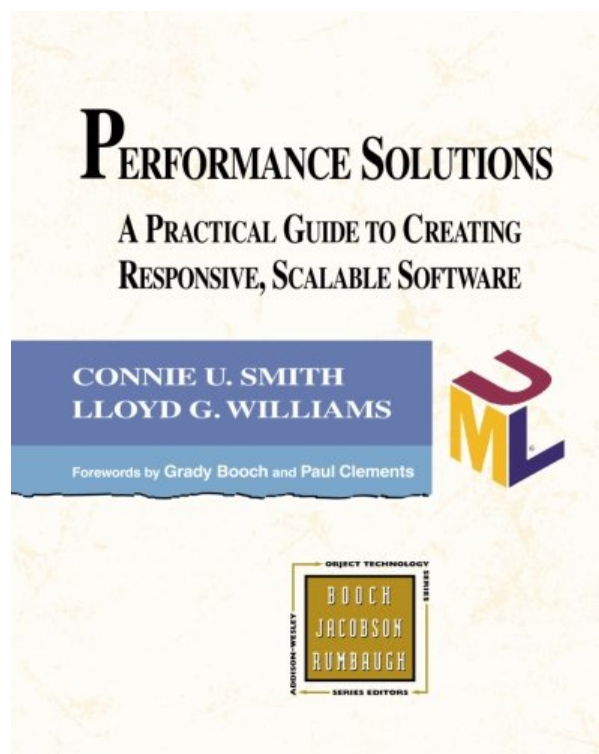


# **PERFORMANCE SOLUTIONS: A PRACTICAL GUIDE TO CREATING RESPONSIVE, SCALABLE SOFTWARE BY CONNIE U. SMITH, LLOYD G. WILLIAMS**



**DOWNLOAD EBOOK : PERFORMANCE SOLUTIONS: A PRACTICAL GUIDE  
TO CREATING RESPONSIVE, SCALABLE SOFTWARE BY CONNIE U. SMITH,  
LLOYD G. WILLIAMS PDF**



# PERFORMANCE SOLUTIONS

A PRACTICAL GUIDE TO CREATING  
RESPONSIVE, SCALABLE SOFTWARE

CONNIE U. SMITH  
LLOYD G. WILLIAMS

Forewords by Grady Booch and Paul Clements



Click link bellow and free register to download ebook:  
**PERFORMANCE SOLUTIONS: A PRACTICAL GUIDE TO CREATING RESPONSIVE,  
SCALABLE SOFTWARE BY CONNIE U. SMITH, LLOYD G. WILLIAMS**

[DOWNLOAD FROM OUR ONLINE LIBRARY](#)

# **PERFORMANCE SOLUTIONS: A PRACTICAL GUIDE TO CREATING RESPONSIVE, SCALABLE SOFTWARE BY CONNIE U. SMITH, LLOYD G. WILLIAMS PDF**

By downloading this soft documents e-book **Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams** in the online web link download, you remain in the initial step right to do. This site really offers you convenience of how to obtain the very best book, from finest seller to the new released e-book. You can discover a lot more books in this site by checking out every web link that we supply. One of the collections, Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams is one of the most effective collections to offer. So, the initial you obtain it, the first you will obtain all favorable concerning this book Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams

From the Publisher

Software Performance is an important area of concern for practitioners. Even so, very few books have been published about it, and these books have been so mathematical in their presentations that few readers have been able understand or apply them. During the pre-publication review process, it became very apparent that Connie Smith and Lloyd Williams were writing a Software Performance book that would be accessible to a much wider audience, and that they were doing so without abandoning rigor or integrity. They were writing a book that practitioners could use to their immediate and practical advantage. Upon completion of their work, we were delighted to have our sense of the book confirmed by Paul Clements in his foreword to the book:

"The result is a book that utterly demystifies the job (no longer an art) of performance engineering. Monsters, begone! Wizards, away! It leaves you feeling that you could really do this on your own."

We take great pride in publishing this book, and truly believe that it will be a lasting contribution to the literature of the software engineering field.

From the Inside Flap

In our roles as consultants, teachers, and mentors to software developers, we often see software products that fail to meet their performance objectives when they are initially constructed. Fixing these problems is costly and causes schedule delays, cost overruns, lost productivity, damaged customer relations, missed market windows, lost revenues and a host of other difficulties. In extreme cases, it may not be possible to fix performance problems without extensive redesign and re-implementation. In those cases, the project either becomes an infinite sink for time and money or is, mercifully, cancelled.

These problems can be prevented by the systematic application of a few simple performance analysis and prediction techniques. Over the years, we have helped many clients produce software that meets performance objectives and is delivered on time and within budget. This book was produced in response to requests from

our clients and students to provide a reference to the techniques that we have used and taught so successfully. Objectives By applying the material in this book you will be able to answer questions such as: Will your users be able to complete their tasks in the allotted time? Are your hardware and network capable of handling the load? Will your system scale up to meet future demand?

More importantly, you will be able to answer these questions before you have committed a lot of time and money to an implementation, only to find that the answers are an emphatic (and expensive) no!

You will learn enough from this book to begin to apply these techniques immediately to manage the performance of your software systems. You will sharpen your skills as you use them on a variety of projects that include: web-based applications, distributed systems, real-time systems, traditional database applications, and others. Software Performance

Performance is any characteristic of a software product that you could, in principle, measure by sitting at the computer with a stopwatch in your hand. The dimensions of performance include: response time, throughput and scalability.

How do projects get in trouble with performance? The problem is often due to a fundamental misunderstanding of how to achieve performance objectives. The approach is frequently: "First, let's make it run; then we'll make it run fast." The idea is to get the functionality right, then tune for performance. Unfortunately, by the time the architecture and design are selected, it may already be too late to achieve adequate performance by tuning.

The proper way to manage software performance is to systematically plan for and predict the performance of the emerging software throughout the development process. This book presents a set of simple techniques that you can use to manage the performance of your software. These techniques do not require an advanced degree in mathematics nor do they take significant amounts of time away from development activities. By applying them you will be able to make informed choices among architectural and design alternatives and proceed with confidence that your software will meet its performance objectives.

The approach to managing performance presented here is unique. It has grown out of over ten years of collaboration between the authors that combines knowledge and experience in both software performance engineering and architectural analysis. Other authors have proposed elaborate modeling techniques which either aren't useful for real systems, or require a Ph.D. in math to apply them. Our approach is practical, useful by non-specialists, and rigorous. Software Performance Engineering

The techniques presented in this book are collectively known as software performance engineering (SPE). SPE is a comprehensive way of managing performance that includes principles for creating responsive software, 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. Modeling is central to both SPE and object-oriented development. 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. SPE uses models to quantitatively assess the performance of the emerging software. This book discusses how to quickly and easily create those quantitative models from the architecture and design models that you produce as part of the object-oriented development process. It also describes how to integrate SPE seamlessly into the overall object-oriented development process.

Finally, these techniques are neither new nor revolutionary. They have evolved from a relatively long history

of proven quantitative disciplines. Neither are they a silver bullet. They must be used to be effective, and it will take some time to learn to apply them, particularly on your initial project. The amount of time required, however, is not excessive and appropriate for most projects in their early stages. We have found that when problems are inevitable your choice is "pay a little now (for scalability and performance) or pay much more later." The only way to determine whether they are inevitable it to use quantitative techniques early. Who Should Read This Book?

This book is primarily intended for experienced software developers who have used object-oriented techniques on one or more development projects and who are ready for the next step: to learn how to create software systems with built-in scalability and performance. The emphasis is on how to apply the techniques. Other readers will also find information that they can use:

Developers who are new to object-oriented techniques will discover how it is possible to manage software for object-oriented systems and find guidelines for creating responsive software that will augment their study of object-oriented development. Performance specialists will learn how to adapt their skills to object-oriented systems as well as discover new approaches that are applicable to other development technologies. Students will learn about performance issues that arise in realistic, large-scale software systems and how to solve them. Project managers will find techniques that they can bring to their projects immediately to improve the way performance is managed. They will also learn how to implement SPE.

Familiarity with object-oriented concepts is helpful but not essential. We explain the portions of the UML notation useful for performance assessments. If you find you would like more background, refer to other books in the Object Technology series.

Familiarity with performance modeling techniques is helpful but not essential. We assume that you will use an SPE tool, such as SPE\*ED, to evaluate your systems. We explain enough of the modeling fundamentals for you to become familiar with how such tools work. References are included with the material if you would like to learn more about the modeling technology.

### Organization of This Book

This book is organized into seven parts:

Part I Introduction and Overview provides an introduction to SPE and an overview of the modeling techniques. Part II SPE Models describes details of the models used in SPE and their solution. Part III Data Collection discusses how to obtain SPE data and some background on performance measurement techniques. Part IV Software Evolution Issues covers techniques for designing performance into software systems, and maintaining performance throughout its life cycle. Part V Applications illustrates the application of SPE techniques to some important types of application domains. Part VI Making SPE Happen discusses how to implement SPE in your development organization. Part VII Appendixes summarize the notation used throughout the book.

0201722291P04062001

### From the Back Cover

"Utterly demystifies the job (no longer the art) of performance engineering. Monsters, begone! Wizards, away! It leaves you feeling that you could really do this on your own. And, thanks to Connie and Lloyd, you can."

--From the Foreword by Paul Clements

Performance, responsiveness, and scalability are vital aspects of today's software. Yet many software systems cannot be used as initially implemented because of performance problems. Such failures can translate into significant costs for software developers due to damaged customer relations, lost income, and time and budget overruns.

Performance Solutions offers straightforward techniques and strategies that can be used by software developers, project managers, and performance specialists to handle the performance of software throughout the development process. In particular the book focuses on how to design performance into software systems early and then maintain performance throughout the software development life cycle.

Software Performance Engineering (SPE) is a systematic, quantitative approach to cost-effectively constructing software systems that meet performance objectives. This book details the various SPE models that can be used for a wide variety of applications, including Web-based and distributed systems. It describes effective data gathering and performance measurement techniques and explores the principles of performance-oriented design. You will also find practical guidance on implementing an SPE program. Specific topics of interest include:

- Reactive versus proactive performance management
- Using UML for SPE
- Specifying key performance scenarios and performance objectives
- Gathering performance data and estimating resource requirements
- Constructing and solving performance models
- Modeling complex interactions in distributed systems
- Planning and conducting performance measurements
- Principles for performance-oriented design
- Patterns that illustrate "best practice" approaches to achieving responsiveness and scalability
- Antipatterns that illustrate what not to do and how to fix a problem when you find it
- SPE activities for mid- to late life cycle
- Performance tuning strategies
- Integrating SPE into your software process
- SPE implementation strategies and critical success factors

Numerous real-world applications and case studies illustrate the application of SPE techniques to important application domains, including Web and embedded real-time systems.

With this book, you will gain the understanding and skills you need to make informed choices among architectural and design alternatives and create responsive, scalable software on time and within your budget.

0201722291B07162001

# **PERFORMANCE SOLUTIONS: A PRACTICAL GUIDE TO CREATING RESPONSIVE, SCALABLE SOFTWARE BY CONNIE U. SMITH, LLOYD G. WILLIAMS PDF**

[Download: PERFORMANCE SOLUTIONS: A PRACTICAL GUIDE TO CREATING RESPONSIVE, SCALABLE SOFTWARE BY CONNIE U. SMITH, LLOYD G. WILLIAMS PDF](#)

Simply for you today! Discover your preferred e-book here by downloading and install as well as getting the soft data of the publication **Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams** This is not your time to commonly go to guide stores to purchase a publication. Here, selections of book Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams and collections are offered to download. One of them is this Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams as your favored book. Obtaining this e-book Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams by on-line in this site could be understood now by seeing the web link page to download and install. It will certainly be simple. Why should be right here?

As one of the window to open the new globe, this *Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams* supplies its impressive writing from the writer. Released in one of the prominent authors, this book Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams turns into one of the most needed publications recently. Really, guide will not matter if that Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams is a best seller or not. Every book will still give best resources to obtain the visitor all finest.

However, some people will seek for the best vendor publication to check out as the initial referral. This is why; this Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams exists to fulfil your necessity. Some people like reading this book Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams because of this prominent publication, but some love this because of preferred author. Or, several also like reading this book [Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams](#) due to the fact that they really should read this book. It can be the one that actually love reading.

# **PERFORMANCE SOLUTIONS: A PRACTICAL GUIDE TO CREATING RESPONSIVE, SCALABLE SOFTWARE BY CONNIE U. SMITH, LLOYD G. WILLIAMS PDF**

This title provides systematic performance planning techniques for diverse computing environments and architectures. It seeks to smoothly integrate performance analysis into an existing software development process.

- Sales Rank: #593695 in Books
- Published on: 2001-09-27
- Released on: 2001-09-17
- Original language: English
- Number of items: 1
- Dimensions: 9.00" h x 1.30" w x 7.20" l, 2.00 pounds
- Binding: Paperback
- 544 pages

From the Publisher

Software Performance is an important area of concern for practitioners. Even so, very few books have been published about it, and these books have been so mathematical in their presentations that few readers have been able understand or apply them. During the pre-publication review process, it became very apparent that Connie Smith and Lloyd Williams were writing a Software Performance book that would be accessible to a much wider audience, and that they were doing so without abandoning rigor or integrity. They were writing a book that practitioners could use to their immediate and practical advantage. Upon completion of their work, we were delighted to have our sense of the book confirmed by Paul Clements in his foreword to the book:

"The result is a book that utterly demystifies the job (no longer an art) of performance engineering. Monsters, begone! Wizards, away! It leaves you feeling that you could really do this on your own."

We take great pride in publishing this book, and truly believe that it will be a lasting contribution to the literature of the software engineering field.

From the Inside Flap

In our roles as consultants, teachers, and mentors to software developers, we often see software products that fail to meet their performance objectives when they are initially constructed. Fixing these problems is costly and causes schedule delays, cost overruns, lost productivity, damaged customer relations, missed market windows, lost revenues and a host of other difficulties. In extreme cases, it may not be possible to fix performance problems without extensive redesign and re-implementation. In those cases, the project either becomes an infinite sink for time and money or is, mercifully, cancelled.

These problems can be prevented by the systematic application of a few simple performance analysis and prediction techniques. Over the years, we have helped many clients produce software that meets performance



objectives and is delivered on time and within budget. This book was produced in response to requests from our clients and students to provide a reference to the techniques that we have used and taught so successfully. Objectives By applying the material in this book you will be able to answer questions such as: Will your users be able to complete their tasks in the allotted time? Are your hardware and network capable of handling the load? Will your system scale up to meet future demand?

More importantly, you will be able to answer these questions before you have committed a lot of time and money to an implementation, only to find that the answers are an emphatic (and expensive) no!

You will learn enough from this book to begin to apply these techniques immediately to manage the performance of your software systems. You will sharpen your skills as you use them on a variety of projects that include: web-based applications, distributed systems, real-time systems, traditional database applications, and others. Software Performance

Performance is any characteristic of a software product that you could, in principle, measure by sitting at the computer with a stopwatch in your hand. The dimensions of performance include: response time, throughput and scalability.

How do projects get in trouble with performance? The problem is often due to a fundamental misunderstanding of how to achieve performance objectives. The approach is frequently: "First, let's make it run; then we'll make it run fast." The idea is to get the functionality right, then tune for performance. Unfortunately, by the time the architecture and design are selected, it may already be too late to achieve adequate performance by tuning.

The proper way to manage software performance is to systematically plan for and predict the performance of the emerging software throughout the development process. This book presents a set of simple techniques that you can use to manage the performance of your software. These techniques do not require an advanced degree in mathematics nor do they take significant amounts of time away from development activities. By applying them you will be able to make informed choices among architectural and design alternatives and proceed with confidence that your software will meet its performance objectives.

The approach to managing performance presented here is unique. It has grown out of over ten years of collaboration between the authors that combines knowledge and experience in both software performance engineering and architectural analysis. Other authors have proposed elaborate modeling techniques which either aren't useful for real systems, or require a Ph.D. in math to apply them. Our approach is practical, useful by non-specialists, and rigorous. Software Performance Engineering

The techniques presented in this book are collectively known as software performance engineering (SPE). SPE is a comprehensive way of managing performance that includes principles for creating responsive software, 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. Modeling is central to both SPE and object-oriented development. 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. SPE uses models to quantitatively assess the performance of the emerging software. This book discusses how to quickly and easily create those quantitative models from the architecture and design models that you produce as part of the object-oriented development process. It also describes how to integrate SPE seamlessly into the overall object-oriented development process.

Finally, these techniques are neither new nor revolutionary. They have evolved from a relatively long history of proven quantitative disciplines. Neither are they a silver bullet. They must be used to be effective, and it will take some time to learn to apply them, particularly on your initial project. The amount of time required, however, is not excessive and appropriate for most projects in their early stages. We have found that when problems are inevitable your choice is "pay a little now (for scalability and performance) or pay much more later." The only way to determine whether they are inevitable is to use quantitative techniques early. Who Should Read This Book?

This book is primarily intended for experienced software developers who have used object-oriented techniques on one or more development projects and who are ready for the next step: to learn how to create software systems with built-in scalability and performance. The emphasis is on how to apply the techniques. Other readers will also find information that they can use:

Developers who are new to object-oriented techniques will discover how it is possible to manage software for object-oriented systems and find guidelines for creating responsive software that will augment their study of object-oriented development. Performance specialists will learn how to adapt their skills to object-oriented systems as well as discover new approaches that are applicable to other development technologies. Students will learn about performance issues that arise in realistic, large-scale software systems and how to solve them. Project managers will find techniques that they can bring to their projects immediately to improve the way performance is managed. They will also learn how to implement SPE.

Familiarity with object-oriented concepts is helpful but not essential. We explain the portions of the UML notation useful for performance assessments. If you find you would like more background, refer to other books in the Object Technology series.

Familiarity with performance modeling techniques is helpful but not essential. We assume that you will use an SPE tool, such as SPE\*ED, to evaluate your systems. We explain enough of the modeling fundamentals for you to become familiar with how such tools work. References are included with the material if you would like to learn more about the modeling technology.

#### Organization of This Book

This book is organized into seven parts:

Part I Introduction and Overview provides an introduction to SPE and an overview of the modeling techniques. Part II SPE Models describes details of the models used in SPE and their solution. Part III Data Collection discusses how to obtain SPE data and some background on performance measurement techniques. Part IV Software Evolution Issues covers techniques for designing performance into software systems, and maintaining performance throughout its life cycle. Part V Applications illustrates the application of SPE techniques to some important types of application domains. Part VI Making SPE Happen discusses how to implement SPE in your development organization. Part VII Appendixes summarize the notation used throughout the book.

0201722291P04062001

From the Back Cover

"Utterly demystifies the job (no longer the art) of performance engineering. Monsters, begone! Wizards, away! It leaves you feeling that you could really do this on your own. And, thanks to Connie and Lloyd, you can."

--From the Foreword by Paul Clements

Performance, responsiveness, and scalability are vital aspects of today's software. Yet many software systems cannot be used as initially implemented because of performance problems. Such failures can translate into significant costs for software developers due to damaged customer relations, lost income, and time and budget overruns.

Performance Solutions offers straightforward techniques and strategies that can be used by software developers, project managers, and performance specialists to handle the performance of software throughout the development process. In particular the book focuses on how to design performance into software systems early and then maintain performance throughout the software development life cycle.

Software Performance Engineering (SPE) is a systematic, quantitative approach to cost-effectively constructing software systems that meet performance objectives. This book details the various SPE models that can be used for a wide variety of applications, including Web-based and distributed systems. It describes effective data gathering and performance measurement techniques and explores the principles of performance-oriented design. You will also find practical guidance on implementing an SPE program. Specific topics of interest include:

- Reactive versus proactive performance management
- Using UML for SPE
- Specifying key performance scenarios and performance objectives
- Gathering performance data and estimating resource requirements
- Constructing and solving performance models
- Modeling complex interactions in distributed systems
- Planning and conducting performance measurements
- Principles for performance-oriented design
- Patterns that illustrate "best practice" approaches to achieving responsiveness and scalability
- Antipatterns that illustrate what not to do and how to fix a problem when you find it
- SPE activities for mid- to late life cycle
- Performance tuning strategies
- Integrating SPE into your software process
- SPE implementation strategies and critical success factors

Numerous real-world applications and case studies illustrate the application of SPE techniques to important application domains, including Web and embedded real-time systems.

With this book, you will gain the understanding and skills you need to make informed choices among architectural and design alternatives and create responsive, scalable software on time and within your budget.

0201722291B07162001

Most helpful customer reviews

32 of 33 people found the following review helpful.

Must reading for OO architects

By Mike Tarrani

This book is one-of-a-kind in that it addresses head on the thorny problems associated with object-oriented performance and scalability. The book is divided into seven parts, which include 16 chapters and two

appendices.

Part I introduces software performance and the authors' software performance engineering (SPE) methodology. Although the book continues to drill down deeper into SPE and associated factors in subsequent chapters, this part of the book is my favorite. What makes it my favorite is the context in which the authors cast performance as a function of resource requirements (workload) and configuration (capacity). Although this is not a new concept, the ensuing discussion leading to SPE modeling strategies and models and the 9-step SPE process exposes the challenges and provides a sense that OO performance can be managed through careful systems analysis, modeling and design early in the life cycle. I also like the way SPE is aligned to the "Unified Software Process", which is the RUP thinly disguised. The chapter that discusses this has some realm gems, including performance patterns and anti-patterns. In addition, the SPE is also aligned to UML, with an excellent discussion on extending the UML and some example scenarios that show how to specify time, concurrency and other performance characteristics.

The SPE models given in Part II cover the full spectrum of system types, including distributed and web-based systems. The material is highly technical and requires close attention. It is also clearly written and will provide the design team (not to mention the post-implementation support team) with analytical techniques and an effective analysis approach to performance management. The highlights of this part of the book were the way middleware overhead is taken into account, scenarios and modeling hints.

Data collection is the topic of Part III and is covered in detail. The chapters I most liked in this part included resource estimation techniques and software measurement and instrumentation. Part IV will be the focal point for designers and architects in that it addresses performance solutions. Chapter topics include: performance-oriented design, performance patterns and performance anti-patterns(excellent material!) and implementation solutions. The latter covers performance tuning, as well as language-dependent and -independent solutions for OO software. The two languages discussed are c++ and java.

Part V is devoted to applications and provides a thorough discussion of web and realtime embedded system performance issues. The format of the two chapters in this part include: performance issues, SPE models, environments and constraints. Realtime systems are given a more in-depth treatment because of the special nature of these systems. I liked the case study that used a telephone switching system for realtime systems because I have professional interests in that area.

The actual steps for implementing SPE is the subject of Part VI, which covers the process, associated life cycle activities, post-deployment performance management and how to integrate SPE into your organization's software process. Part VII's two appendices are valuable. Appendix A covers UML notation and Appendix B covers SPE modeling notations. Both are useful as quick reference guides.

This is a comprehensive book that, while focused on a narrow topic, covers all issues and factors in minute detail. The book complements two other outstanding works, Software Reliability Engineered Testing by John D. Musa, and Testing Object-Oriented Systems by Robert V. Binder. Although the latter books are more focused on testing, the material dovetails nicely with the SPE approach given in this book.

0 of 0 people found the following review helpful.

so it runs quickly from the outset and more easily achieves scalability

By J. Phillips

This book attempts to usher in formal rigorous Software Performance Engineering as a modelling process; building OO software, so it runs quickly from the outset and more easily achieves scalability, into a likely already existing regime. It ensures performance is achieved by a (9-step) prescriptive methodology. Like Connie Smith's "Performance Engineering of Software Systems", it is well written and conceived.

See all 2 customer reviews...

# **PERFORMANCE SOLUTIONS: A PRACTICAL GUIDE TO CREATING RESPONSIVE, SCALABLE SOFTWARE BY CONNIE U. SMITH, LLOYD G. WILLIAMS PDF**

In getting this **Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams**, you could not always go by strolling or using your electric motors to the book shops. Get the queuing, under the rainfall or warm light, as well as still look for the unidentified book to be in that publication store. By seeing this web page, you can just hunt for the Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams and also you could discover it. So now, this moment is for you to go for the download link and also acquisition Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams as your very own soft documents publication. You could read this publication Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams in soft documents only and also save it as your own. So, you don't need to hurriedly put the book Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams into your bag almost everywhere.

From the Publisher

Software Performance is an important area of concern for practitioners. Even so, very few books have been published about it, and these books have been so mathematical in their presentations that few readers have been able understand or apply them. During the pre-publication review process, it became very apparent that Connie Smith and Lloyd Williams were writing a Software Performance book that would be accessible to a much wider audience, and that they were doing so without abandoning rigor or integrity. They were writing a book that practitioners could use to their immediate and practical advantage. Upon completion of their work, we were delighted to have our sense of the book confirmed by Paul Clements in his foreword to the book:

"The result is a book that utterly demystifies the job (no longer an art) of performance engineering. Monsters, begone! Wizards, away! It leaves you feeling that you could really do this on your own."

We take great pride in publishing this book, and truly believe that it will be a lasting contribution to the literature of the software engineering field.

From the Inside Flap

In our roles as consultants, teachers, and mentors to software developers, we often see software products that fail to meet their performance objectives when they are initially constructed. Fixing these problems is costly and causes schedule delays, cost overruns, lost productivity, damaged customer relations, missed market windows, lost revenues and a host of other difficulties. In extreme cases, it may not be possible to fix performance problems without extensive redesign and re-implementation. In those cases, the project either becomes an infinite sink for time and money or is, mercifully, cancelled.

These problems can be prevented by the systematic application of a few simple performance analysis and prediction techniques. Over the years, we have helped many clients produce software that meets performance objectives and is delivered on time and within budget. This book was produced in response to requests from our clients and students to provide a reference to the techniques that we have used and taught so successfully.

Objectives By applying the material in this book you will be able to answer questions such as: Will your users be able to complete their tasks in the allotted time? Are your hardware and network capable of handling the load? Will your system scale up to meet future demand?

More importantly, you will be able to answer these questions before you have committed a lot of time and money to an implementation, only to find that the answers are an emphatic (and expensive) no!

You will learn enough from this book to begin to apply these techniques immediately to manage the performance of your software systems. You will sharpen your skills as you use them on a variety of projects that include: web-based applications, distributed systems, real-time systems, traditional database applications, and others. Software Performance

Performance is any characteristic of a software product that you could, in principle, measure by sitting at the computer with a stopwatch in your hand. The dimensions of performance include: response time, throughput and scalability.

How do projects get in trouble with performance? The problem is often due to a fundamental misunderstanding of how to achieve performance objectives. The approach is frequently: "First, let's make it run; then we'll make it run fast." The idea is to get the functionality right, then tune for performance. Unfortunately, by the time the architecture and design are selected, it may already be too late to achieve adequate performance by tuning.

The proper way to manage software performance is to systematically plan for and predict the performance of the emerging software throughout the development process. This book presents a set of simple techniques that you can use to manage the performance of your software. These techniques do not require an advanced degree in mathematics nor do they take significant amounts of time away from development activities. By applying them you will be able to make informed choices among architectural and design alternatives and proceed with confidence that your software will meet its performance objectives.

The approach to managing performance presented here is unique. It has grown out of over ten years of collaboration between the authors that combines knowledge and experience in both software performance engineering and architectural analysis. Other authors have proposed elaborate modeling techniques which either aren't useful for real systems, or require a Ph.D. in math to apply them. Our approach is practical, useful by non-specialists, and rigorous. Software Performance Engineering

The techniques presented in this book are collectively known as software performance engineering (SPE). SPE is a comprehensive way of managing performance that includes principles for creating responsive software, 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. Modeling is central to both SPE and object-oriented development. 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. SPE uses models to quantitatively assess the performance of the emerging software. This book discusses how to quickly and easily create those quantitative models from the architecture and design models that you produce as part of the object-oriented development process. It also describes how to integrate SPE seamlessly into the overall object-oriented development process.

Finally, these techniques are neither new nor revolutionary. They have evolved from a relatively long history of proven quantitative disciplines. Neither are they a silver bullet. They must be used to be effective, and it

will take some time to learn to apply them, particularly on your initial project. The amount of time required, however, is not excessive and appropriate for most projects in their early stages. We have found that when problems are inevitable your choice is "pay a little now (for scalability and performance) or pay much more later." The only way to determine whether they are inevitable it to use quantitative techniques early. Who Should Read This Book?

This book is primarily intended for experienced software developers who have used object-oriented techniques on one or more development projects and who are ready for the next step: to learn how to create software systems with built-in scalability and performance. The emphasis is on how to apply the techniques. Other readers will also find information that they can use:

Developers who are new to object-oriented techniques will discover how it is possible to manage software for object-oriented systems and find guidelines for creating responsive software that will augment their study of object-oriented development. Performance specialists will learn how to adapt their skills to object-oriented systems as well as discover new approaches that are applicable to other development technologies. Students will learn about performance issues that arise in realistic, large-scale software systems and how to solve them. Project managers will find techniques that they can bring to their projects immediately to improve the way performance is managed. They will also learn how to implement SPE.

Familiarity with object-oriented concepts is helpful but not essential. We explain the portions of the UML notation useful for performance assessments. If you find you would like more background, refer to other books in the Object Technology series.

Familiarity with performance modeling techniques is helpful but not essential. We assume that you will use an SPE tool, such as SPE\*ED, to evaluate your systems. We explain enough of the modeling fundamentals for you to become familiar with how such tools work. References are included with the material if you would like to learn more about the modeling technology.

#### Organization of This Book

This book is organized into seven parts:

Part I Introduction and Overview provides an introduction to SPE and an overview of the modeling techniques. Part II SPE Models describes details of the models used in SPE and their solution. Part III Data Collection discusses how to obtain SPE data and some background on performance measurement techniques. Part IV Software Evolution Issues covers techniques for designing performance into software systems, and maintaining performance throughout its life cycle. Part V Applications illustrates the application of SPE techniques to some important types of application domains. Part VI Making SPE Happen discusses how to implement SPE in your development organization. Part VII Appendixes summarize the notation used throughout the book.

0201722291P04062001

From the Back Cover

"Utterly demystifies the job (no longer the art) of performance engineering. Monsters, begone! Wizards, away! It leaves you feeling that you could really do this on your own. And, thanks to Connie and Lloyd, you can."

--From the Foreword by Paul Clements

Performance, responsiveness, and scalability are vital aspects of today's software. Yet many software systems cannot be used as initially implemented because of performance problems. Such failures can translate into significant costs for software developers due to damaged customer relations, lost income, and time and budget overruns.

Performance Solutions offers straightforward techniques and strategies that can be used by software developers, project managers, and performance specialists to handle the performance of software throughout the development process. In particular the book focuses on how to design performance into software systems early and then maintain performance throughout the software development life cycle.

Software Performance Engineering (SPE) is a systematic, quantitative approach to cost-effectively constructing software systems that meet performance objectives. This book details the various SPE models that can be used for a wide variety of applications, including Web-based and distributed systems. It describes effective data gathering and performance measurement techniques and explores the principles of performance-oriented design. You will also find practical guidance on implementing an SPE program. Specific topics of interest include:

- Reactive versus proactive performance management
- Using UML for SPE
- Specifying key performance scenarios and performance objectives
- Gathering performance data and estimating resource requirements
- Constructing and solving performance models
- Modeling complex interactions in distributed systems
- Planning and conducting performance measurements
- Principles for performance-oriented design
- Patterns that illustrate "best practice" approaches to achieving responsiveness and scalability
- Antipatterns that illustrate what not to do and how to fix a problem when you find it
- SPE activities for mid- to late life cycle
- Performance tuning strategies
- Integrating SPE into your software process
- SPE implementation strategies and critical success factors

Numerous real-world applications and case studies illustrate the application of SPE techniques to important application domains, including Web and embedded real-time systems.

With this book, you will gain the understanding and skills you need to make informed choices among architectural and design alternatives and create responsive, scalable software on time and within your budget.

0201722291B07162001

By downloading this soft documents e-book **Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams** in the online web link download, you remain in the initial step right to do. This site really offers you convenience of how to obtain the very best book, from finest seller to the new released e-book. You can discover a lot more books in this site by checking out every web link that we supply. One of the collections, Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software By Connie U. Smith, Lloyd G. Williams is one of the most effective collections to offer. So, the initial you obtain it, the first you will obtain all favorable concerning this book Performance Solutions: A Practical Guide To Creating Responsive, Scalable Software



By Connie U. Smith, Lloyd G. Williams