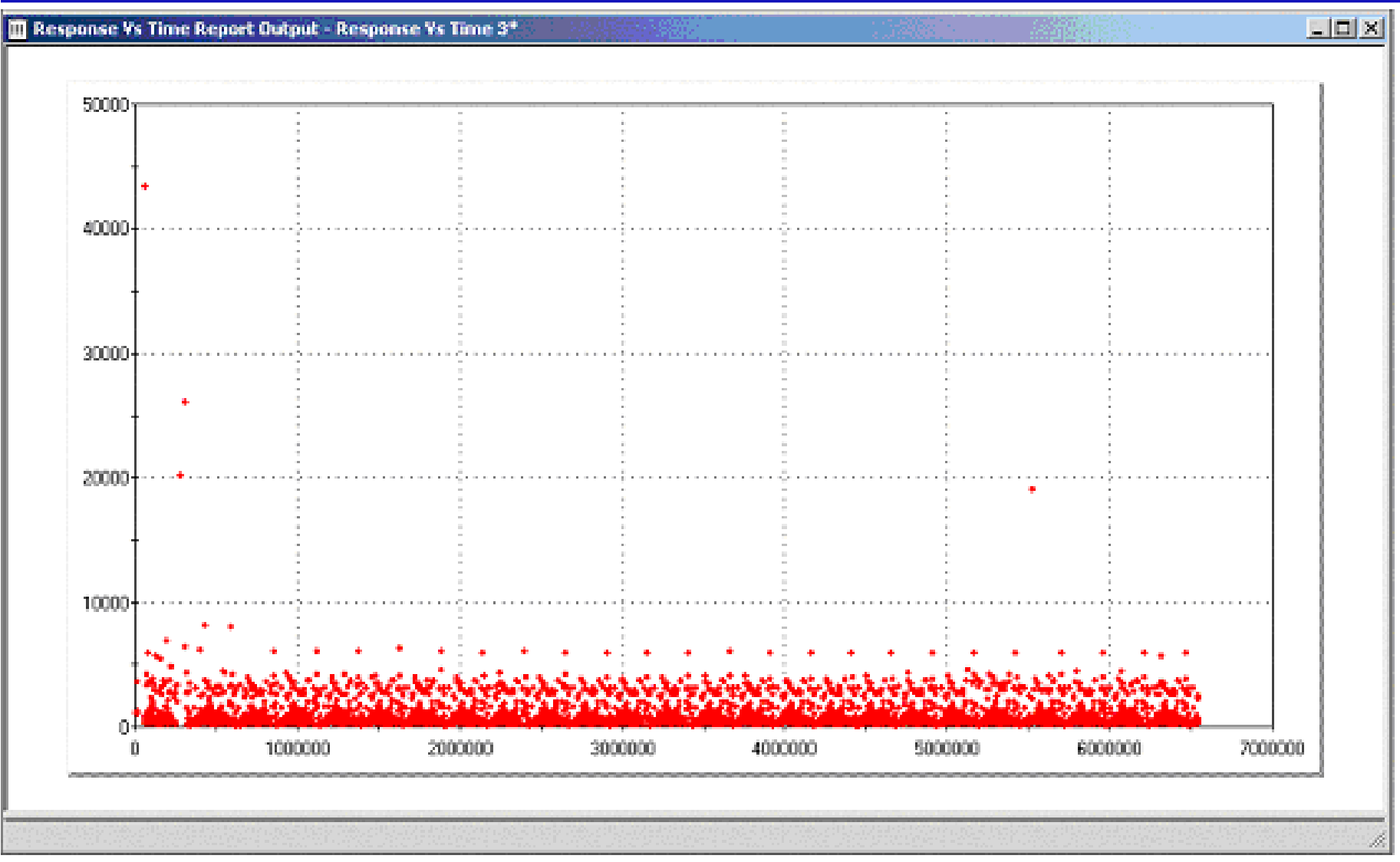


Analyze



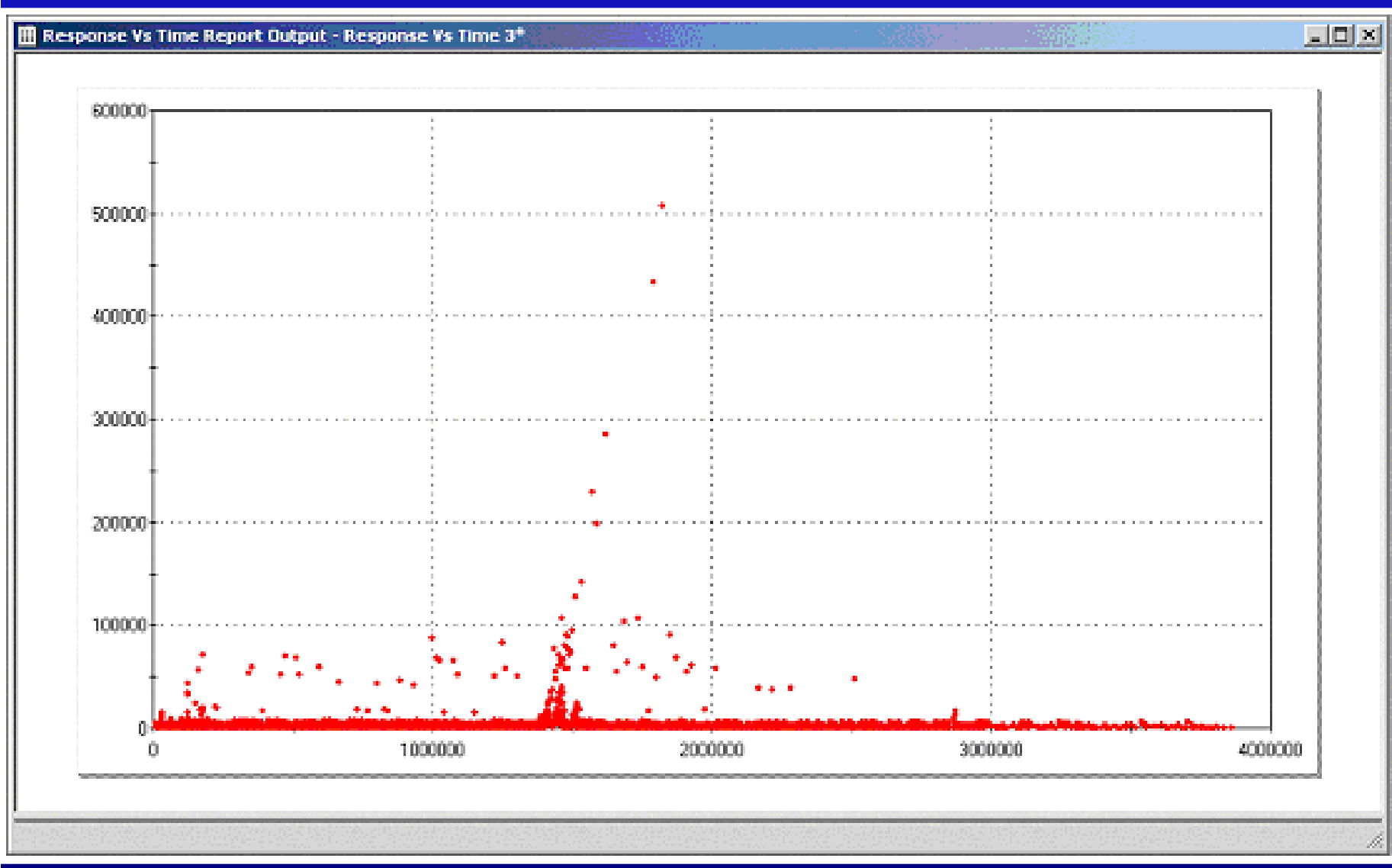


Analyze



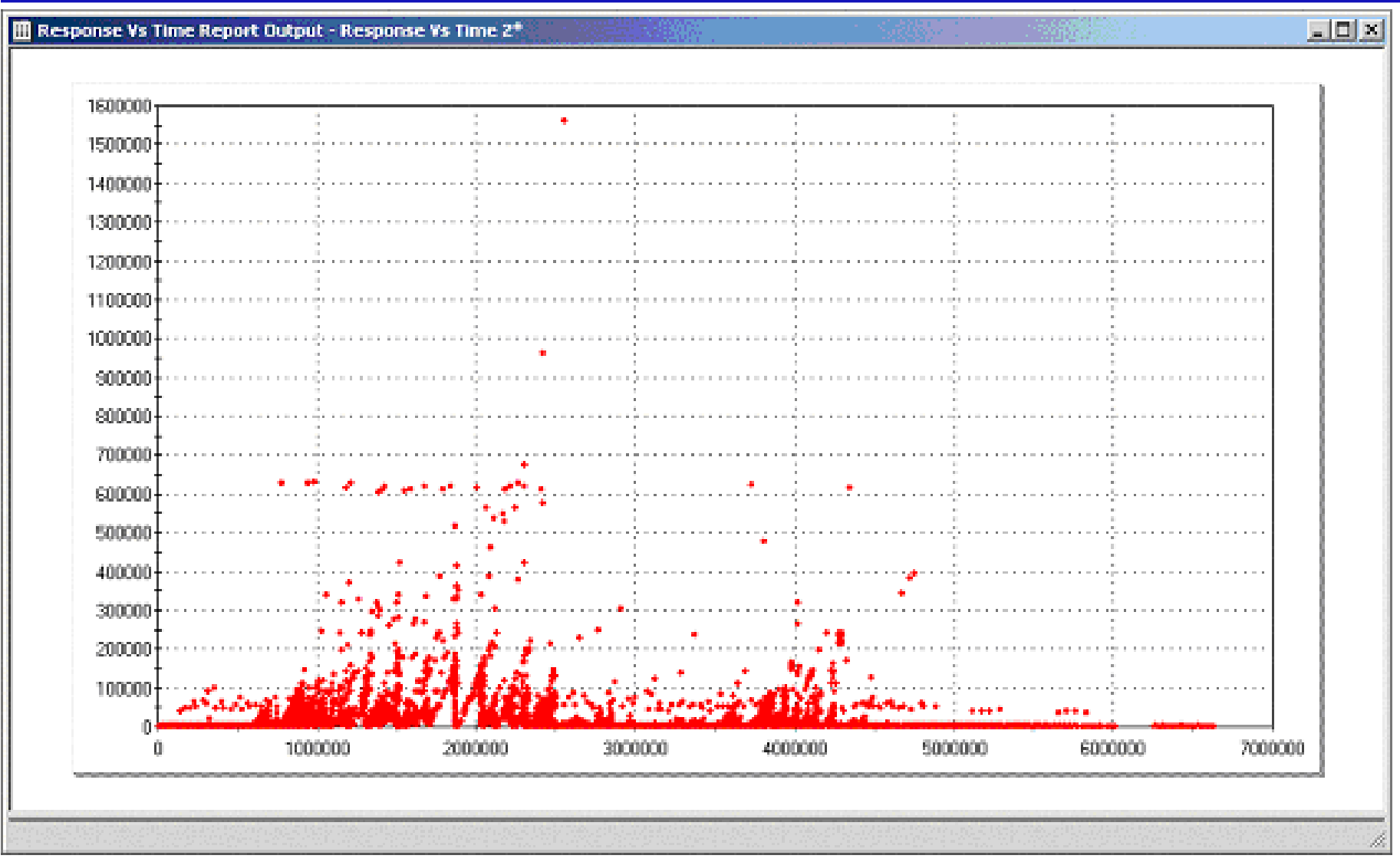


Analyze



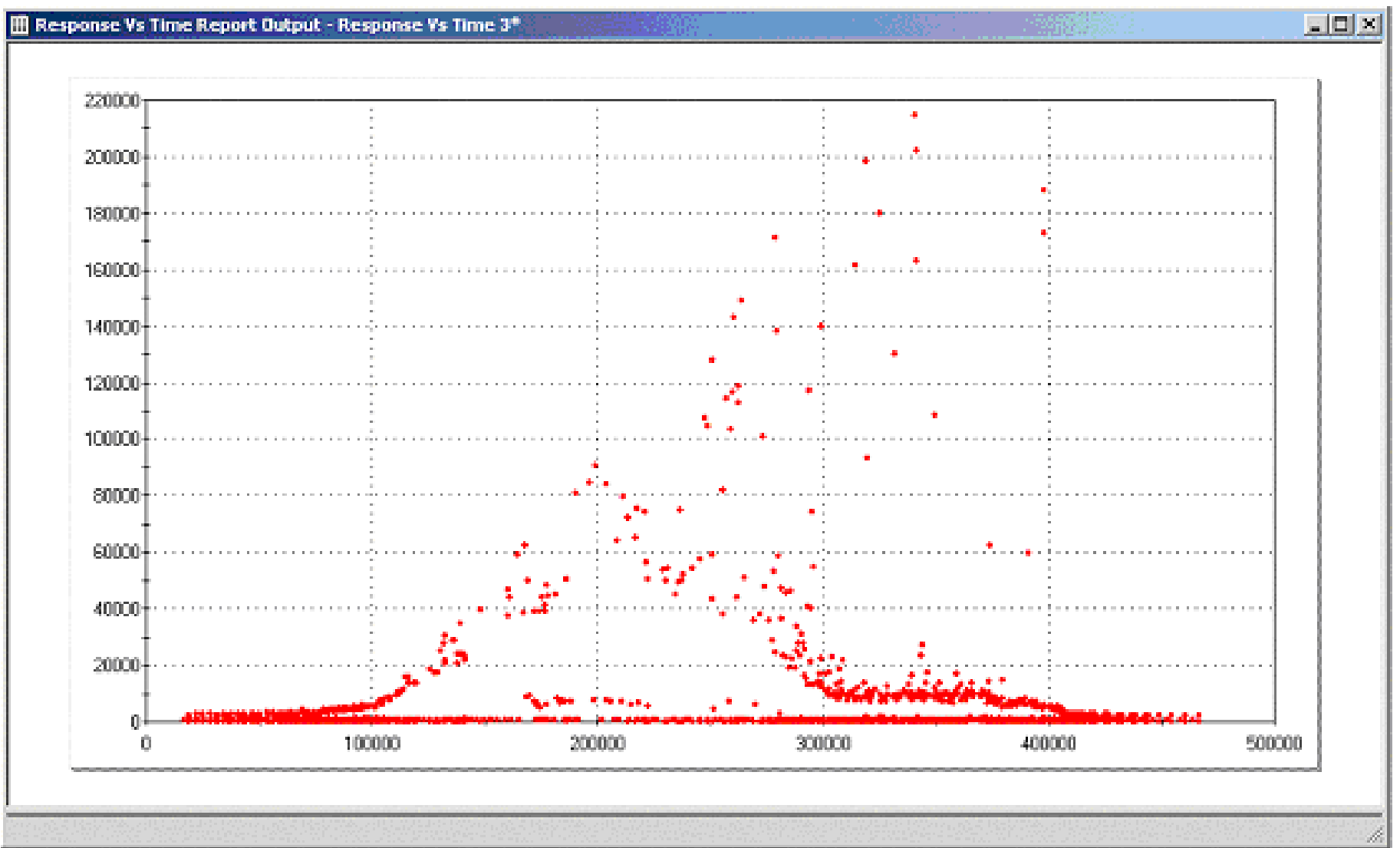


Analyze



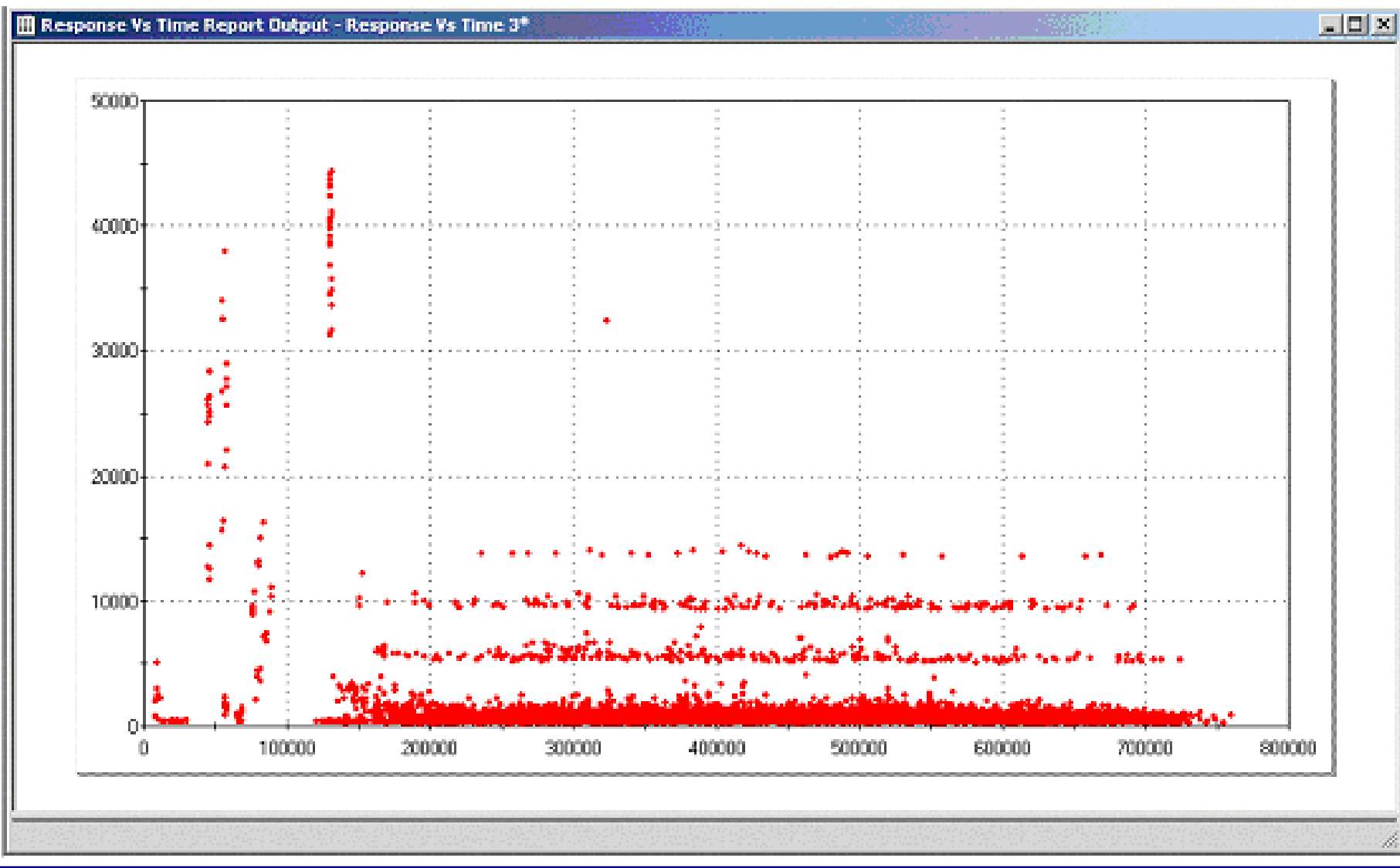


Analyze



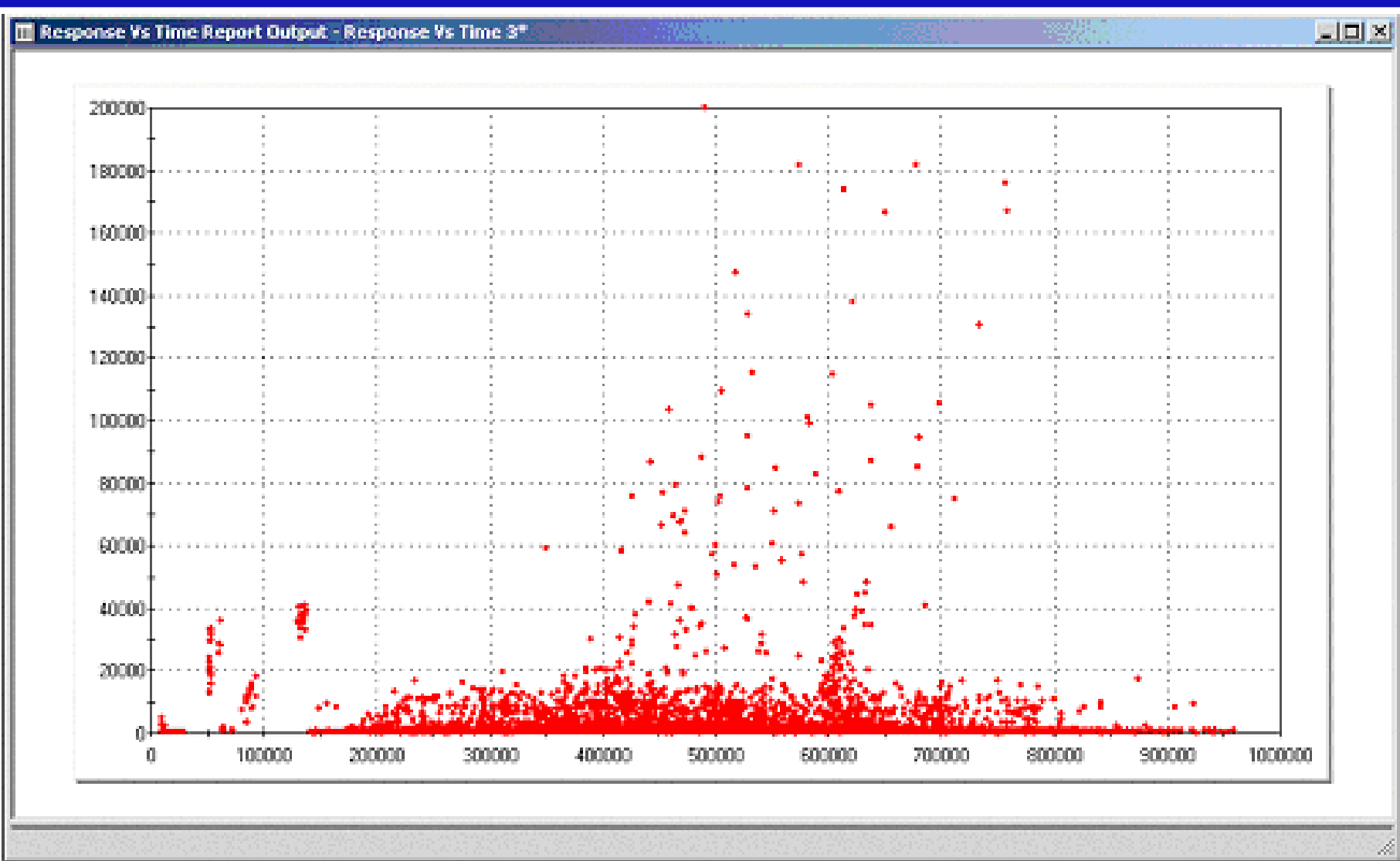


Analyze



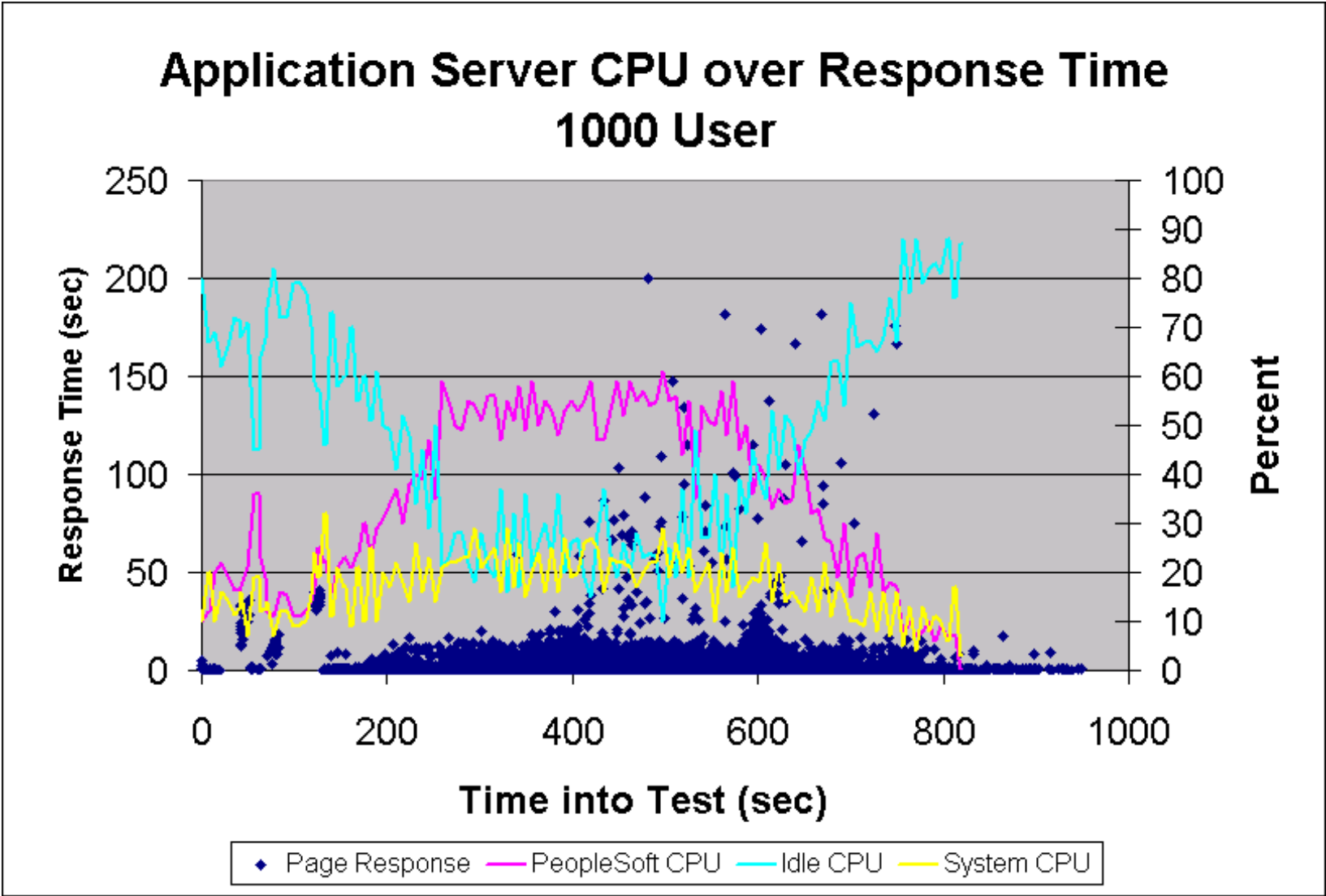


Analyze



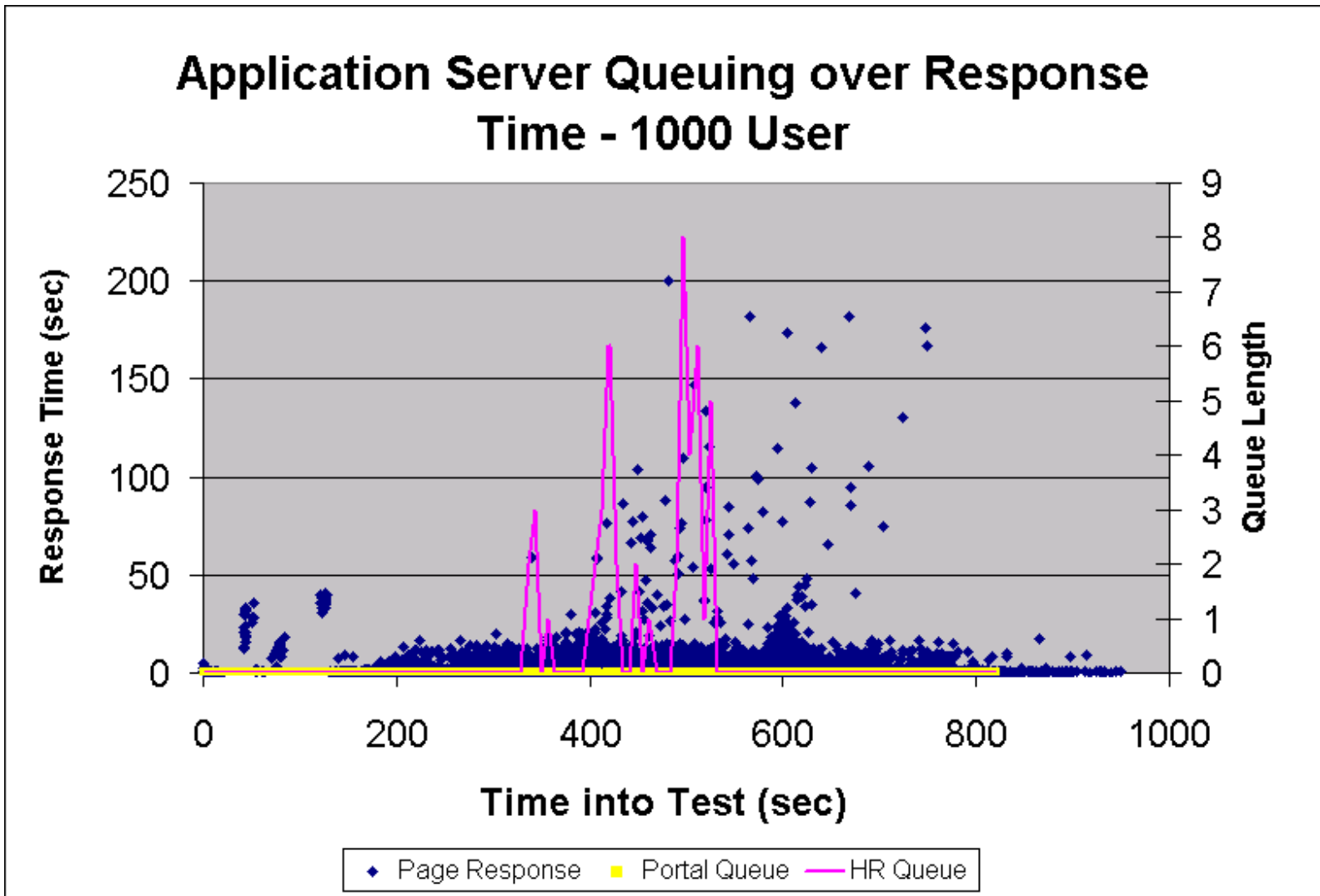


Analyze





Analyze





Report





*“Linear extrapolation
of performance test results is,
at best, black magic.*”

*Don’t do it (unless your name is Connie Smith, PhD.
or Daniel Menasce, PhD.)”*

--Scott Barber





Report

Timely

Stakeholders need data to make decisions. Many decisions can't wait until tomorrow.

Relevant

Reports are only interesting if they contain data that is useful.

Audience Appropriate

A great report for developers is probably a lousy report for executives.

Visual

Try to use pictures over numbers and numbers over words. Save words for recommendations.

Intuitive

Strive to make reports compelling without explanation.

Supported

Unless you are hiding something, make the supporting data available to the team.





Report

Facts:

- Most people will never read performance test results docs.
- Most people don't really understand the underlying components to performance.
- It is our job to make it easy for them to understand, and understand quickly.
- Being skilled at graphical presentation of technical information is critical for us to help others understand the message we are delivering.
- Confusing charts and tables lead to wrong decisions causing lost \$ and ruined reputations.





Report

What consumers of reports want:

- ✦ Answers... **NOW!** (They might not even know the question)
- ✦ To understand information intuitively.
- ✦ Simple explanations of technical information.
- ✦ To be able to make decisions quickly and have the information to support those decisions.
- ✦ “Trigger phrases” to use with others.
- ✦ Concise summaries and conclusions.
- ✦ Recommendations and options.

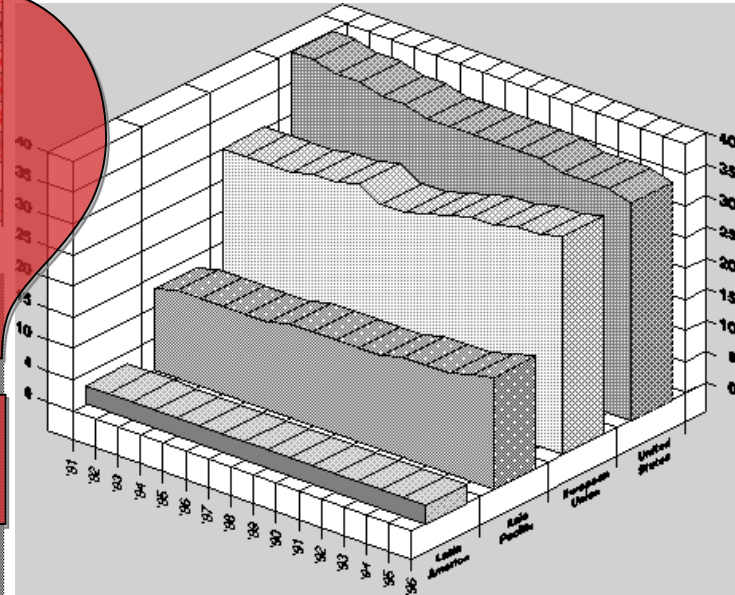
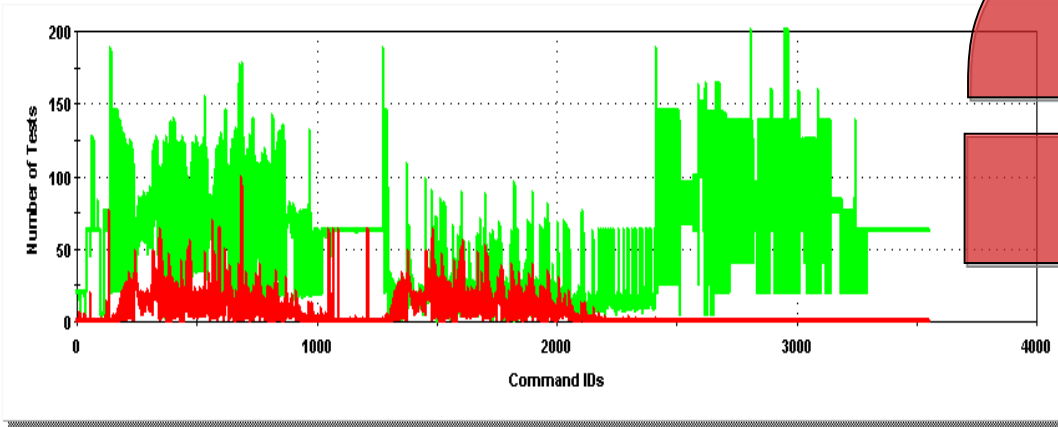
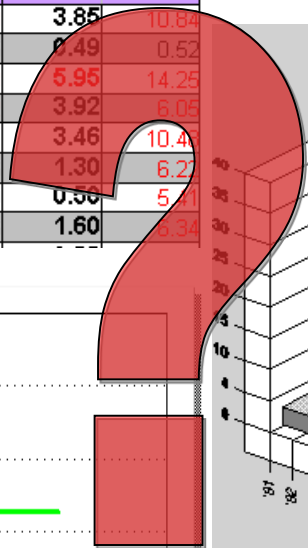




Report

What consumers of reports usually get:

Timer Name	Baseline		250		500		750	
	95th	Max	95th	Max	95th	Max	95th	Max
ec_Main_Page	10.46	18.09	6.41	8.22	6.33	8.33	3.85	10.84
ec_logon_help	0.98	0.98	0.56	0.59	0.55	0.55	0.49	0.52
ec_login	5.35	7.92	6.66	11.84	6.75	17.03	5.95	14.25
quick_learns	6.66	6.67	5.91	10.98	5.92	11.02	3.92	6.05
view_quick_learn	15.66	17.11	5.53	10.72	3.89	10.61	3.46	10.4
view_faq_window	2.45	2.45	1.47	1.52	1.53	1.66	1.30	6.2
view_faq	0.67	0.67	0.60	0.63	0.58	0.69	0.56	5.7
view_ec_status	8.08	12.55	1.73	6.66	1.80	1.86	1.60	6.34





Report

Strive for something better:

- ✍ Concise verbal descriptions.
- ✍ Well formed, informative charts (pretty pictures).
- ✍ Focus on requirements and business issues.
- ✍ Don't be afraid to make recommendations or draw conclusions!
- ✍ Make all supporting data available to everyone, all the time (Don't sit on data 'cause they won't understand it).
- ✍ Report \neq Document
- ✍ Report ***AT LEAST*** every 48 hours during execution.





Report

Inspired by “ET”:

Edward Tufte, Ph.D., Professor Emeritus of political science, computer science and statistics, and graphic design at Yale.

According to ET:

Power Corrupts...





PowerPoint Corrupts Absolutely.





Report

Appendix Level

	A	B	C	D	E	F	G
1	Filename	Input_Time	Output_Time	Elapsed_Time	Region	Output_Time_Bucket	5min_bucket
2	A000251_Tx	28890	109390	80500	A	0	0
3	C000245_Tx	67660	168280	100620	C	0	0
4	H000237_Tx	210530	293110	82580	H	0	0
5	P000242_Tx	358610	418880	60270	P	0	1
6	W000249_Tx	498560	693870	195310	W	1	2
7	A000252_Tx	635260	714870	79610	A	1	2
8	C000246_Tx	669440	781560	112120	C	1	2
9	H000238_Tx	922930	1009260	86330	H	1	3
10	P000243_Tx	961990	1026050	64060	P	1	3
11	W000250_Tx	1103720	1183920	80200	W	1	3
12	A000253_Tx	1241300	1317870	76570	A	2	4
13	C000247_Tx	1282800	1387090	104290	C	2	4





I terate





Iterate

Guess where I am...

For now, remember that you can never have too much data, and that you'll rarely have enough time to analyze the data you do have.

--SlideMaster





Performance Testing Principles

Context

Project context is central to successful performance testing.

Criteria

Business, project, system, & user success criteria.

Design

Identify system usage, and key metrics; plan and design tests.

Install

Install and prepare environment, tools, & resource monitors.

Script

Script the performance tests as designed.

Execute

Run and monitor tests. Validate tests, test data, and results.

Analyze

Analyze the data individually and as a cross-functional team.

Report

Consolidate and share results, customized by audience.

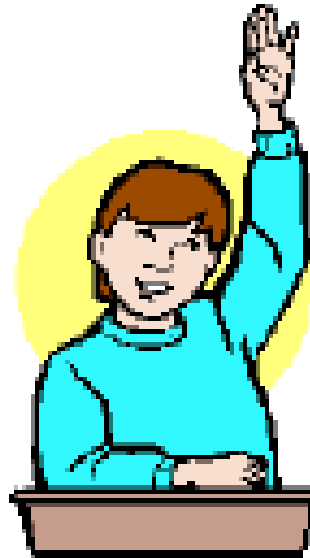
Iterate

"Lather, rinse, repeat" as necessary.





Questions





Contact Info

Scott Barber
Chief Technologist
PerfTestPlus, Inc

E-mail:

sbarber@perftestplus.com

Web Site:

www.PerfTestPlus.com

