Whitney Associates

Performance Engineering Solutions

Assuring Performance in Large IT Systems

March 2004

1.0 Introduction

Distributed computing has captured almost universal attention as corporations struggle to gain competitive advantage in the information age. Currently, however, only 16% of distributed computing projects are completed within budget and on time. Fifty-three percent of projects run over budget, with an average cost overrun of 90%. Worse, fully 33% of all these projects are abandoned.[1] Figures on the success of eCommerce projects are yet to be compiled, but are likely to follow a similar pattern.

So what's happening in the information technology industry. Are colleges and universities no longer turning out the bright, motivated graduates who will become the leaders we need to implement information systems successfully? Are hardware and software vendors manufacturing products of inferior quality that are too difficult to use? Is distributed computing simply a bad idea?

On the contrary, distributed computing and component based solutions promise some very great advances in information technology. The potential for flexibility, scalability and cost-effectiveness with such solutions is unprecedented. Furthermore, these systems offer distinct business advantages, providing easier access to data and applications in the global enterprise.

Yet the very aspects of the systems that give them power and flexibility also pose significant new challenges and risks. Their open, distributed nature makes orchestrating and maintaining them far more complex than mainframe-based systems. In addition, most distributed computer systems pioneer new territory in both functionality and performance.

To ensure that today's complex systems need these goals a simple engineering process is required to address them. *Performance Assurance* is the process that, when applied during the design and development cycle, will greatly increase the chances of a project achieving user performance requirements first time round.

[1] Research from a Standish Group Survey

However, make no mistake, *Performance Assurance* will add cost to a project. Nonetheless, it is the only way to avoid the information system disasters making headlines every day. A few successful IT organisations have begun to implement *performance assurance* as a quintessential component of distributed system development. This paper discusses the primary risk factors in development projects, the keys to a successful risk management program, and the tools required.

2.0 Risk Factors

The are many risk factors that can lead to a poorly performing system:

2.1 Inadequately Estimating Demand

Any system, no matter how well designed, will not be able to cope with a massive increase in user demand. Therefore, it is imperative to determine the workload requirements and scalability requirement of the system correctly and in advance.

2.2 Loose Performance Requirements

Poorly specified performance requirements can lead to a poorly performing system because designers can easily, ignore them!

2.3 Unbalanced Architecture

A well-performing system will have a balanced architecture where no component is too over- or under- utilized. For example, you don't put a jet engine in a Ford Fiesta.

2.4 Poor Work Scheduling

An architecture may be capable of performing the work, but if the work is not scheduled or prioritized properly then performance problems can occur. This is particularly important at times of high load when low priority operations need to be suspended.

2.5 No Live Monitoring

A system cannot be left to run unmonitored and still deliver good performance as time progresses. It is crucial that changes in system performance are monitored and planned for.

2.6 Single Threaded Code

Not every part of the application needs to be multi-threaded, the trick is in determining which parts need to be multi-threaded.

2.7 Poor Testing

Poor testing will not find performance bottlenecks and may even give a false impression about the capability of the system.

2.8 Believing Vendors' Benchmarks

Although vendors do not lie when presenting benchmark results it must remembered a vendor's benchmark is only an indication of performance. The vendor's benchmark may not represent your actual use of the system or the configuration of your system.

There are also management and political risks:

2.9 Wishful Thinking

There is the hope that performance will just be fine and that there is no significant way of designing it into the system anyway. The project managers that support this idea also hope that one day they will win the lottery!

2.10 Leading Edge Technology

Some managers believe that if the architecture chosen looks like leading edge technology and has all the latest software then it is sure to perform.

2.11 "Tune it Later" Approach

There is an approach that says we can tune the system later after it is built. This approach is commonly used, but has 2 distinct problems –

i) changes come at the end of the development cycle when time is short and changes are the most costly;

ii) it assumes that the problem can be solved by simple changes.

2.12 Process Not Understood by Management

More depth of understanding is required to forecast the benefit of *performance assurance*. It takes a good manager to understand the benefits of the *performance assurance* process.

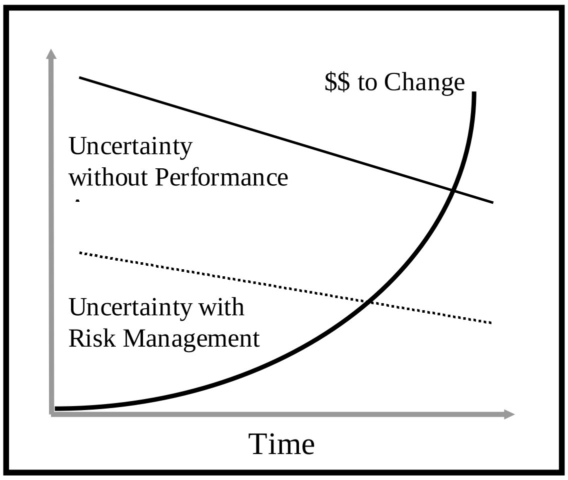
3.0 Strategy

Performance Assurance benefits a project by avoiding costly mistakes. This involves both foreseeing structural and functional problems before they arise, and detecting existing problems in the early stages so that they can be corrected at a lower cost. Since distributed systems present so many risk factors, *Performance Assurance* in a computer system development project offers great rewards. Organisations that have implemented *Performance Assurance* to address each of these areas have had a far greater chance of developing a successful system than those that don't.

A successful *Performance Assurance* program should begin early and continue throughout the project life cycle. For systems development, this is especially true; Figure 1 illustrates why. The cost of making a change to a system increases dramatically with time. On the other hand, the uncertainty of how a system will function and perform decreases with time; in the early stages of a project, uncertainty can be quite high. Without *Performance Assurance*, much of the early phase of system design is "guesswork," increasing the chances of needing to make costly changes later in the development life cycle. *Performance Assurance* reduces uncertainty, allowing necessary changes to be implemented early or eliminating the need for a change altogether.

Determining the need for changes in the early stages of a project pays handsomely. Making a change during the design phase is vastly simpler than making a change during development; re-designing a system on paper is always much easier than re-designing one in the data center. If hardware and software have already been purchased, there is less latitude in what can be done to correct a problem. Furthermore, as shown in Figure 1, the later a problem is found, the more costly it is to fix.

FIGURE 1. The Benefits of Performance Assurance



Late stage changes and fixes are costly for many reasons. The further into development, the more there is to repair if a flaw appears. Database tables, stored procedures and triggers, C and C++ routines, GUI windows and much more could all be impacted by a single change. Worse, if the system fails or needs modification during the production phase, the cost of downtime—not to mention lost business, customers, or reputation—must be factored into the cost of a fix.

In short, *Performance Assurance* applied early can save a great deal of time and money in the long run and boost the overall quality of an information system. But the early stage of a project is not the only time *Performance Assurance* delivers value.

Applying *Performance Assurance* throughout the project life cycle, from planning through production, is the key to a successful distributed system. While risk and uncertainty are especially high in the early stages of a project, they never disappear—every addition or modification to a system introduces new risk.

During development, the addition of new features affects existing code and data structures. The sooner bugs and errors are detected, the less costly they are to fix, so new features should be tested as they are added. Likewise, during production, the addition or removal of users, data, hardware, software and networks, and the imposition of new requirements all contribute to the need for continued risk management.

4.0 The Process

A successful *Performance Assurance* program covers the full project life cycle. Different project stages call for different approaches. The following constitute the key ingredients for a successful *Performance Assurance* program. The amount of work performed at each stage will depend on the complexity of the project.

4.1 Define Goals

Before proceeding with the project the performance requirements and the workload the system is going to be subjected to must be known. To quote a project manager “if the performance box isn’t there to be ticked it doesn’t get done.”

The performance requirements must be driven by user and business requirements but consideration must be given to the cost of achieving those requirements. For example, on-line credit card authentication is desirable, but expensive to implement, while off-line and email notification is a more cost-effective solution.

The workload that the system is expected to process must also be defined. Workload for many eCommerce systems is very difficult to ascertain since the potential customer base is so large. However, the investment in an IT system must be based on a business plan that has estimates for the traffic to the site and these can provide the basis for defining the workload. However, the key to successful IT system will be in ensuring that they can scale and have few logical bottlenecks.

4.2 Assess Performance Risk

Performance Assurance is essentially a risk mitigation process and therefore the amount of assurance effort and detail required for each project will be different. For example an internal project that uses known technology for a low number of users will require less effort than a high volume, new technology front end system. The risk assessment process helps guide the performance assurance process to address the most critical areas.

4.3 Performance Modelling

A performance model provides a means of evaluating a system design before the actual system is implemented. Typically these modes are either built using mathematical analytical techniques or computer based simulation models. A model of a system, including descriptions of the hardware, software and networks, is constructed. The proposed workload is then simulated on the model system, and statistics are gathered. Response time, throughput and other system performance statistics can be determined, and often bottlenecks are pinpointed. System scalability can be tested simply by scaling the model or workload.

Since modifying a model or workload and performing multiple simulations is much more economical and facile than modifying an actual system, it is an efficient method for evaluating design alternatives in the early stages of a project. A system can be modeled and simulated early, revealing design problems. Simulation results can be analysed to determine the source of any problems, then the system design is duly modified. The new system is then modeled and simulated. If new problems arise, more modifications can be made until the system meets requirements.

Thus, modeling and simulation address system performance and scalability in the early project phases, reducing the risk of starting out on the wrong path.

Performance models will also identify the critical components that are candidates for benchmarking.

4.3 Benchmarking

While simulation is ideal for evaluating design alternatives in the early stages of a project, benchmarking an actual implementation or part of an implementation is crucial to verifying performance and scalability.

The simulation model should be used to determine what part of the system is the most performance critical and therefore needs further examination.

Benchmarking involves running an actual workload on a real, physical system, with actual data or realistic test data. A benchmark is the only way to validate performance with certainty in a complex system.

Benchmarking can benefit a project as soon as development begins. After choosing the best design alternative based on simulation and modeling, construction can begin on the various modules of the system. Performance of a system component or prototype should be evaluated as soon as it becomes available. A simple benchmark can reveal performance problems early in development, when they can be fixed more easily and affect fewer system components.

4.4 Performance Budgets

Performance budgets are a crucial component in large project where software components are constructed. A performance budget is allocated to individual subsystem or modules so that developers for those modules are aware of the performance requirements of their code. Without performance budgets the designers do not know if they should optimise their code or not, leading to code that should be optimized not being so or even code that doesn’t need to be optimised being optimised!

Also at this stage, it is important to define and implement metric that will measure system performance and usage.

4.5 Performance Testing

Performance testing, once the system has completed functional testing, will highlight any final performance problems. However, care must be taken to ensure that performance testing is carried out correctly.

To complete performance testing correctly it is important that realistic test data are used, a large workload load is generated, and that databases, etc., are sized to the same they expect to be in real life.

4.6 Tuning

The operational use of the system will always be different from the expected usage predicted in the requirements and therefore the configuration options in the system must be tuned to achieve optimal performance.

Tuning is performed by collecting various metric defined during the performance budget stage and other system utilities.

5.0 Summary

In an era of increasing global competition, where information is the key to success, eCommerce offers many advantages, from the perspectives of both business and technology. Although distributed computing presents new challenges, simulation, benchmarking, performance budgets and testing applied early and program-wide eliminate much of the risk associated with eCommerce development. ECommerce-Performance Ltd provides clients with the advantage they need to succeed in today's rapidly evolving technology environment.

7.0 Contact Information

Contact Andrew Lee +44 (0)1684 551733

E-mail: info@WhitneyAssociates.com
Web: http://www.WhitneyAssociaties.com