Register | Login | Logout | Contact Us

Java-Success.com

Industrial strength Java/JEE Career Companion to open more doors

search here … Go

Home | Java FAQs | 600+ Java Q&As | Career | Tutorials | Member | Why?

Can u Debug? | Java 8 ready? | Top X | Productivity Tools | Judging Experience?

Home › Interview › Pressed for time? Java/JEE Interview FAQs › Java Key Area Essentials › 6 Scaling your Java applications interview Q&As

6 Scaling your Java applications interview Q&As

Posted on September 8, 2014 by Arulkumaran Kumaraswamipillai — No Comments ↓

Q1. What is the difference between performance and scalability?

A1. The performance and scalability are two different things. For example, if you are in the business of transporting people in a van, the performance is all about utilizing more powerful engine to transporting your people quicker to their destination. Scalability is all about catering for increase in demand for such transportation as your business grows by either increasing the capacity of individual actors (e.g. bigger van from 8 seater to 12 seater) or adding more actors (e.g. from a single van to 3 vans).

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

open all | close all

⊞ Ice Breaker Interview
⊞ Core Java Interview (
⊞ JEE Interview Q&A (3
⊟ Pressed for time? Jav
⊞ Job Interview Ice B
⊞ FAQ Core Java Jol
⊞ FAQ JEE Job Inter
⊞ FAQ Java Web Ser
⊞ Java Application Ar
⊞ Hibernate Job Inter
⊞ Spring Job Intervie
⊟ Java Key Area Ess
♦ Design pattern
♥ Top 10 causes
♥♦ 01: 30+ Writir
♦ 12 Java desigr
♦ 18 Agile Devel
♦ 5 Ways to debi
♦ 9 Java Transac
♦ Monitoring/Pro
02: ♥♦ 13 Tips tc

Q2. What are the different types of scalability?

A2. Vertical Scaling: You can increase the seating capacity of a van or use more powerful vans to reduce the time it takes to reach the destination. In a computer term, increase CPU, memory, etc to increase the capacity or tune the code/ database to reduce the time it takes to process. This means we have just increased the capacity of each actor. You can also vertically scale an application via multi-threading or using non-blocking I/O.

Horizontal Scaling: In a horizontal scaling model, instead of increasing the seating capacity of each individual actor in the system, we simply add more actors to the system. This means more vans. In terms of the computers, adding more nodes and servers.

Q3. How will you scale your data store?

A3. The scalability of database is critical because data is often a shared resource, and it becomes the main contact point for nearly every web request. The most important question you have to ask when considering the scalability of your database is, “What kind of system am I working with?” Are you working with a read-heavy or a write-heavy system?

Scaling Reads: If your website is primarily a read-centric system, vertically scale your data store with a caching strategy that uses memory cache (e.g. ehcache) or a CDN (Content Delivery Newtork). You can also add more CPU/RAM/Disk to scale vertically.

Scaling Writes: If your website is primarily a write-heavy system, you want to think about using a horizontally scalable datastore such as MongoDB (NoSQL database), Riak, Cassandra or HBase. MongoDB is a NoSQL database with great features like replication and sharding built in. This allows you to scale your database to as many servers as you would like by distributing content among them. A database shard (“sharding”) is the phrase used to describe a horizontal partition in a database or search engine. The idea behind sharding is to split data among multiple machines while ensuring that the data is always accessed from the correct

15 Security key a
4 FAQ Performa
4 JEE Design Pa
5 Java Concurre
6 Scaling your Ja
8 Java memory i
⊞ OOP & FP Essentia
⊞ Code Quality Job I
⊞ SQL, XML, UML, JSO
⊞ Hadoop & BigData Int
⊞ Java Architecture Inte
⊞ Scala Interview Q&As
⊟ Spring, Hibernate, & I
⊟ Spring (18)
⊞ Spring boot (4)
⊞ Spring IO (1)
⊞ Spring JavaConf
01: ♥♦ 13 Spring
01b: ♦ 13 Spring
02: ► Spring DII
03: ♥♦ Spring DI
04 ♦ 17 Spring b
05: ♦ 9 Spring Be
06: ♥ Debugging
07: Debugging S
Spring loading p
⊟ Hibernate (13)
01: ♥♦ 15+ Hiber
01b: ♦ 15+ Hiber
02: Understandir
03: Identifying ar
04: Identifying ar
05: Debugging H
06: Hibernate Fir
07: Hibernate mi
08: Hibernate au
09: Hibernate en
10: Spring, Java
11: Hibernate de
12: Hibernate cu
⊞ AngularJS (2)

place. Since sharding spreads the database across multiple machines, the database programmer specifies explicit sharding rules to determine which machines any piece of data will be stored on. Sharding may also be referred to as horizontal scaling or horizontal partitioning. Oracle uses (RAC – Real Application Cluster) where small server blades are genned-in to an Oracle RAC cluster over a high-speed interconnect.

SQL Vs NoSQL

SQL Model:

The relational model takes data and store them in many normalized interrelated tables that contain rows and columns. Tables relate with each other through foreign keys. When looking up data, the desired information needs to be collected by joining many related tables and combined before it can be provided to the application.

NoSQL Model

NoSQL databases have a very different model. NoSQL databases have been built from the ground up to be distributed, scale-out technologies and therefore fit better with the highly distributed nature of the three-tier Internet architecture. A document-oriented NoSQL database takes the data you want to store and aggregates it into documents using the JSON format. Each JSON document can be thought of as an object to be used by your application. This might relate to data agregated from 10+ tables in an SQL model.

Q4. What is BigData?
A4. Big data is the term for a collection of data sets so large and complex that becomes very difficult to work with using most relational database management systems and desktop statistics and visualization packages, requiring instead “massively parallel software running on tens, hundreds, or even thousands of servers”. Apache™ Hadoop® is an open

⊞ Git & SVN (6)
⊞ JMeter (2)
⊞ JSF (2)
⊞ Maven (3)
⊞ Testing & Profiling/Sa
⊞ Other Interview Q&A f
⊞ ▶Free Java Interviev

16 Technical Key Areas

open all | close all
⊞ Best Practice (6)
⊞ Coding (26)
⊞ Concurrency (6)
⊞ Design Concepts (7)
⊞ Design Patterns (11)
⊞ Exception Handling (3
⊞ Java Debugging (21)
⊞ Judging Experience In
⊞ Low Latency (7)
⊞ Memory Managemen
⊞ Performance (13)
⊞ QoS (8)
⊞ Scalability (4)
⊞ SDLC (6)
⊞ Security (13)
⊞ Transaction Managen

80+ step by step Java Tutorials

open all | close all
⊞ Setting up Tutorial (6)
⊞ Tutorial - Diagnosis (2
⊞ Akka Tutorial (9)
⊞ Core Java Tutorials (2
⊞ Hadoop & Spark Tuto

source software project that enables the distributed processing of large data sets across clusters of commodity servers. It is designed to scale up from a single server to thousands of machines, with a very high degree of fault tolerance.

Hadoop uses MapReduce to understand and assign work to nodes in the cluster and HDFS(Hadoop Distributed File System), which is file system that spans all the nodes in a Hadoop cluster for data storage.

Q5. What are the general scaling practices for a medium size system in Java?
A5.

– Using non-blocking IO
– Favoring lock free algorithms and concurrent collection classes like *ConcurrentHashmap*.
– Vertical scaling — more CPU, RAM, etc.
– Caching data.
– Favor stateless idempotent methods.
– Using big JVM heaps.
– Using JMS — publish/subscribe model.
– Using resource pooling – e.g. database connection pooling, JMS connection factory pooling, thread pooling, etc.
– Using flyweight design pattern to minimize object creation.
– Tuning garbage collection to minimize pauses.

Q6. What are the general scaling practices for a large size system in Java?
A6.

– Use of RTSJ (Real Time Specification for Java) which addresses issues like

- During garbage collection all threads are blocked and the garbage collection time can expand to minutes. These huge latencies effectively limit memory which limits scalability.
- Increased garbage collection latencies make Java less useful for application that use heart beats, make

⊞ JEE Tutorials (19)
⊞ Scala Tutorials (1)
⊞ Spring & HIbernate Tu
⊞ Tools Tutorials (19)
⊞ Other Tutorials (45)

100+ Java pre-interview coding tests

open all | close all
⊞ Can you write code? (
⊞ ♦ Complete the given
⊞ Converting from A to I
⊞ Designing your classe
⊞ Java Data Structures
⊞ Passing the unit tests
⊞ What is wrong with th
⊞ Writing Code Home A
⊞ Written Test Core Jav
⊞ Written Test JEE (1)

How good are your .....?

open all | close all
⊞ Career Making Know-
⊞ Job Hunting & Resum

real-time trades, etc.

- Java supports a strict priority based threading model.

– Use of Server clusters/JVM clustering (e.g. terracotta).
– Use of distributed cache.
– Use of Big Data like MongDB and NoSQL where applicable.
– Use of SEDA based architecture.
– Use of Message Oriented Middle-ware for loosely coupled and asynchronous architecture.
– Use of CDN (Content Delivery Networks) for storing web resources.
– Use proven libraries like MINA, Cameron FIX, Javalution, Trove collection, etc.

Popular Posts

♦ 11 Spring boot interview questions & answers

828 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

768 views

18 Java scenarios based interview Questions and Answers

400 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

389 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

296 views

♦ 7 Java debugging interview questions & answers

293 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

286 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

279 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

240 views

001B: ♦ Java architecture & design concepts interview questions & answers

202 views

Bio | **Latest Posts**

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ More JSF interview Q&A

4 FAQ Performance tuning in Java interview Q&As ›

Posted in Java Key Area Essentials, member-paid, Scalability

Tags: Architect FAQs

Leave a Reply

Logged in as geethika. Log out?

Comment

Post Comment

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from "Blah blah" to "Wow"? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.