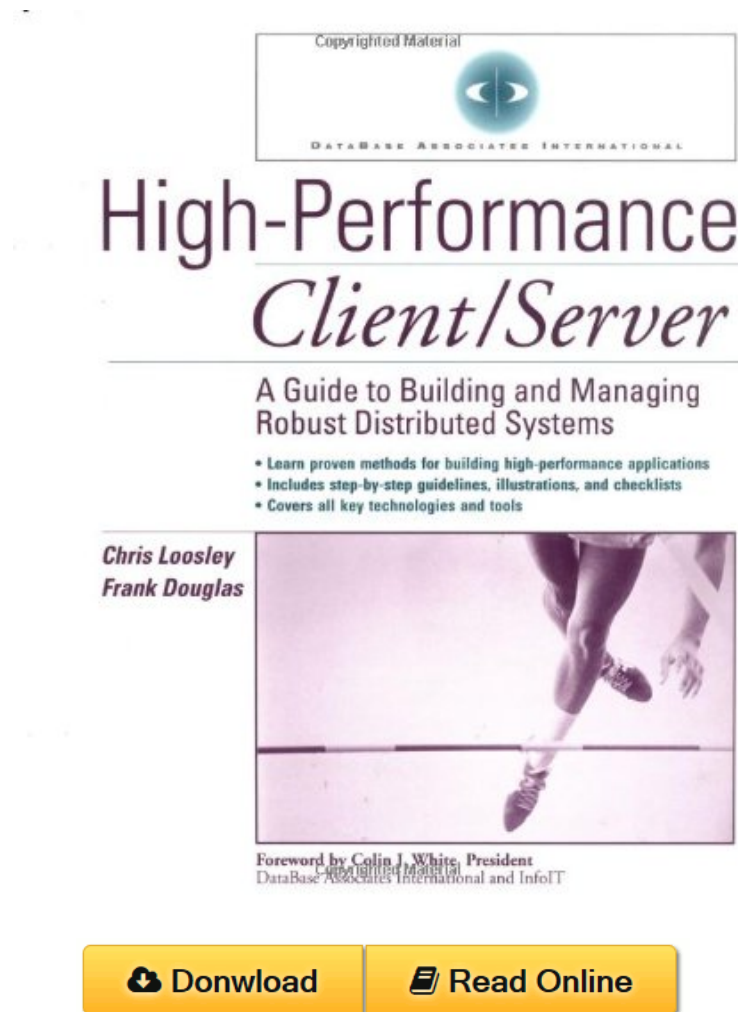


High-Performance Client/Server: A Guide to Building and Managing Robust Distributed Systems

Von *Chris Loosley, Frank Douglas*
ePub | *DOC | audiobook | ebooks | Download PDF



Produktinformation -Verkaufsrang: #3390294 in BcherMarke: Wiley und SonsVerffentlicht am: 1997-12Anzahl der Produkte: 1Abmessungen: 9.19 x 1.63b x 7.52l, .0 Pfund Einband: Taschenbuch752 Seiten | File size: 17.Mb

Von Chris Loosley, Frank Douglas : High-Performance Client/Server: A Guide to Building and Managing Robust Distributed Systems before purchasing it in order to gage whether or not it would be worth my time, and all praised High-Performance Client/Server: A Guide to Building and Managing Robust Distributed Systems:

KundenrezensionenHilfreichste Kundenrezensionen0 von 0 Kunden fanden die folgende Rezension hilfreich. Brilliant, comprehensive, humorous, definitive treatmentVon Lou Agosta (lagosta@21stcentury.net)Now that client-server is mainstream, it is OK to knock it. Specifically, performance problems arise with enterprise client-server systems due to the complexity of the distributed processing. The more hops between platforms, the more overhead, and the more points of failure. Client-server architecture is inherently distributed and often has to occur across multiple platforms connected by skinny wide area network (WAN) pipes. What can help? This wide-ranging diagnosis and treatment of the many aspects of the dilemma is highly recommended for its extent, depth, humor, and penetrating insight. Part One on performance fundamentals list 23 components of response time, offering incisive distinctions for both the beginner

and the advanced practitioner. When cross-referenced in the extended application resource usage matrix to identify bottlenecks, these components become powerful drivers of response time tuning using trade-offs, choices, and priorities to squeeze every ounce of performance out of the available computing configuration. Part Two treats the software performance engineering process. One important goal is to build a performance model out of the hardware/software environments, application flow, data structures and business factors. In addition to complex interrelations of computing components, Loosley provides pointers to some very simple principles and methods for tuning complex systems. Part Three on principles is the heart of the book. Software engineering principles - formality, completeness, simplicity - provide the foundation for design principles of abstraction, decomposition, and information hiding - which, in turn, support refinement, independence, and localization (p. 207). These are explained and applied in sufficient depth and detail so that practicing performance engineers will find both helpful tips and techniques, amusing anecdotes, and theoretic principles. Queuing theory is explained but not treated mathematically other than the marshaling of a few simple metrics of practical interest. With the emergence of parallel processing as a relatively new candidate solution for decision support and data warehouse applications, the Chapter on The Parallelism Principle contains one of the best concise explanations I have seen in the literature of the differences between massive parallel processing, non-uniform memory architecture, and clusters as a processing resource. Part Four on Applications drills down into middleware and performance. The authors argue the concept of logical unit of work transaction management is sound and well proven in the world of host-centered (i.e., mainframe) computing. However, when the architecture of synchronous communications is transferred to distributed client-server, then problems arise. As soon as one of the multitude of processors waits, the entire system is at risk of log jamming. And since all computers wait at the same speed - both a humorous and sobering anecdote called "Bell's Law" - no amount of souped up hardware or software will make a difference. The authors document at least twenty points in which enterprise client-server - that is, three tiered - systems can experience bottle necks. The problem with client-server is that frequently only the database and the client workstation are suitably instrumented to gather performance metrics and data; and, even then, it is the interaction between the component that is most significant, not what goes on within each taken in isolation. Therefore, there is no easy answer. The bottlenecks must be worked and pushed down stream and squeezed out of the system. However, in the view of the authors, what will make a difference is a high performance architecture built on a form of asynchronous multi transaction workflow using decoupled processes (sometimes called "message-queuing" or "MQ" middleware). This is a major conclusion for which the authors argue persuasively in the climatic Chapter 16 on Architecture for High Performance. Part Five on Technologies looks at the inner workings of relational database management systems, transaction managers and monitors, and data warehousing technologies such as OLAP, ROLAP, and multidimensional data analysis (MDA). The availability of this material rounds out the completeness and comprehensive scope of the treatment provided. The authors set a high standard for collecting insightful and humorous one-liners, with proper credit to many other brilliant contributors, which also cut to the heart of the challenges of delivering performance in a client-server environment. "Software workloads expand to consume the available computing capacity" (p. 11). "There is nothing so useless as doing with great efficiency what should not be done at all" (p. 80). "Always work on the biggest bottleneck" (p. 113). "If you can't see the bug, you are looking in the wrong place" (p. 333). These zingers kept me returning to this text time and again. The book contains a rich assortment of illustrative and instructive graphics. The figures and tables are superbly drawn and produced in attractive gray scale. The performance guidelines - and as benefits a thorough compendium of over 700 pages of encyclopedic proportions - are separately listed at the back in a voluminous section of over 500 entries, extensive enough to be designated as Part Six. Given the challenges of mainstream client-server, Loosley and Douglas are like the cavalry to the rescue with a comprehensive and richly-laden resource of distinctions and methods for understanding, addressing, and resolving the dilemmas faced by those tasked with building and managing distributed client-server. -- review originally submitted to Computing Reviews in 1998 -- but the good folks there already had someone else reviewing the book, so they decided not to publish it

Von 0 Kunden fanden die folgende Rezension hilfreich. Find your way through current design implementation issues
Von Ein Kunde
I came upon this book during my quest for some tools that would help me put some perspective on current software design and architecture issues. I am sure I was not alone in my efforts to understand the evolving client / server paradigm and the performance issues involved. I am a believer in the need to build performance into systems, that it can not be conveniently added afterwards. I also believe that scalability and adaptability are characteristics that result from good design. I was more than a little overwhelmed by all the marketing hype for the latest generation of Internet-related tools and the general level of discussion of "middleware". Then, I found this book! I was very pleased to see that the issues of partitioning logic across physical and functional layers were treated as equally important. The models proposed (a five-tier functional model spread over the traditional three-tier physical one) provide a very clear image of the role of the three main classes of middleware. In my opinion, this element of the book justifies the cost of the purchase many times over all by itself. The inevitable and inexorable evolution of the client / server model is well documented and the conclusions are well justified. The newest model is very pertinent to the issues being faced by architects everywhere and does a much better job of separating the data locality issues from the similar issues related to the granularity and coupling of the application logic in modern

computing environments. The role of the Internet technologies in enabling and driving this evolution is clearly described. The authors also provide a background that will allow the reader to follow the future evolution of the model and the eventual introduction of still more functional layers with ease. The paradigm is, indeed, that powerful. The book manages to address two disparate audiences very well. First, readers such as myself looking for design and management background material on the topic are very well served. At the same time, practitioners will find very comprehensive summaries of much of the current state of knowledge in the field. There are supplemental reading lists which will lead the reader to more focused discussions of any specific topics that may require further detail. I would expect this book to prove to be very influential. It challenges several established notions about client / server architectures which have led many shops to struggle with their implementations. The proposed model, properly applied, should lead to more robust designs that will prove to be more scaleable and adaptable than many we have seen. Highly Recommended!

0 von 0 Kunden fanden die folgende Rezension hilfreich. An outstanding overview of performance engineering. Von Ein Kunde This book should be required reading for all professional software systems engineers and software development engineers. It is an outstanding overview of the performance engineering discipline as it applies to client/server architectures. I am a practicing systems performance engineer with 20 years experience--10 in development and the last 10 in systems engineering. Many of the things I've learned through "experience" are included in this book. It is definitely worth the money. There is only one area which is not addressed in the book--how to apply SPE to one-of-a-kind complex multiprocessing/multitasking shared resource systems development which are without precedent (as far as I know this is not addressed anywhere in the literature). Such systems do not lend themselves to the preassigned quantitative software budgets required by the literal application of SPE. But because I am major proponent of designing in flexibility and performance all the other SPE principles certainly apply. Again, I highly recommend this book.

Kurzbeschreibung "Chris is one of the industry's most important thinkers on database design ...I would strongly recommend this book to readers trying to get past the buzzwords and focus on what really makes a difference in achieving high-performance distributed systems." --David Stodder Editor-in-Chief, Database Programming Design Performance is not simply a matter of tuning the code or the computing environment--it starts with designing performance into the application from the outset, and spans all phases of the system life cycle. Drawing on his 25 years of experience, Chris Loosley explains the principles of software performance engineering and applies them to all facets of distributed systems architecture and design. Along the way, he summarizes his conclusions in over 250 useful, easily referenced guidelines. And he covers all the key topics, with chapters on Middleware, Architecture, Design, Tools, Databases, Replication, Warehousing, and Transaction Monitors. Loosley's conclusions about the architecture and design of enterprise systems challenge many current middleware trends. Applying the performance principles, Loosley explains why the key to creating truly scalable distributed systems is to decompose complex business applications into multitransaction workflows, and to use asynchronous data replication, parallel processing, and batching techniques. Contributors Sid Adelman, Sid Adelman and Associates Nagraj Alur, DataBase Associates International Charles Brett, C3B Consulting Tom Cushing, Advanced Computer Services Mike Ferguson, DataBase Associates International John Kneiling, DataBase Associates International David Linthicum, Ernst Young Alejandro Mimo, DataBase Associates International Neal Nelson, Neal Nelson and Associates George Peters, MCI Systemhouse Colin White, DataBase Associates International Paul Winsberg, DataBase Associates International Synopsis "Chris is one of the industry's most important thinkers on database design ...I would strongly recommend this book to readers trying to get past the buzzwords and focus on what really makes a difference in achieving high-performance distributed systems." --David Stodder Editor-in-Chief, Database Programming Design Performance is not simply a matter of tuning the code or the computing environment--it starts with designing performance into the application from the outset, and spans all phases of the system life cycle. Drawing on his 25 years of experience, Chris Loosley explains the principles of software performance engineering and applies them to all facets of distributed systems architecture and design. Along the way, he summarizes his conclusions in over 250 useful, easily referenced guidelines. And he covers all the key topics, with chapters on Middleware, Architecture, Design, Tools, Databases, Replication, Warehousing, and Transaction Monitors. Loosley's conclusions about the architecture and design of enterprise systems challenge many current middleware trends. Applying the performance principles, Loosley explains why the key to creating truly scalable distributed systems is to decompose complex business applications into multitransaction workflows, and to use asynchronous data replication, parallel processing, and batching techniques. Contributors Sid Adelman, Sid Adelman and Associates Nagraj Alur, DataBase Associates International Charles Brett, C3B Consulting Tom Cushing, Advanced Computer Services Mike Ferguson, DataBase Associates International John Kneiling, DataBase Associates International David Linthicum, Ernst Young Alejandro Mimo, DataBase Associates International Neal Nelson, Neal Nelson and Associates George Peters, MCI Systemhouse Colin White, DataBase Associates International Paul Winsberg, DataBase Associates International Autorenkommentar All about performance, for those

who don't specialize in it After working with enterprise systems for over 25 years, I see performance as a fundamental cornerstone of application quality. Unfortunately, performance is also one area of software design that is frequently misunderstood, forgotten, ignored, or postponed, often with disastrous results. We cannot hope to achieve and maintain acceptable levels of application performance by simply tuning the code or tweaking the computing environment. It requires an integrated approach--Software Performance Engineering--that spans all phases of the system life cycle. The new technologies of distributed computing do nothing to diminish this need; rather, they reinforce it by offering many new possibilities for performance problems. Therefore, the book has three aims: 1. To explain clearly the factors that determine application performance and the basic principles of all performance-related thinking; 2. To draw conclusions and state useful guidelines that show how to apply those principles systematically to design, develop, and deploy effective applications for the distributed enterprise; 3. To do so using language and examples that are understandable and interesting to any professional working with today's information processing technologies. If our book is read only by performance specialists or theoreticians, we have failed. To clarify our perspective, we thought about calling the book "Designing and Developing Information Processing Applications that Perform Acceptably in an Enterprise-Wide Distributed Computing Environment," but publishers have performance standards too! I will read all e-mail and try to respond to any questions. Stay tuned for details of a web-site, too.