Creating Effective Load Models for Performance Testing with Incomplete Empirical Data

Updated for the:

3rd World Congress for Software Quality

September, 2005 Munich, Germany

Initial Research Presented for the:

6th IEEE International Workshop on Web Site Evolution

September, 2004 Chicago, IL

Scott Barber

Chief Technology Officer

PerfTestPlus, Inc.

Common Modeling Techniques…

Tend to fall into one of two categories:

Rigorous

- Time consuming
- Mathematically intensive and/or complex
- High degree of accuracy (when done well)
- Requires empirical data

Overly Simplistic

- Quick
- Little to no math needed
- Occasionally accurate (generally by accident)
- Ignores empirical data

There is very little in between to assist the modeler in industry that desires rigor, but barely has time for simplistic!

Rigorous Techniques

Connie U. Smith, PhD. - *Performance Solutions: A Practical Guide to Creating Responsible, Scalable Software* [1]

Alberto Savoia - "Web Load Test Planning: Predicting how your Web site will respond to stress" [2]

Daniel Menasce, PhD. – *Capacity Planning for Web Performance: Metrics, Models and Methods* [3] & *Scaling for E-Business* [4]

J.D. Meier - *Improving .NET Application Performance and Scalability* [5]

**All Require Empirical Data

Performance Solutions…

"Software Performance Engineering (SPE) Approach" to Building Usage Models:

"SPE is a comprehensive way of managing performance that includes principles for creating responsive software, performance patterns and antipatterns for performance-oriented design, techniques for eliciting performance objectives, techniques for gathering the data needed for evaluation, and guidelines for the types of evaluation to be performed at each stage of the development process.

"SPE is model-based… By building and analyzing models of the proposed software, we can explore its characteristics to determine if it will meet its requirements before we actually commit to building it". [6]

Performance Solutions…

"Performance testing is vital to confirm that the software's performance actually meets its performance objective, and that the models are representative of the software's behavior". [7]

To elaborate on Smith's statement:

- Performance testing is vital because models are not perfect.
- Validation of the model's predictions via testing ensures the model matches the implementation.
- SPE model predictions are only validated accurate load models.
- Chapter 3 details how to model application usage in terms of performance with UML, *BUT*
- There is no discussion about how to determine or estimate actual usage of the application under test.

Conclusion: With no empirical (i.e. actual usage) data, SPE models cannot be validated.

"Web Load Test Planning…"

"Website Usage Signature (WUS) Approach" to Building Usage Models:

"In order to be worth anything, I believe that Web site load tests should reproduce the anticipated loads as accurately and realistically as possible. In order to do that you will need to study previous load patterns and design test scenarios that closely recreate them... ". [8]

Savoia expands, saying:

- The preferred method of determining actual load and load patterns is accomplished by analyzing log files from the existing version of the application.
- In the absence of relevant log files…[generate] log files via use of a limited beta release of the application to a representative sampling of users

Conclusion: With no empirical (i.e. actual usage) data from previous versions or limited releases, WUS is not applicable.

Scaling… & *Capacity Planning…*

"Capacity Planning Approach" to Building Usage Models:

In *Scaling…*[9] Menascé advocates:

- WUS-like usage models
- SPE-like architectural models
- Forecasting mathematically from these

In *Capacity Planning…*[10] Menascé discusses:

- Methods for generating models for performance testing and capacity planning when actual usage data isn't available.
- Though these methods are based in mathematical concepts that few practicing performance testers are skilled in.

Conclusion: With no empirical (i.e. actual usage) data the methods in *Scaling…* are not applicable and the mathematics involved with the methods in *Capacity Planning…* make them unusable in much of industry.

Improving .NET…

"End-to-End Approach" to Building Usage Models:

"You can determine patterns and call frequency by using either the predicted usage of your application based on market analysis, or if your application is already in production, by analyzing server log files. Some important questions to help you determine the workload for your application include:

- *What are your key scenarios?...*
- *What is the possible set of actions that a user can perform?...*
- *…*
- *What is the duration for which the test needs to be executed?..."* [11]

Conclusion: Each question is explained and has examples, but how to collect the data is not addressed, making the method unreliable without empirical data.

State-of-the-Practice

In Practice:

- Empirical Data is Uncommon
- Complex Math Skills are Rare
- Time is not a Luxury we have
- The Highest Volume is Often on “Go-Live Day”
- Few Modeling Tools and Methods are Easily Available
- Most Usage Models are little more than “Semi-Educated” Guesses

State-of-the-Practice

"State-of-the-Practice Approaches" to Building Usage Models:

Common Approaches in Industry:

- Evaluation of use cases and design documents.
- Information from a key stakeholder (i.e. an interview).
- Observe user acceptance and/or usability testing.
- Evaluate similar sites and generic web usage data.
- Personal use and experience.

Which lead to:

- Low confidence in predicting performance under actual usage.
- High likelihood of undetected performance issues in production.
- Little or no validation of the load testing model.
- A propensity for accepting load generation tool limitations.

Hybrid Approach

Based on the experiences of 13 expert consultants… [12], [13]

- Start with available production data from existing versions of the app.
- Where there is no empirical usage data, start with interviews and documentation.
- Collect data from a limited beta roll-out whenever possible.
- Compare/supplement data with generic or competitive data.
- Develop and distribute for review one or more draft load models.
- Run simple in house usage experiments/surveys.
- Enhance the model based on experiments and common sense.
- Create best, expected and worst case models based on the distribution of usage data collected.
- Develop programs and or scripts to implement the model(s).

Hybrid Approach

Roughly a dozen "blind" field tests suggest that… [14],[15],[16],[17]

- The SPE, Capacity Planning, WUS, End-to-End and Hybrid approaches:
  - Are statistically equivalent when followed precisely.
  - Make predictions with acceptable and explainable degrees of error.
  - Are relatively easy to enhance to accurately predict production.
- State-of-the-Practice approaches:
  - Do work sometimes, but often yield critically misleading predictions.
  - Are not enhanced or reused after the initial predictions are made.
  - As often as not, lead to poor business decisions.
- WUS, End-to-End and Hybrid approaches:
  - Are roughly equivalent in terms of time and difficulty to conduct.
  - Result in nearly identical predictions for applications with feature additions.
  - Do not require significant training to implement.
- Business Users, Managers, Testers and Developers preferred the pictorial documentation of the Hybrid approach more than 10 to 1 over documentation methods recommended by the other approaches.

Summary

SPE, Capacity Planning, WUS, End-to-End and Hybrid approaches to developing usage models typically yield "good enough" predictions.

State-of-the-Practice approaches to developing usage models typically yield "unreliable" predictions.

SPE and Capacity Planning approaches to developing usage models are rarely applicable in industry.

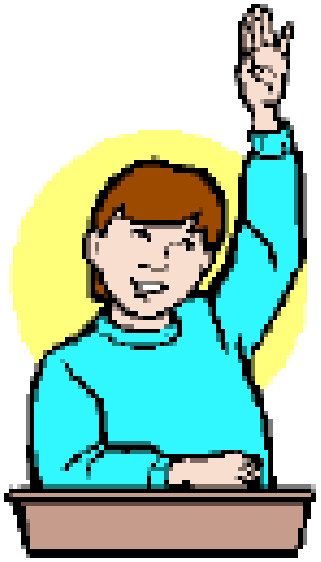
WUS, End-to-End and Hybrid approaches to developing usage models are generally viable typically for industry use.

The Hybrid approach is an acceptable and more accurate alternative to State-of-the-Practice approaches to developing usage models when empirical data is not available.

References

1. Smith, Connie U.; Williams, Lloyd G. (2002), Performance Solutions: A Practical Guide to Creating Responsible, Scalable Software, Pearson Education, Inc.
2. Savoia, Alberto (2001) “Web Load Test Planning: Predicting how your Web site will respond to stress”, STQE Magazine.
3. Menascé, Daniel A.; Almeida, Virgilio A.F. (1998), Capacity Planning for Web Performance: Metrics, Models and Methods, Prentice Hall PTR.
4. Menascé, Daniel A.; Almeida, Virgilio A.F. (2000), Scaling for E-Business: Technologies, Models, Performance and Capacity Planning, Prentice Hall PTR.
5. Meier, J.D.; Vasireddy, Srinath; Babbar, Ashish; Mackman, Alex (2004) “Improving .NET Application Performance and Scalability”, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag/html/scalenet.asp, last accessed June 7, 2004
6. Smith, 2002
7. Smith, 2002
8. Savoia, 2001
9. Menascé, 1988
10. Menascé, 2000
11. Meier, 2004
12. Barber, Scott (2002), “User Experience, not Metrics”, Rational Developer Network
13. Barber, Scott (2003), “Beyond Performance Testing”, Rational Developer Network
14. Tani, Steve, “In search of Data/Case studies #2”, 6/27/2004, via e-mail
15. White, Nathan, “Tool for Overall Workload Distribution Figure”, 9/19/2002, via e-mail
16. Rodgers, Nigel, “UCML”, 6/16/2004, via e-mail
17. Warren, Brian, Ceridian, “UCML 1.1 Visio stencils”, 6/9/2004, via e-mail

Questions



Contact Information

Scott Barber

Chief Technology Officer

PerfTestPlus, Inc

E-mail:

sbarber@perftestplus.com

Web Site:

www.PerfTestPlus.com