STeP-IN SUMMIT 2008

5th International Conference On Software Testing

Performance Assurance – The need for Predictability

by

Hemalatha Murugesan
Head Enterprise Performance Testing Services

INFOSYS TECHNOLOGIES LTD

Practices Experience Knowledge Automation

Produced By

STeP-IN Forum

www.stepinforum.org

Hosted By

QSIT The Knowledge Corporation

www.qsitglobal.com

Introduction

In today's rush to deploy applications, it is critical that applications perform to the diversified needs of the end users at acceptable response times. However, most of the applications developed are not budgeted enough with time, effort and above all the necessary skilled people to implement best practices and approaches. Performance testing is recognized as one of the critical component of the software assurance process. In today's time to market, many of the required processes, approaches are not rigorously applied or scientifically measured leading to performance problems in the production environment. Currently assuring performance is more reactive than being proactive in most of the software application development. Being reactive is not only very expensive to fix the defects; it also affects future enhancements or increase performance assurance leading to lot of rework and re-engineering on the design.

Diversification, broadening customer base, aggressive competition, and increasingly integrated world are leading to wide scale up-gradation of legacy applications and other platform flexibility related requirements from Information Technology. While rapid / flexible deployment and easy maintainability are critical for organizations; ensuring high quality performance levels of these applications within the expected turnaround time is the key differentiator to end user experience. Users access the systems in different time zones, from various geographical locations, through diverse IT infrastructure backbones, over different bandwidths and speeds through different types of computer systems as well. The need to assure Business users and Senior management that the migrated or newly deployed applications is performance intensive has gained a significant importance that is being monitored periodically.

Current State

In the traditional approach, performance is evaluated by conducting performance tests on the software once it is built. The evaluation is typically done towards later stages of the Software Development Life Cycle (SDLC). The tests and its analysis report merely remains on paper as implementation of the changes will result in massive rework with a lot of effort, resources and time. One of the key challenges in performance testing lies in the difference between production and test environments. Many testing teams find an application's performance issues in production difficult to understand because of differences in hardware and architecture. An application may meet business requirements in the test environment, but may not meet those expectations in

production. Added to this is the complexity of interfacing with a mammoth suite of various legacy applications, acquired applications, ERP products – SAP, etc and with many other complex systems leading to severe performance issues and at times, inability to identify where the problem is. More applications built from discrete, integrated components mean more opportunities for performance problems within each component and at each integration point.

Moreover, in any Enterprise Organization, there are multiple applications and shared services forming a highly complex heterogeneous mesh of support systems, each vying for its own share of common resources, such as CPU utilization, network bandwidth, database, LDAP, DNS servers, load balancing, storage etc. to name a few. In such a complex technological environment with diverse functional needs, performance testing even a single application is highly complicated as it is not just about applying load to the system and just monitoring it. Over 95% of developed applications suffer from medium to catastrophic performance issues during testing leading to at times, even halting the entire application deployment phase. As such, to consistently win in the market place, businesses must strive to have Information Technology (IT), infrastructure that is Best-In-Class, both in Performance & Availability.

Need for Predictability

The need for Predictability of performance is becoming more important to assist the development team in identifying and mitigating the risks much ahead in the software development life cycle than post implementation. There is no systematic way of validating the performance requirements right at the architecture or design stage. Most of the Traditional empirical methods fail to encompass complexities of open and networked systems. Predictability in Performance is required not just on application alone but on a combination of software's and systems as well as on the infrastructure. By deploying appropriate Predictive Models and accurately identifying parameters for analysis, it is possible to mitigate the performance issues and risks much earlier in the cycle, thereby significantly providing value to the entire software development process.

A more recent approach to Software Performance Evaluation is through Performance Modelling Techniques. Performance models are developed and evaluated to indicate performance characteristics of software and hardware systems. Predictive Performance Modelling Solution helps in identifying the problems much earlier in the cycle and predicts the behaviour of the application as in Production.

Having a replica of production environment for performance testing may not be a feasible option, providing high levels of predictability through scientific prediction models help bring in higher confidence on the services. In each phase, performance related data is estimated for models and the level of accuracy of these data estimates increased by evaluating it at interim phases. Infosys' internal studies and client experience indicate that this model usage led to an increase in level of accuracy to over 95% giving a more comfort feel on the deployment of the applications. Return on Investment (ROI) was recorded at over 40% in each stage of the testing cycle.

Predictive models adopt discreet event simulation techniques along with layered queuing theories to model and simulate a Multi-layered distributed enterprise application. It adopts a continuous performance evaluation process through out the SDLC, right from the architecture stage, Proof of Concept and Design to coding and testing phases.

Performance Predictive Model

The three key stages in this Performance Predictive Model are:

- **Workload Characterization / Modelling**
- **Profiling** and
- **Simulation**

This model has helped to predict performance in production with data from the test environment with reliable results. Analytical and simulation techniques provide means to evaluate these performance models. This model can be applied for both:

a) Application that is already deployed in Production:

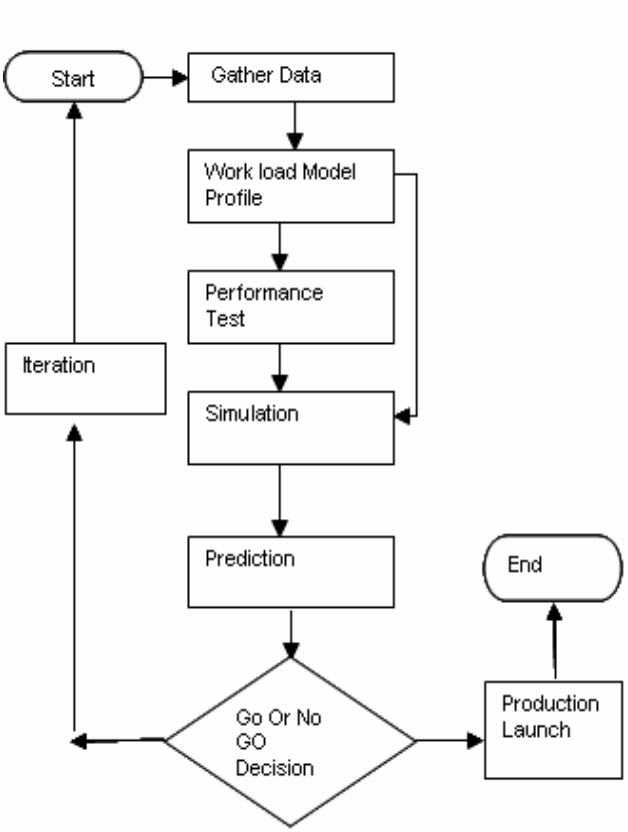
With a lot of legacy and existing applications being upgraded or migrated to web, this model can be implemented with much ease as Workload Pattern with associated logs and already existing performance issues is already available. Due to the data already being available along with the actual production deployed environment, there is an increase in predictability of the results.

b) New Application being developed:

The difference here is that actual transaction information along with deployable production environment is not available upfront leading to making some assumptions. There is a lot of dependency on the application and infrastructure/network team to

seek information to derive workload pattern. The process varies slightly when compared to already deployed application to generate the Workload model. Multiple rounds of meetings and brainstorming with various stakeholders is required so as to increase the predictability in performance.

Predictive Model Framework



Gather Data	•Collect Performance requirements & Web logs •Collect infrastructure details
Workload Model	•Analyze web logs and produce workload report. •Generate Transactional load profile for the application
Profile	•Aspectize the code •Conduct Profiling through multiple layers
Performance Test	•Design The performance test based on workload model and Profiling result •Execute the test and monitor hardware utilization
Simulation	•Simulate and generate simulation output.
Prediction	•Analyze result from simulator •Plot the result against actual load profile and predict the production performance

Fig 1.0 Predictive performance

The above diagram briefly summarizes the model approach. Each phase is important before proceeding to the next stage as the output of each phase is the input to the succeeding phase.

Phases in Predictive Model

i) Gathering Data

The first phase in the Predictive Modelling is to gather data to understand what is the architecture, production environment, etc. This Non-functional requirements are gathered which could be either for an already deployed application or to be developed system. Some of the attributes that needs to be considered while gathering the data but not limited to are:

- Roadmap of the business organization and their objectives
- Performance Goals
- Business growth and projected user base expansions
- CAPEX and OPEX
- Existing and prospective projected infrastructure
- Network environment, software and hardware along with required technical personnel to run the systems efficiently
- Service level agreements and likely competitive landscape

The data could be collated from documents such as the project charter, architecture details, application documents, deployment architecture documents, applications, web server, database server's information, web logs, usage trends logs, productions logs, and production support calls report in the case of already deployed systems. Meeting with key stakeholders is very essential to understand who the audiences are and what the SLA's expected are.

ii) Workload Characterization/Modelling

It is the process of identifying the way in which the application will be accessed over a period of time. The workload that a system is subjected too can vary from hour to hour, day to day, month to month, or season to season. Depending on the pattern of usage at a particular timeframe, there could be multiple Workloads for a system. All these Workloads have to be taken into account in order to do a realistic performance test of the system which closely resembles the production environment. The Application Usage Pattern and the Usage Volumes of each of these workloads need to be identified. Infosys has developed various tools to help develop the arrival patterns and estimation of loads in the systems. The Workload Characterization tool upon providing the relevant inputs identifies the critical transactions, finds out the intensity of the transactions and estimates the system workload (resource visits). The output provides info on Business Transactions with the corresponding throughput, number of concurrent users under each load distribution like Peak, Heavy, Medium and light loads.

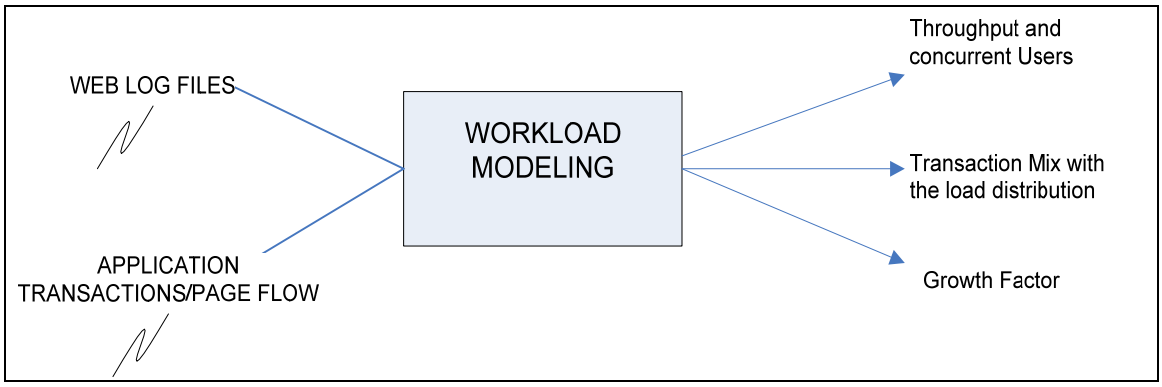


Fig 2.0 Workload Modelling – I/O

iii) Profiling

Profiling is a very important phase in Predictive Modelling for it provides info on Elapsed time and CPU times for the methods along with memory utilization. It captures info on SQL statements triggered, Memory leak detection, method call info and many more details. Aspectize the code to provide inputs on the CPU time of the methods, memory leaks, inefficient algorithms, inter-tier, inter-method etc. There are many commercial tools such JProbe, Optimizeit, JProfiler, Memory profiler, Heap analyzer, etc. available that can be used to profile the code. This stage helps addressing most of the code related performance issues at method level, component level as well as system level for a single user.

iv) Performance Test

To further enhance the accuracy of the data, performance (stress) testing can be done which also provides inputs on the capacity of the infrastructure as well on the network. This doesn't limit just to application under test (AUT) but also seeks inputs on various other systems being deployed in the same infrastructure and how this AUT will perform as well. The performance (stress) test needs to be executed for capturing the Application response time, throughput, utilization details etc. A proper analysis of the stress test needs to be done before feeding the data to the simulator. Stress testing should not be considered as a replacement for profiling. The stress test requirements are captured from the workload model results. The inputs include the mix of transactions, the total load to be ramped up on each of the transaction, the ramp up intervals, etc.

Computation of Service Demand: The Utilization law is used to calculate the service demand

U = X.S

U → Utilization
X → Throughput
S → Service Demand

v) Simulation

Simulation assists in Modelling, Sizing and Capacity Planning of a system including suggesting options for optimized system for highest ROI. Infosys has developed Simulation tools incorporating various performance related theories and laws. It follows a systematic approach to predict the performance of applications using Discrete Event Simulation with layered queuing theory. It provides an opportunity to evaluate the performance of business critical applications right at the architecture phase, even before the design and deployment, and continues to do so throughout the software development life cycle (SDLC) of your application.

The outputs of Workload Modelling, Profiling, and Performance test results as well as infrastructure, CPU utilizations etc. is fed into the Simulator and run. The simulation can be started once at-least 2 of the above data is fed into the simulator. The simulation process enables to perform the capacity planning for almost all platforms. Simulator actually simulates the Production environment by considering the application architecture, the arrival rate, number of transactions and the think time. The accuracy of output data depends on the accuracy of data that has been fed. A snapshot of sample template to provide inputs into the Simulation:

Total Number Of Concurrent Users	Test duration (min)	Response times (seconds)	Think time (seconds)	Throughput (Bytes/sec)	Web Server #1 CPU Utilization	Web Server #2 CPU Utilization	Application Server #1 CPU Utilization	Application Server #2 CPU Utilization	Database Server CPU Utilization

The outputs of the simulation are but not limited to are:

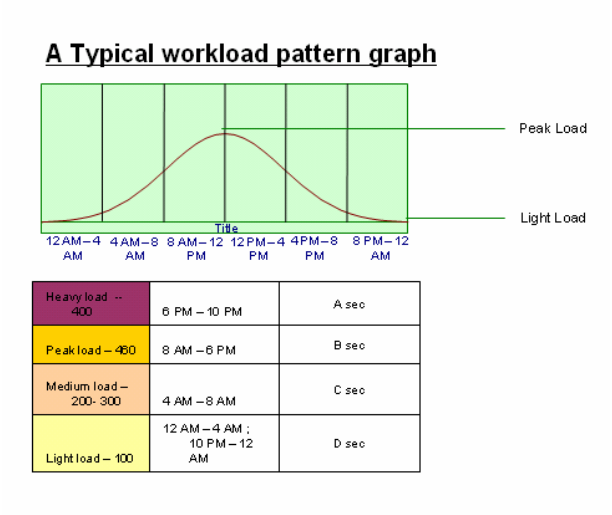
- Response Time
- Throughput
- Software and Hardware Options
- Deployment Configuration
- Cost of the System – by having the right architecture for the system
- Helps in Modeling, Sizing and Capacity Planning of a system, including suggesting options for optimized system for highest ROI

iii) Prediction

Production response time prediction is derived at by plotting simulator results against the 24 hr workload derived from workload modelling and taking the weighed average. The outputs of the Prediction are essentially dependent on how accurate the inputs have been made at various stages in Predictive Modelling phases. As the stage progresses in the SDLC, the accuracy increases providing strengthening evidences to development, infrastructure team as well as business users on the performance of the system and infrastructure. This also provides the response times against said architecture configurations and resource utilizations against various permutations and combinations.

From Simulator we would determine the response times at Peak load, Light load, Medium and Heavy Load.

Sample Outputs



Comparison of Response time Prediction with Production data				
User Load Category	**Users**	**Production Data** Response Time(sec)	**Predicted Data** Response Time(sec)	**Accuracy**
Medium Load (2693 - 5385 users)	5358	0.578	0.581	99.484%
Heavy Load (5386 - 8078 users)	6930	0.685	0.734	93.324%
Peak Load (8079 - 10770 users)	8576	0.599	NA	NA

Note: The response time at the peak load could not be predicted due to Infrastructure constraints.

The Predictive model finally derives a Dashboard report along various performance indicators – Enterprise Metrics, System Level Metrics, Application Level Metrics,

Network level metrics, etc. This model can be used when there are not enough data to run a test with large number of users or when there are not enough licenses to run a test with large number of users or when the test environment similar to production environment is not available.

The Predictive Model output gives a holistic view of the system's application performance providing the necessary Performance assurance not only to the stakeholders but to the senior management to make critical business decisions and investments. This model has helped to predict performance in production with data from the test environment with reliable results. Analytical and simulation techniques provide means to evaluate these performance models. In each phase, performance related data is estimated for models and the level of accuracy of these data estimates increased by evaluating it at interim phases. An upfront investment to implement Predictive Model has resulted in huge savings $$$ as well as defect fixes leading to deployment and maintenance of robust production deployment.

Infosys' internal studies and client experience indicate that this model usage led to an increase in level of accuracy to over 90% giving a more comfort feel on the deployment of the applications. Predictive models adopt discreet event simulation techniques to model and simulate a Multi-layered distributed enterprise application. It adopts a continuous performance evaluation process through out the SDLC, right from the architecture stage, Proof of Concept and Design to coding and testing phases. In one of the sample, in the Pre-production environment, on a 24 hr scale the median prediction response was at 5.96 sec using Predictive Modeling while the actual median reported in Production Environment was 5.97 sec. The Prediction accuracy achieved was 99.9% at much lesser cost and effort not to mention the time as well.

Conclusion

As more and more systems are made available and open to more users, the need to assure performance is going to very high on the radar of any senior management. Performance testing is all about predicting production impact of a new code drop. It is part of a process in which one establishes a base line, runs a test, identifies gaps and performs production impact on those gaps. The quality of the testing depends on the strength of the weakest link. Major performance problems in production can only be corrected by recreating the production scenario in a controlled environment and arrive at the solution through proper modelling, performance testing and analysis. The key to this approach lies in accurately identifying parameters that can help recreate the production workload for performance testing. Predictive Modelling helps capturing

serious design and scalability flaws as early as design-implementation stage of the SDLC. By adopting early techniques of capturing the performance issues using the right models, predictability of the systems, software's and infrastructure increases manifold thereby ensuring deployment of highly efficient and effective system in the production. This definitely brings in both tangible and intangible benefits with good ROI and client satisfaction as well as to the application team. This also leads to increased CAPEX investment and reduced OPEX, less maintenance and production support costs thereby back feeding for better planning and strategic investments to beat the competition.

Speaker profile(s)

Name: Hemalatha Murugesan

Education: 1994: Bachelor of Engineering

Total Experience: 11+ years

Software Testing Experience: 11+ Years

Experience:

Hemalatha Murugesan is heading the Enterprise Performance Testing Solutions (EPTS) - IVS, Infosys Technologies Ltd, India. Hema has incubated EPTS which offers performance testing solutions and services and has setup the State – of – Art Performance Testing Lab at Infosys. She has also setup Testing Competency centres at Cognizant Technology Solutions and Aditi Technologies. She has published many papers at various international testing conferences as well as co-authored a book in testing published by Tata-McGraw-Hill – "*Software Testing: Effective Methods, Tools and Techniques*". She has published papers on topics such as testing being career mainstream, on testing lifecycle, functional testing as well as ROI through performance testing. She has published an article in the forthcoming HSBC's Guide to Trade Cash and Treasury Management in Asia Pacific (2008) – "*Performance Testing of Cash Management Applications*" and in Standard Chartered Middle East and Africa – The Guide to Working Capital Management 2007/2008 – "*Ensuring High Availability of Corporate Treasury Applications in South East Asian Countries*" which is currently under publication. She has spoken at leading engineering colleges to showcase the benefits of taking testing as a career. She has presented a talk on *Performance Testing – on what it is not?* Through EduSat – reaching out to over 120 engineering colleges under VTU using the satellite medium.

She has over 11 years of experience spanning all areas of testing – including functional, automation, internalization, performance, etc. She has worked on multiple technologies – J2EE, .net, legacy, RDBMS, etc. across various domains; banking, insurance, retail, healthcare, etc. She has led consulting assignments for leading Fortune 100 clients and setup Performance Testing Centre of Excellences. Her interests lie in pioneering new areas of testing working closely with the Industry trends. She is the recipient of the Infosys Excellence Award for Enterprise Performance Testing Solutions in Business Solutions as well as the Infosys Value Champion Excellence Award for 2007.