Scalable Transactions For Scalable Distributed Database

Distributed and Cloud Computing: From Parallel Processing to the Internet of Things offers complete coverage of modern distributed computing technology including clusters, the grid, service-oriented architecture, massively parallel processors, peer-to-peer networking, and cloud computing. It is the first modern, up-to-date distributed systems textbook; it explains how to create high-performance, scalable, reliable systems, exposing the design principles, architecture, and innovative applications of parallel, distributed, and cloud computing systems. Topics covered by this book include: facilitating management, debugging, migration, and disaster recovery through virtualization; clustered systems for research or ecommerce applications; designing systems as web services; and social networking systems using peer-to-peer computing. The principles of cloud computing are discussed using examples from open-source and commercial applications, along with case studies from the leading distributed computing vendors such as Amazon, Microsoft, and Google. Each chapter includes exercises and further reading, with lecture slides and more available online. This book will be ideal for students taking a distributed systems or distributed computing class, as well as for professional system designers and engineers looking for a reference to the latest distributed technologies including cloud, P2P and grid computing. Complete coverage of modern distributed computing technology including clusters, the grid, service-oriented architecture, massively parallel processors, peer-to-peer networking, and cloud computing Includes case studies from the leading distributed computing vendors: Amazon, Microsoft, Google, and more Explains how to use virtualization to facilitate management, debugging, migration, and disaster recovery Designed for undergraduate or graduate students taking a distributed systems course—each chapter includes exercises and further reading, with lecture slides and more available online

This book is a spin-off of a by-invitation-only workshop on self- properties in complex systems held in summer 2004 in Bertinoro, Italy. The workshop aimed to identify the conceptual and practical foundations for modeling, analyzing, and achieving self-* properties in distributed and networked systems. Based on the discussions at the workshop, papers were solicited from workshop participants and invited from leading researchers in the field. Besides presenting sound research results, the papers also present visionary statements, thought-provoking ideas, and exploratory results. The 27 carefully reviewed revised full papers, presented together with a motivating introduction and overview, are organized in topical sections on self-organization, self-awareness, self-awareness versus self-organization, supporting self-properties, and peer-to-peer algorithms.*

In the race to compete in today's fast-moving markets, large enterprises are busy adopting new technologies for creating new products, processes, and business models. But one obstacle on the road to digital transformation is placing too much emphasis on technology, and not enough on the types of processes technology enables. What if different lines of business could build their own services and applications—and decision-making was distributed rather than centralized? This report explores the concept of a digital business platform as a way of empowering individual business sectors to act on data in real time. Much innovation in a digital enterprise will increasingly happen at the edge, whether it involves business users (from marketers to data scientists) or IoT devices. To facilitate the process, your core IT team can provide these sectors with the digital tools they need to innovate quickly. This report explores: Key cultural and organizational changes for developing business capabilities through cross-functional product teams A platform for integrating applications, data sources, business partners, clients, mobile apps, social networks, and IoT devices Creating internal API programs for building innovative edge services in low-code or no-code environments Tools including Integration Platform as a Service, Application Platform as a Service, and Integration Software as a Service The challenge of integrating microservices and serverless architectures Event-driven architectures for processing and reacting to events in real time You'll also learn about a complete pervasive integration solution as a core component of a digital business platform to serve every audience in your organization.

This is the only book to cover infrastructure aspects of sensor networks in a comprehensive fashion. The only other books on sensor networks do not cover this topic or do so only superficially as part of a less-focussed multi-authored treatment.

Multimedia Information Retrieval and Management

A FRAMEWORK FOR SCALABLE DISTRIBUTED JOB PROCESSING WITH DYNAMIC LOAD BALANCING USING DECENTRALIZED APPROACH

From Parallel Processing to the Internet of Things

The Big Ideas Behind Reliable, Scalable, and Maintainable Systems

Designing for Scalability with Erlang/OTP

ACM/IFIP/USENIX International Middleware Conference, Toronto, Canada, October 18-20, 2004, Proceedings

This book constitutes the thoroughly refereed post-conference proceedings of the 4th IFIP TC2 Central and East European Conference on Software Engineering Techniques, CEE-SET 2009, held in Krakow, Poland, in October 2009. The 19 revised full papers presented were carefully reviewed and selected from 63 submissions. The papers are organized in topical sections on software architectures and development; modelling and formal methods in software development; measurements, testing, and quality of software.

AN ESSENTIAL GUIDE TO USING BLOCKCHAIN TO PROVIDE FLEXIBILITY, COST-SAVINGS, AND SECURITY TO DATA MANAGEMENT, DATA ANALYSIS, AND INFORMATION SHARING Blockchain for Distributed Systems Security contains a description of the properties that underpin the formal foundations of Blockchain technologies and explores the practical issues for deployment in cloud and Internet of Things (IoT) platforms. The authors—noted experts in the field—present security and privacy issues that must be addressed for Blockchain technologies to be adopted for civilian and military domains. The book covers a range of topics including data provenance in cloud storage, secure IoT models, auditing architecture, and empirical validation of permissioned Blockchain platforms. The book's security and privacy analysis helps with an understanding of the basics of Blockchain and it explores the quantifying impact of the new attack surfaces introduced by Blockchain technologies and platforms. In addition, the book contains relevant and current updates on the topic. This important resource: Provides an overview of Blockchain-based secure data management and storage for cloud and IoT Covers cutting-edge research findings on topics including invariant-based supply chain protection, information sharing framework, and trust worthy information federation Addresses security and privacy concerns in Blockchain in key areas, such as preventing digital currency miners from launching attacks against mining pools, empirical analysis of the attack surface of Blockchain, and more Written for researchers and experts in computer science and engineering, Blockchain for Distributed Systems Security contains the most recent information and academic research to provide an understanding of the application of Blockchain technology.

Tim Ewald, COM columnist for DOC Magazine, explains how COM+ works, and then sets out specific rules intended as concrete guidelines to help developers build COM+ systems.

With the advent of the Internet and Internet-connected devices, modern applications can experience very rapid growth of users from all parts of the world. A growing user base leads to greater usage and large data sizes, so scalable database systems capable of handling the great demands are critical for applications. With the emergence of cloud computing, a major movement in the industry, modern applications depend on distributed data stores for their scalable data management solutions. Many large-scale applications utilize NoSQL systems, such as distributed key-value stores, for their scalability and availability properties over traditional relational database systems. By simplifying the design and interface, NoSQL systems can provide high scalability and performance for large data sets and high volume workloads. However, to provide such benefits, NoSQL systems sacrifice traditional consistency models and support for transactions typically available in database systems. Without transaction semantics, it is harder for developers to reason about the correctness of the interactions with the data. Therefore, it is important to support transactions for distributed database systems without sacrificing scalability. In this thesis, I present new techniques for scalable transactions for scalable database systems. Distributed data stores need scalable transactions to take advantage of cloud computing, and to meet the demands of modern applications. Traditional techniques for transactions may not be appropriate in a large, distributed environment, so in this thesis, I describe new techniques for distributed transactions, without having to sacrifice traditional semantics or scalability. I discuss three facets to improving transaction scalability and support in distributed database systems. First, I describe a new transaction commit protocol that reduces the response times for distributed transactions. Second, I propose a new transaction programming model that allows developers to better deal with the unexpected behavior of distributed transactions. Lastly, I present a new scalable view maintenance algorithm for convergent join views. Together, the new techniques in this thesis contribute to providing scalable transactions for modern, distributed database systems.

Middleware 2004

SCI: Scalable Coherent Interface

Parallel and Distributed Processing and Applications

Data Management in the Cloud

Technical Report Number 697

Patterns and Paradigms for Scalable, Reliable Services

Data is at the center of many challenges in system design today. Difficult issues need to be figured out, such as scalability, consistency, reliability, efficiency, and maintainability. In addition, we have an overwhelming variety of tools, including relational databases, NoSQL datastores, stream or batch processors, and message brokers. What are the right choices for your application? How do you make sense of all these buzzwords? In this practical and comprehensive guide, author Martin Kleppmann helps you navigate this diverse landscape by examining the pros and cons of various technologies for processing and storing data. Software keeps changing, but the fundamental principles remain the same. With this book, software engineers and architects will learn how to apply those ideas in practice, and how to make full use of data in modern applications. Peer under the hood of the systems you already use, and learn how to use and operate them more effectively Make informed decisions by identifying the strengths and weaknesses of different tools Navigate the trade-offs around consistency, scalability, fault tolerance, and complexity Understand the distributed systems research upon which modern databases are built Peek behind the scenes of major online services, and learn from their architectures

The race to compete in today's fast-moving markets, large enterprises are busy adopting new technologies for creating new products, processes, and business models. But one obstacle on the road to digital transformation is placing too much emphasis on technology, and not enough on the types of processes technology enables. What if different lines of business could build their own services and applications-and decision-making was distributed rather than centralized? This report explores the concept of a digital business platform as a way of empowering individual business sectors to act on data in real time. Much innovation in a digital enterprise will increasingly happen at the edge, whether it involves business users (from marketers to data scientists) or IoT devices. To facilitate the process, your core IT team can provide these sectors with the digital tools they need to innovate quickly

The LNCS journal Transactions on Large-Scale Data- and Knowledge-Centered Systems focuses on data management, knowledge discovery, and knowledge processing, which are core and hot topics in computer science. Since the 1990s, the Internet has become the main driving force behind application development in all domains. An increase in the demand for resource sharing (e.g., computing resources, services, metadata, data sources) across different sites connected through networks has led to an evolution of data- and knowledge management systems from centralized systems to decentralized systems enabling large-scale distributed applications providing high scalability. This, the 48th issue of Transactions on Large-Scale Data- and Knowledge-Centered Systems, contains 8 invited papers dedicated to the memory of Prof. Dr. Roland Wagner. The topics covered include distributed database systems, NewSQL, scalable transaction management, strong consistency, caches, data warehouse, ETL, reinforcement learning, stochastic approximation, multi-agent systems, ontology, model-driven development, organisational modelling, digital government, new institutional economics and data governance.

Principles of Transaction Processing is a comprehensive guide to developing applications, designing systems, and evaluating engineering products. The book provides detailed discussions of the internal workings of transaction processing systems, and it discusses how these systems work and how best to utilize them. It covers the architecture of Web Application Servers and transactional communication paradigms. The book is divided into 11 chapters, which cover the following: Overview of transaction processing application and system structure Software abstractions found in transaction processing systems Architecture of multitier applications and the functions of transactional middleware and database servers Queued transaction processing and its internals, with IBM's Websphere MQ and Oracle's Stream AQ as examples Business process management and its mechanisms Description of the two-phase locking function, B-tree locking and multigranularity locking used in SQL database systems and nested transaction locking System recovery and its failures Two-phase commit protocol Comparison between the tradeoffs of replicating servers versus replication resources Transactional middleware products and standards Future trends, such as cloud computing platforms, composing scalable systems using distributed computing components, the use of flash storage to replace disks and data streams from sensor devices as a source of transaction requests. The text meets the needs of systems professionals, such as IT application programmers who construct TP applications, application analysts, and product developers. The book will also be invaluable to students and novices in application programming. Complete revision of the classic "non mathematical" transaction processing reference for systems professionals. Updated to focus on the needs of transaction processing via the Internet-- the main focus of business data processing investments, via web application servers, SOA, and important new TP standards. Retains the practical, non-mathematical, but thorough conceptual basis of the first edition.

Building Scalable Applications

Technicle Report Number 696

Special Issue on Data and Knowledge Management in Grid and PSP Systems

Conceptual and Practical Foundations

Implement Robust, Fault-Tolerant Systems

Transactional COM+

Cloud computing has emerged as a successful paradigm of service-oriented computing and has revolutionized the way computing infrastructure is used. This success has seen a proliferation in the number of applications that are being deployed in various cloud platforms. There has also been an increase in the scale of the data generated as well as consumed by such applications. Scalable database management systems form a critical part of the cloud infrastructure. The attempt to address the challenges posed by the management of big data has led to a plethora of systems. This book aims to clarify some of the important concepts in the design space of scalable data management in cloud computing infrastructures. Some of the questions that this book aims to answer are: the appropriate systems for a specific set of application requirements, the research challenges in data management for the cloud, and what is novel in the cloud for database researchers? We also aim to address one basic question: whether cloud computing poses new challenges in scalable data management or it is just a reincarnation of old problems? We provide a comprehensive background study of state-of-the-art systems for scalable data management and analysis. We also identify important aspects in the design of different systems and the applicability and scope of these systems. A thorough understanding of current solutions and a precise characterization of the design space are essential for clearing the "cloudy skies of data management" and ensuring the success of DBMSs in the cloud, thus emulating the success enjoyed by relational databases in traditional enterprise settings. Table of Contents: Introduction / Distributed Data Management / Cloud Data Management: Early Trends / Transactions on Co-located Data / Transactions on Distributed Data / Multi-tenant Database Systems / Concluding Remarks

Designing a New Class of Distributed Systems closely examines the Distributed Intelligent Managed Element (DIME) Computing Model, a new model for distributed systems, and provides a guide to implementing Distributed Managed Workflows with High Reliability, Availability, Performance and Security. The book also explores the viability of self-optimizing, self-monitoring autonomous DIME-based computing systems. Designing a New Class of Distributed Systems is designed for practitioners as a reference guide for innovative distributed systems design. Researchers working in a related field will also find this book valuable.

Explains fault tolerance in clear terms, with concrete examples drawn from real-world settings Highly practical focus aimed at building "mission-critical" networked applications that remain secure

The LNCS journal Transactions on Large-Scale Data- and Knowledge-Centered Systems focuses on data management, knowledge discovery, and knowledge processing, which are core and hot topics in computer science. Since the 1990s, the Internet has become the main driving force behind application development in all domains. An increase in the demand for resource sharing across different sites connected through networks has led to an evolution of data- and knowledge-management systems from centralized systems to decentralized systems enabling large-scale distributed applications providing high scalability. Current decentralized systems still focus on data and knowledge as their main resource. Feasibility of these systems relies basically on P2P (peer-to-peer) techniques and the support of agent systems with scaling and decentralized control. Synergy between Grids, P2P systems, and agent technologies is the key to data- and knowledge-centered systems in large-scale environments. This, the third issue of Transactions on Large-Scale Data- and Knowledge-Centered Systems, contains two kinds of papers: Firstly, a selection of the best papers from the third International Conference on Data Management in Grid and Peer-to-Peer Systems, Globe 2010, and secondly, a selection of 6 papers from the 18 papers submitted in response to the call for papers for this issue. The topics covered by this special issue include replication, the semantic web, information retrieval, data storage, source selection, and large-scale distributed applications.

Scalable Infrastructure for Distributed Sensor Networks

Third International Symposium, ISPA 2005, Nanjing, China, November 2-5, 2005, Proceedings

Production-Ready Microservices

Special Issue In Memory of Univ. Prof. Dr. Roland Wagner

Building Scalable and High-performance Java Web Applications Using J2EE Technology

6th International Conference, PPAM 2005, Poznan, Poland, September 11-14, 2005, Revised Selected Papers

Middleware components are becoming increasingly important as applications share computational resources in large distributed environments, such as web services, high-end clusters with ever larger number of processors, computational grids and an increasingly large server farms. One of the main challenges in such environments is to achieve scalability of synchronization. Another challenge is posed by requirement for shared resources with a need for QoS and real-time support. In general, concurrency services arbitrate resource requests in distributed systems. But concurrency protocols currently lack scalability and support for service differentiation based on QoS requirements. Adding such guarantees enables resource sharing and computing with distributed objects in systems with a large number of nodes and supporting a wide range of QoS metrics. The objective of this thesis is to enhance middleware services to provide scalability of synchronization and to support service differentiation based on priorities. We have designed and implemented middleware protocols in support of these objectives. Its essence is a novel, peer-to-peer, fully decentralized protocol for multi-mode hierarchical locking, which is applicable to transaction-style processing and distributed agreement. We discuss the design and implementation details of the protocols and demonstrate high scalability combined with low response times in high-performance cluster environments as well as TCP/IP networks when compared to a prior protocol for distributed synchronization. The prioritized version of the protocol is shown to offer differentiated response times to real-time applications with support for protocols to bound priority inversion such as PCEP and PIP. Our approach was originally motivated by CORBA concurrency services. Beyond CORBA, its principles are shown to provide benefits to general distributed concurrency services and transaction models. Besides its technical strengths, our approach is intriguin.

This book constitutes the refereed proceedings of the ACM/IFIP/USENIX International Conference on Distributed Systems Platforms, Middleware 2004, held in Toronto, Canada in October 2004. The 25 revised full papers presented together with an invited paper were carefully reviewed and selected from a total of 194 submissions. The papers are organized in topical sections on peer-to-peer computing; routing protocols and overlay; middleware for replication and overlay; middleware for replication and transactions; publish/subscribe systems; Web services: composition, integration, and interoperabilty; middleware for mobility; application servers, enterprise computing, and software engineering.

The unprecedented scale at which data is both produced and consumed today has generated a large demand for scalable data management solutions facilitating fast access from all over the world. As one consequence, a plethora of non-relational, distributed NoSQL database systems have risen in recent years and today's data management system landscape has thus become somewhat hard to overlook. As another consequence, complex polyglot designs and elaborate schemes for data distribution and delivery have become the norm for building applications that connect users and organizations across the globe – but choosing the right combination of systems for a given use case has become increasingly difficult as well. To help practitioners stay on top of that challenge, this book presents a comprehensive overview and classification of the current system landscape in cloud data management as well as a survey of the state-of-the-art approaches for efficient data distribution and delivery to end-user devices. The topics covered thus range from NoSQL storage systems and polyglot architectures (backend) over distributed transactions and Web caching (network) to data access and rendering performance in the client (end-user). By distinguishing popular data management systems by data model, consistency guarantees, and other dimensions of interest, this book provides an abstract framework for reasoning about the overall design space and the individual positions claimed by each of the systems therein. Building on this classification, this book further presents an application-driven decision guidance tool that breaks the process of choosing a set of viable system candidates for a given application scenario down into a straightforward decision tree.

The fourth edition of this classic textbook provides major updates. This edition has completely new chapters on Big Data Platforms (distributed storage systems, MapReduce, Spark, data stream processing, graph analytics) and on NoSQL, NewSQL and polystore systems. It also includes an updated web data management chapter that includes RDF and semantic web discussion, an integrated database integration chapter focusing both on schema integration and querying over these systems. The peer-to-peer computing chapter has been updated with a discussion of blockchains. The chapters that describe classical distributed and parallel database technology have all been updated. The new edition covers the breadth and depth of the field from a modern viewpoint. Graduate students, as well as senior undergraduate students studying computer science and other related fields will use this book as a primary textbook. Researchers working in computer science will also find this textbook useful. This textbook has a companion web site that includes background information on relational database fundamentals, query processing, transaction management, and computer networks for those who might need this background. The web site also includes all the figures and presentation slides as well as solutions to exercises (restricted to instructors).

Fast and Scalable Cloud Data Management

Transactions on Large-Scale Data- and Knowledge-Centered Systems III

Distributed and Cloud Computing

Designing Distributed Systems

Parallel Processing and Applied Mathematics

One of the biggest challenges for organizations that have adopted microservice architecture is the lack of architectural, operational, and organizational standardization. After splitting a monolithic application or building a microservice ecosystem from scratch, many engineers are left wondering what's next. In this practical book, author Susan Fowler presents a set of microservice standards in depth, drawing from her experience standardizing over a thousand microservices at Uber. You'll learn how to design microservices that are stable, reliable, scalable, fault tolerant, performant, monitored, documented, and prepared for any catastrophe. Explore production-readiness standards, including: Stability and Reliability: develop, deploy, introduce, and deprecate microservices; protect against dependency failures Scalability and Performance: learn essential components for achieving greater microservice efficiency Fault Tolerance and Catastrophe Preparedness: ensure availability by actively pushing microservices to fail in real time Monitoring: learn how to monitor, log, and display key metrics; establish alerting and on-call procedures Documentation and Understanding: mitigate tradeoffs that come with microservice adoption, including organizational sprawl and technical debt

Scalable Coherent Interface (SCI) is an innovative interconnect standard (ANSI/IEEE Std 1596-1992) addressing the high-performance computing and networking domain. This book describes in depth one specific application of SCI: its use as a high-speed interconnection network (often called a system area network, SAN) for compute clusters built from commodity workstation nodes. The editors and authors, coming from both academia and industry, have been instrumental in the SCI standardization process, the development and deployment of SCI adapter cards, switches, fully integrated clusters, and software systems, and are closely involved in various research projects on this important interconnect. This thoroughly cross-reviewed state-of-the-art survey covers the complete hardware/software spectrum of SCI clusters, from the major concepts of SCI, through SCI hardware, networking, and low-level software issues, various programming models and environments, up to tools and application experiences.

Everything you ever wanted to know about multimedia retrieval and management. This comprehensive book offers a full picture of the cutting-edge technologies necessary for a profound introduction to the field. Leading experts also cover a broad range of practical applications.

This volume comprises the proceedings of the 6th International Conference on Parallel Processing and Applied Mathematics - PPAM 2005, which was held in Poznan, the industrial, academic and cultural center in the western part of Poland, during September 11–14, 2005.

Technological Fundamentals and Applications

Technologies, Web Services, and Applications

Compiler Optimizations for Scalable Parallel Systems

Self-star Properties in Complex Information Systems

Transactions on Large-Scale Data- and Knowledge-Centered Systems XLVIII

Blockchain for Distributed Systems Security

This is the book for Gophers who want to learn how to build distributed systems. You know the basics of Go and are eager to put your knowledge to work. Build distributed services that are highly available, resilient, and scalable. This book is just what you need to apply Go to real-world situations. Level up your engineering skills today. Take your Go skills to the next level by learning how to design, develop, and deploy a distributed service. Start from the bare essentials of storage handling, then work your way through networking a client and server, and finally to distributing server instances, deployment, and testing. All this will make coding in your day job or side projects easier, faster, and more fun. Create your own distributed services and contribute to open source projects. Build networked, secure clients and servers with gRPC. Gain insights into your systems and debug issues with observable services instrumented with metrics, logs, and traces. Operate your own Certificate Authority to authenticate internal web services with TLS. Automatically handle when nodes are added or removed to your cluster with service discovery. Coordinate distributed systems with replicated state machines powered by the Raft consensus algorithm. Lay out your applications and libraries to be modular and easy to maintain. Write CLIs to configure and run your applications. Run your distributed system locally and deploy to the cloud with Kubernetes. Test and benchmark your applications to ensure they're correct and fast. Dive into writing Go and join the hundreds of thousands who are using it to build software for the real world. What You Need: Go 1.13+ and Kubernetes 1.16+

A collection of seven long articles, this book comprehensively discusses significant projects in scalable computing in various research organizations around the world. It represents the quantitative and qualitative growth of work in the area. Contents:Experiences with Shared Virtual Memory on System Area Network Clusters: System Simulation, Implementation, and EmulationAverage-Case Scalability Analysis of Parallel ComputationsParallel IO Prefetching and CachingA C++/Tuple-Lock Implementation for Distributed ObjectsStatic Data Allocation and Load Balancing Techniques for Heterogeneous SystemsBuilding a Global Object Space for Supporting Single System Image on a ClusterA Computation-Centric Multilocation Consistency Model for Shared Memory Readership: Graduate students, academics and researchers in supercomputing and computer engineering. Keywords:Clusters;Data Allocation;Global Object Space;Load Balancing;Location Consistency;Scalability Analysis;Shared Virtual Memory;Tuple Locks;Work Stealing

Modern cloud computing systems usually provide a highly scalable and fault-tolerant data store that sacrifices other features. Often, these systems may not support transactions at all or else restrict transactions to one data item each. Recently techniques to support multi-item transactions in these types of systems have been successfully developed but have focused on transactions across homogeneous data stores. However, applications often need to store different data in different storage systems perhaps for legacy or interoperability reasons. We propose an approach that enables multi-item transactions across multiple heterogeneous data stores using only a minimal set of commonly implemented features such as single item consistency, conditional updates, and the ability to store additional metadata. We define an client-coordinate transaction commitment protocol that does not rely on a central coordinating infrastructure. We implement this as a Java library, we call Cherry Garcia (CG), that supports data store abstractions to Windows Azure Storage (WAS), Google Cloud Storage (GCS) and our own high-performance key-value store called Tora.

This integrated collection covers a range of parallelization platforms, concurrent programming frameworks and machine learning settings, with case studies.

Scalable Distributed Concurrency Protocol with Priority Support

Designing a New Class of Distributed Systems

Understanding Distributed Systems

Advances in Software Engineering Techniques

Design and secure highly available, cost-effective data streaming applications with Amazon Kinesis

Scaling Up Machine Learning

Comprehensively discusses significant projects in scalable computing in various research organizations around the world.

Scalable Transactions for Scalable Distributed Database Systems

Scaling Java enterprise applications beyond just programming techniques--this is the next level. This volume covers all the technologies Java developers need to build scalable, high-performance Web applications. The book also covers servlet-based session management, EJB application logic, database design and integration, and more.

Explore Kinesis managed services such as Kinesis Data Streams, Kinesis Data Analytics, Kinesis Data Firehose, and Kinesis Video Streams with the help of practical use cases Key FeaturesGet well versed with the capabilities of Amazon KinesisExplore the monitoring, scaling, security, and deployment patterns of various Amazon Kinesis servicesLearn how other Amazon Web Services and third-party applications such as Splunk can be used as destinations for Kinesis dataBook Description Amazon Kinesis is a collection of secure, serverless, durable, and highly available purpose-built data streaming services. This data streaming service provides APIs and client SDKs that enable you to produce and consume data at scale. Scalable Data Streaming with Amazon Kinesis begins with a quick overview of the core concepts of data streams, along with the essentials of the AWS Kinesis landscape. You'll then explore the requirements of the use case shown through the book to help you get started and cover the key pain points encountered in the data stream life cycle. As you advance, you'll get to grips with the architectural components of Kinesis, understand how they are configured to build data pipelines, and delve into the applications that connect to them for consumption and processing. You'll also build a Kinesis data pipeline from scratch and learn how to implement and apply practical solutions. Moving on, you'll learn how to configure Kinesis on a cloud platform. Finally, you'll learn how other AWS services can be integrated into Kinesis. These services include Redshift, Dynamo Database, AWS S3, Elastic Search, and third-party applications such as Splunk. By the end of this AWS book, you'll be able to build and deploy your own Kinesis data pipelines with Kinesis Data Streams (KDS), Kinesis Data Firehose (KFH), Kinesis Video Streams (KVS), and Kinesis Data Analytics (KDA). What you will learnGet to grips with data streams, decoupled design, and real-time stream processingUnderstand the properties of KFH that differentiate it from other Kinesis servicesMonitor and scale KDS using CloudWatch metricsSecure KDA with identity and access management (IAM)Deploy KVS as infrastructure as code (IaC)Integrate services such as Redshift, Dynamo Database, and Splunk into KinesisWho this book is for This book is for solutions architects, developers, system administrators, data engineers, and data scientists looking to evaluate and choose the most performant, secure, scalable, and cost-effective data streaming technology to overcome their data ingestion and processing challenges on AWS. Prior knowledge of cloud architectures on AWS, data streaming technologies, and architectures is expected.

Principles of Transaction Processing

Foundations of Scalable Systems

Reliable Distributed Systems

Languages, Compilation Techniques, and Run Time Systems

Architecture and Software for High-Performance Compute Clusters

Annual Review of Scalable Computing

Scalable parallel systems or, more generally, distributed memory systems offer a challenging model of computing and pose fascinating problems regarding compiler optimization, ranging from language design to run time systems. Research in this area is foundational to many challenges from memory hierarchy optimizations to communication optimization. This unique, handbook-like monograph assesses the state of the art in the area in a systematic and comprehensive way. The 21 coherent chapters by leading researchers provide complete and competent coverage of all relevant aspects of compiler optimization for scalable parallel systems. The book is divided into five parts on languages, analysis, communication optimizations, code generation, and run time systems. This book will serve as a landmark source for education, information, and reference to students, practitioners, professionals, and researchers interested in updating their knowledge about or active in parallel computing.

If you need to build a scalable, fault tolerant system with requirements for high availability, discover why the Erlang/OTP platform stands out for the breadth, depth, and consistency of its features. This hands-on guide demonstrates how to use the Erlang programming language and its OTP framework of reusable libraries, tools, and design principles to develop complex commercial-grade systems that simply cannot fail. In the first part of the book, you ' ll learn how to design and implement process behaviors and supervision trees with Erlang/OTP, and bundle them into standalone nodes. The second part addresses reliability, scalability, and high availability in your overall system design. If you ' re familiar with Erlang, this book will help you understand the design choices and trade-offs necessary to keep your system running. Explore OTP ' s building blocks: the Erlang language, tools and libraries collection, and its abstract principles and design rules Dive into the fundamentals of OTP reusable frameworks: the Erlang process structures OTP uses for behaviors Understand how OTP behaviors support client-server structures, finite state machine patterns, event handling, and runtime/code integration Write your own behaviors and special processes Use OTP ' s tools, techniques, and architectures to handle deployment, monitoring, and operations

Distributed databases, which rely on redundant and distributed storage across multiple servers, are able to provide mission-critical data management services at large scale. Parallelism is the key to the scalability of distributed databases, but concurrent queries having conflicts may block or abort each other when strong consistency is enforced using rigorous concurrency control protocols. This thesis studies the techniques of building scalable distributed databases under strong consistency guarantees even in the face of high contention workloads. The techniques proposed in this thesis share a common idea, conflict mitigation, meaning mitigating conflicts by rescheduling operations in the concurrency control in the first place instead of resolving contending conflicts. Using this idea, concurrent queries under conflicts can be executed with high parallelism. This thesis explores this idea on both databases that support serializable ACID (atomic, consistency, isolation, durability) transactions, and eventually consistent NoSQL systems. First, the epoch-based concurrency control (ECC) technique is proposed in ALOHA-KV, a new distributed key-value store that supports high performance read-only and write-only distributed transactions. ECC demonstrates that concurrent serializable distributed transactions can be processed in parallel with low overhead even under high contention. With ECC, a new atomic commitment protocol is developed that only requires amortized one round trip for a distributed write-only transaction to commit in the absence of failures. Second, a novel paradigm of serializable distributed transaction processing is developed to extend ECC with read-write transaction processing support. This paradigm uses a newly proposed database operator, functors, which is a placeholder for the value of a key, which can be computed asynchronously in parallel with other functor computations of the same or other transactions. Functor-enabled ECC achieves more fine-grained concurrency control than transaction level concurrency control, and it never aborts transactions due to read-write or write-write conflicts but allows transactions to fail due to logic errors or constraint violations while guaranteeing serializability. Lastly, this thesis explores consistency in the eventually consistent system, Apache Cassandra, for an investigation of the consistency violation, referred to as "consistency spikes". This investigation shows that the consistency spikes exhibited by Cassandra are strongly correlated with garbage collection, particularly the "stop-the-world" phase in the Java virtual machine. Thus, delaying read operations arti cially at servers immediately after a garbage collection pause can virtually eliminate these spikes. All together, these techniques allow distributed databases to provide scalable and consistent storage service.

In many systems, scalability becomes the primary driver as the user base grows. Attractive features and high utility breed success, which brings more requests to handle and more data to manage. But organizations reach a tipping point when design decisions that made sense under light loads suddenly become technical debt. This practical book covers design approaches and technologies that make it possible to scale an application quickly and cost-effectively. Author Ian Gorton takes software architects and developers through the principles of foundational distributed systems. You'll explore the essential ingredients of scalable solutions, including replication, state management, load balancing, and caching. Specific chapters focus on the implications of scalability for databases, microservices, and event-based streaming systems. You will focus on: Foundations of scalable systems: Learn basic design principles of scalability, its costs, and architectural tradeoffs Designing scalable services: Dive into service design, caching, asynchronous messaging, serverless processing, and microservices Designing scalable data systems: Learn data system fundamentals, NoSQL databases, and eventual consistency versus strong consistency Designing scalable streaming systems: Explore stream processing systems and scalable event-driven processing

Scalable Transactions for Scalable Distributed Database Systems

Distributed Services with Go

Parallel and Distributed Approaches

Building Scalable and Consistent Distributed Databases Under Conflicts

4th IFIP TC 2 Central and East European Conference on Software Engineering Techniques, CEE-SET 2009, Krakow, Poland, October 12-14, 2009. Revised Selected Papers

Building Standardized Systems Across an Engineering Organization

Welcome to the proceedings of ISPA 2005 which was held in the city of Nanjing. Parallel computing has become a mainstream research area in computer science and the ISPA conference has become one of the premier forums for the presentation of new and exciting research on all aspects of parallel computing. We are pleased to present the proceedings for the 3rd International Symposium on Parallel and Distributed Processing and Applications (ISPA 2005), which comprises a collection of excellent technical papers, and keynote speeches. The papers accepted cover a wide range of exciting topics, including architectures, software, networking, and applications. The conference continues to grow and this year a record total of 968 manuscripts (including workshop submissions) were submitted for consideration by the Program Committee or workshops. From the 645 papers submitted to the main conference, the Program Committee selected only 90 long papers and 19 short papers in the program. Eight workshops complemented the outstanding paper sessions.

Learning to build distributed systems is hard, especially if they are large scale. It's not that there is a lack of information out there. You can find academic papers, engineering blogs, and even books on the subject. The problem is that the available information is spread out all over the place, and if you were to put it on a spectrum from theory to practice, you would find a lot of material at the two ends, but not much in the middle. That is why I decided to write a book to teach the fundamentals of distributed systems so that you don't have to spend countless hours scratching your head to understand how everything fits together. This is the guide I wished existed when I first started out, and it's based on my experience building large distributed systems that scale to millions of requests per second and billions of devices. If you develop the back-end of web or mobile applications (or would like to!), this book is for you. When building distributed systems, you need to be familiar with the network stack, data consistency models, scalability and reliability patterns, and much more. Although you can build applications without knowing any of that, you will end up spending hours debugging and re-designing their architecture, learning lessons that you could have acquired in a much faster and less painful way.

Principles of Distributed Database Systems

Scalable Data Streaming with Amazon Kinesis

Scalable Distributed Transactions Across Heterogeneous Stores

Designing Data-Intensive Applications