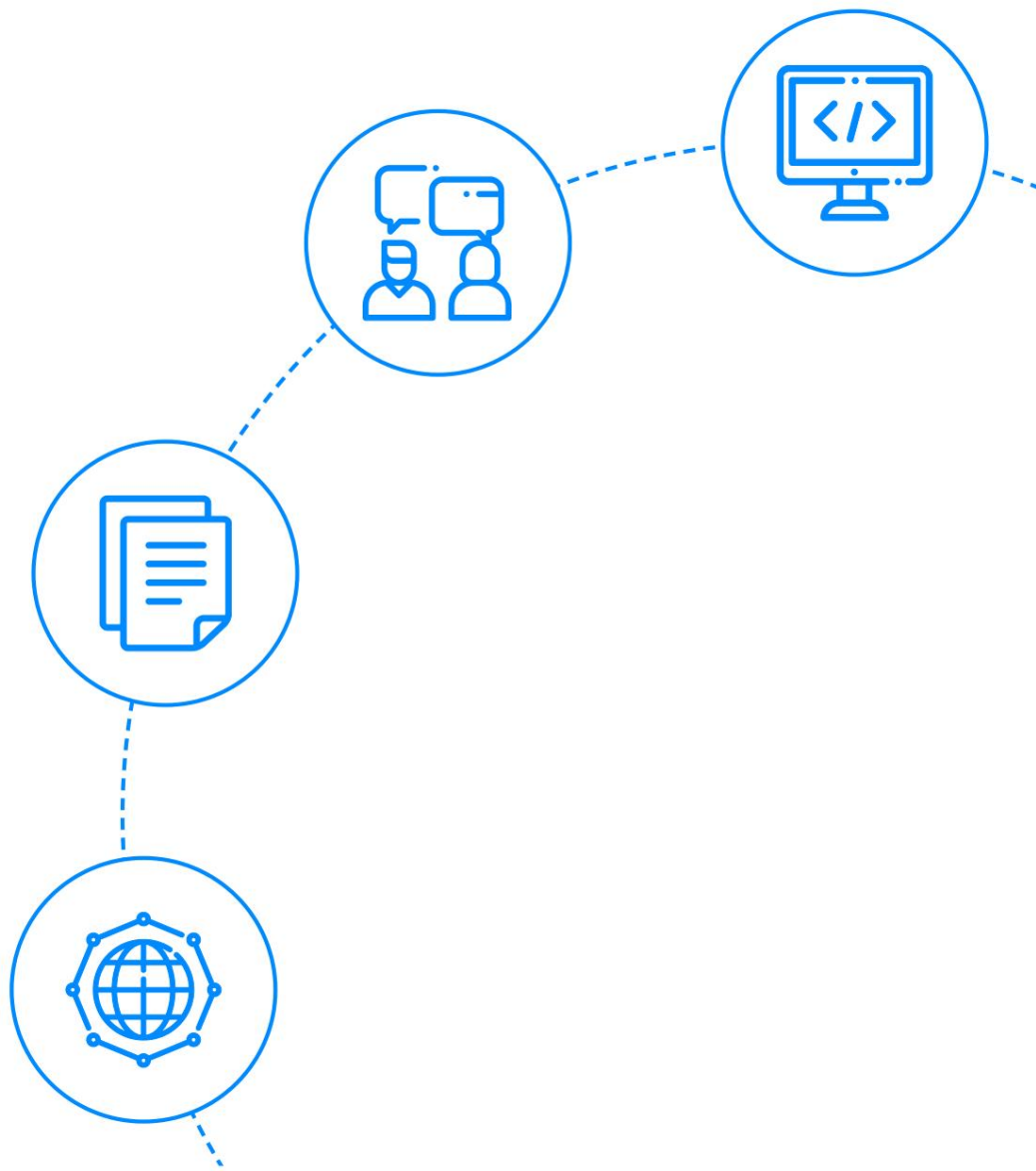




InterviewBit

सिस्टम डिजाइन साक्षात्कार प्रश्न



का लाइव संस्करण देखने के लिए
पेज, यहां [क्लिक करें](#)।

अंतर्वस्तु

फ्रेशर्स के लिए सिस्टम डिज़ाइन साक्षात्कार प्रश्न

1. सीएपी प्रमेय क्या है?
2. क्षेत्रीय स्केलिंग, लम्बवत स्केलिंग से किस प्रकार भिन्न है?
3. भार संतुलन से आप क्या समझते हैं? सिस्टम डिजाइन में यह महत्वपूर्ण क्यों है?
4. किसी सिस्टम की लेटेंसी, थ्रूपुट और उपलब्धता से आप क्या समझते हैं?
5. शेयरिंग क्या है?
6. NoSQL डेटाबेस SQL डेटाबेस से कैसे अलग है?
7. शार्डिंग विभाजन से किस प्रकार भिन्न है?
8. प्रदर्शन और मापनीयता एक दूसरे से कैसे संबंधित है?
9. कैशिंग क्या है? में उपलब्ध विभिन्न कैशे अद्यतन रणनीतियाँ क्या हैं कैशिंग?
10. सिस्टम डिजाइन में उपलब्ध विभिन्न संगति पैटर्न क्या हैं?
11. सामग्री वितरण नेटवर्क से आप क्या समझते हैं?
12. नेता चुनाव से आप क्या समझते हैं?
13. आप सिस्टम डिज़ाइन साक्षात्कार प्रश्नों का उत्तर कैसे देते हैं?
14. वितरित प्रणालियों में कुछ डिज़ाइन मुद्दे क्या हैं?

सिस्टम डिजाइन साक्षात्कार प्रश्न अनुभव

15. व्हाट्सएप या फेसबुक मैसेंजर जैसी वैश्विक चैट सेवा डिजाइन करें।
16. आप TinyURL या bit.ly जैसी URL छोटा करने वाली सेवा कैसे डिज़ाइन करते हैं?
17. Quora, Reddit या HackerNews जैसे फ़ोरम जैसे सिस्टम डिज़ाइन करें।
18. फेसबुक का न्यूजफीड सिस्टम डिजाइन करें।

सिस्टम डिजाइन साक्षात्कार प्रश्न अनुभव

(..... जारी)

19. पार्किंग सिस्टम डिजाइन करें?
20. आप एक सिफारिश प्रणाली कैसे डिजाइन करते हैं?
21. GitHub या Firebase साइटों के लिए API रेट लिमिटर सिस्टम डिज़ाइन करें।
22. आप वैश्विक फ़ाइल संग्रहण और Google डिस्क जैसी फ़ाइल साझाकरण सेवाओं को कैसे डिज़ाइन करते हैं, ड्रॉपबॉक्स आदि?
23. टाइप-फ़ॉरवर्ड सर्च इंजन सेवा डिज़ाइन करें।
24. नेटफ्लिक्स डिजाइन करें।
25. टिक-टैक-टो गेम डिजाइन करें।
26. एक यातायात नियंत्रण प्रणाली डिजाइन करें।
27. वेब क्रॉलर डिज़ाइन करें।
28. एटीएम प्रणाली डिजाइन करें।
29. Uber, Ola या Ly प्रकार के सिस्टम डिज़ाइन करें।

आएँ शुरू करें

सिस्टम डिज़ाइन क्या है?

सिस्टम पहलुओं जैसे मॉड्यूल, आर्किटेक्चर, घटकों और उनके इंटरफेस, और निर्दिष्ट आवश्यकताओं के आधार पर सिस्टम के लिए डेटा स्थापित करने की प्रक्रिया को सिस्टम डिज़ाइन के रूप में जाना जाता है। यह किसी कंपनी या संगठन के विशिष्ट उद्देश्यों और अपेक्षाओं को पूरा करने वाली प्रणालियों को पहचानने, बनाने और डिज़ाइन करने की प्रक्रिया है। सिस्टम डिज़ाइन सिस्टम के विश्लेषण, आर्किटेक्चरल पैटर्न, एपीआई, डिज़ाइन पैटर्न, के बारे में अधिक है। और इसे कोडिंग के बारे में सब कुछ एक साथ चिपकाना है।

क्योंकि आपका एप्लिकेशन आर्किटेक्चरल लोड को संभालने में सक्षम होगा, आपके सिस्टम को आपके एप्लिकेशन की आवश्यकताओं के लिए पर्याप्त रूप से डिज़ाइन करने से अनावश्यक लागत और रखरखाव के प्रयास समाप्त हो जाएंगे, साथ ही साथ आपके अंतिम-उपयोगकर्ताओं के लिए एक बेहतर अनुभव प्रदान होगा।

जब तकनीकी साक्षात्कार की बात आती है तो सिस्टम डिज़ाइन को नज़रअंदाज़ करना असंभव है! साक्षात्कार में, लगभग हर आईटी दिग्गज, चाहे वह फेसबुक, अमेज़ॉन, Google या कोई अन्य हो, सिस्टम डिज़ाइन अवधारणाओं पर आधारित प्रश्नों की एक श्रृंखला पूछता है। जैसे स्केलेबिलिटी, लोड बैलेंसिंग, कैशिंग, और इसी तरह। तो बिना किसी और विराम के, आइए हम सिस्टम डिज़ाइन पर अक्सर पूछे जाने वाले साक्षात्कार प्रश्नों को देखें।

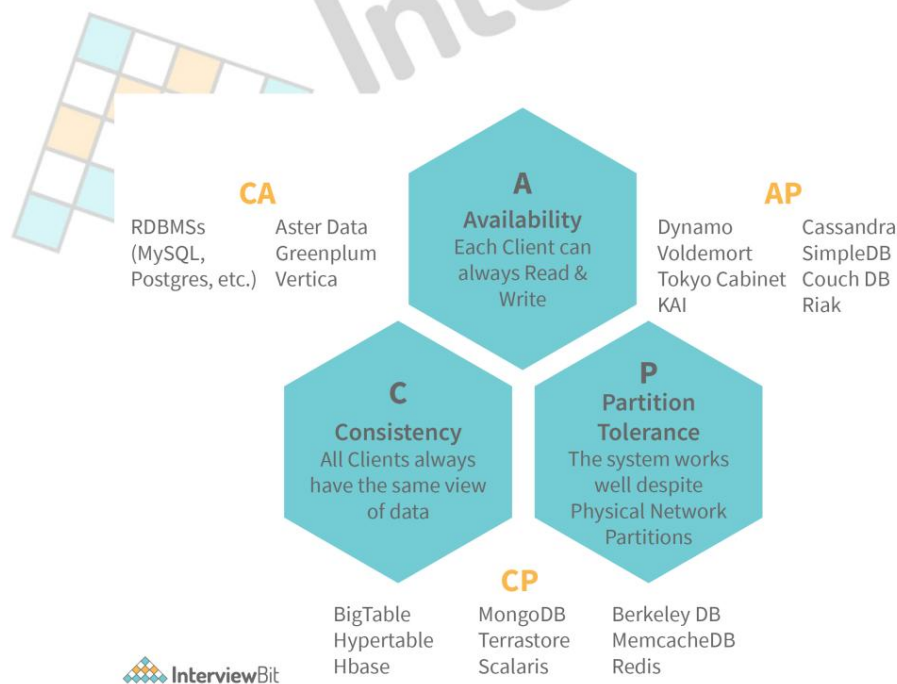
फ़ेशर्स के लिए सिस्टम डिज़ाइन साक्षात्कार प्रश्न

1. सीएपी प्रमेय क्या है?

सीएपी (संगति-उपलब्धता-विभाजन सहिष्णुता) प्रमेय कहता है कि एक वितरित प्रणाली सी, ए और पी को एक साथ गारंटी नहीं दे सकती है। यह अधिकतम 3 में से कोई 2 गारंटी प्रदान कर सकता है। आइए इसे एक डिस्ट्रिब्यूटेड डेटाबेस सिस्टम की मदद से समझते हैं।

- संगति: यह बताता है कि डेटा को डेटाबेस में एक ऑपरेशन के निष्पादन के अनुरूप रहना है। उदाहरण के लिए, डेटाबेस अद्यतन के बाद, सभी प्रश्नों को एक ही परिणाम प्राप्त करना चाहिए।
- उपलब्धता: डेटाबेस में डाउनटाइम नहीं हो सकता है और हमेशा उपलब्ध और उत्तरदायी होना चाहिए।
- विभाजन सहिष्णुता: संचार के अस्थिर होने के बावजूद डेटाबेस सिस्टम को कार्य करना चाहिए।

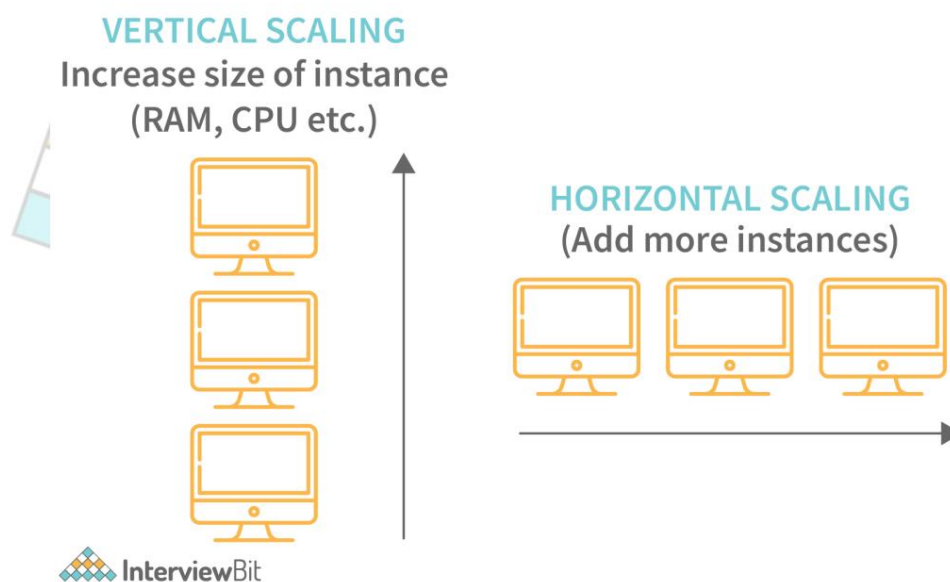
निम्नलिखित छवि दर्शाती है कि कौन से डेटाबेस सीएपी प्रमेय के किन पहलुओं की एक साथ गारंटी देते हैं। हम देखते हैं कि RDBMS डेटाबेस एक साथ निरंतरता और उपलब्धता की गारंटी देते हैं। Redis, MongoDB, Hbase डेटाबेस संगति और विभाजन सहिष्णुता की गारंटी देते हैं। Cassandra, CouchDB उपलब्धता और विभाजन सहनशीलता की गारंटी देता है।



2. क्षैतिज स्केलिंग, लम्बवत स्केलिंग से किस प्रकार भिन्न है?

- हॉरिजॉन्टल स्केलिंग से तात्पर्य उस नेटवर्क में अधिक कंप्यूटिंग मशीनों को जोड़ने से है जो उपकरणों के वितरित नेटवर्क में प्रोसेसिंग और मेमोरी वर्कलोड को साझा करता है। सरल शब्दों में, मौजूदा पूल में सर्वर के अधिक उदाहरण जोड़े जाते हैं और इन उपकरणों में कुशल तरीके से ट्रैफ़िक लोड वितरित किया जाता है।
- वर्टिकल स्केलिंग से तात्पर्य संसाधन क्षमता को उन्नत करने की अवधारणा से है जैसे कि रैम बढ़ाना, एकल मशीन के कुशल प्रोसेसर आदि को जोड़ना या अधिक क्षमता वाली नई मशीन पर स्विच करना। कोड हेरफेर की आवश्यकता के बिना सर्वर की क्षमता को बढ़ाया जा सकता है।

यह नीचे दी गई छवि में प्रदर्शित किया गया है:

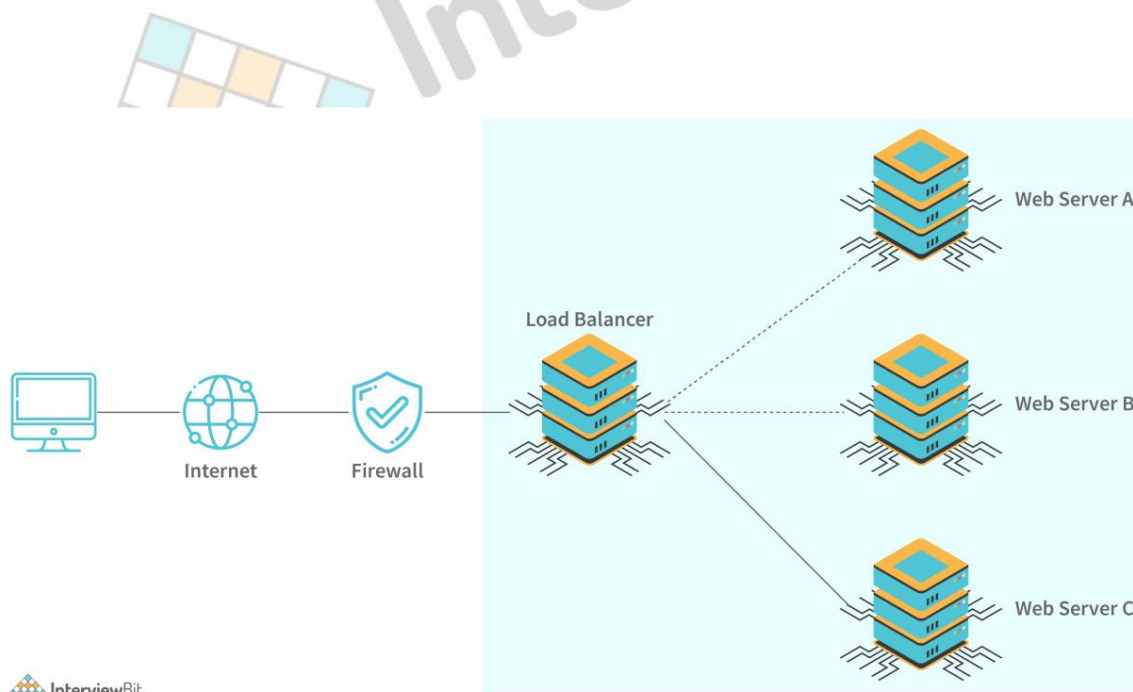


क्षैतिज स्केलिंग बनाम लंबवत स्केलिंग:

| श्रेणी | क्षैतिज स्केलिंग | लंबवत स्केलिंग |
|----------------|---|---|
| भार संतुलन | एकाधिक मशीनों में अनुरोध ट्रैफ़िक वितरित करने के लिए लोड संतुलन की आवश्यकता होती है। | चूंकि केवल एक ही मशीन है, इसलिए लोड बैलेंसर की आवश्यकता नहीं है। |
| असफलता लचीलापन | यह एप्लिकेशन विफलता के लिए अधिक प्रतिरोधी है क्योंकि यदि एक सर्वर विफल हो जाता है, तो ट्रैफ़िक दूसरे को भेज दिया जाता है सर्वर। | यह विफलता के लिए अधिक प्रवण है क्योंकि केवल एक मशीन है और इसके विफलता के परिणामस्वरूप विफलता होती है संपूर्ण आवेदन। |
| मशीन संचार | चूंकि इसमें कई मशीनें शामिल हैं, इसलिए नेटवर्क का होना बहुत जरूरी है संचार। | लंबवत स्केलिंग का उपयोग करता है मशीन के भीतर अंतर-प्रक्रिया संचार जो इसे काफी तेज बनाता है। |

3. भार संतुलन से आप क्या समझते हैं? क्यों यह है सिस्टम डिजाइन में महत्वपूर्ण है?

लोड बैलेंसिंग से तात्पर्य विभिन्न बैकएंड सर्वरों के समूह में आने वाले ट्रैफिक को कुशलतापूर्वक वितरित करने की अवधारणा से है। इन सर्वरों को सर्वर पूल कहा जाता है। आधुनिक समय की वेबसाइटों को ग्राहकों के लाखों अनुरोधों को पूरा करने और प्रतिक्रियाओं को तेज़ और विश्वसनीय तरीके से वापस करने के लिए डिज़ाइन किया गया है। इन अनुरोधों को पूरा करने के लिए, अधिक सर्वरों को जोड़ने की आवश्यकता है। ऐसे परिदृश्य में, अनुरोध ट्रैफिक को प्रत्येक सर्वर पर कुशलतापूर्वक वितरित करना आवश्यक है ताकि उन्हें अनुचित भार का सामना न करना पड़े। लोड बैलेंसर अनुरोधों का सामना करने वाले ट्रैफिक पुलिस सिपाही के रूप में कार्य करता है और उन्हें उपलब्ध सर्वरों पर इस तरह से रूट करता है कि एक भी सर्वर अभिभूत न हो जो संभवतः एप्लिकेशन के प्रदर्शन को खराब कर सकता है।



जब कोई सर्वर डाउन हो जाता है, तो लोड बैलेंसर ट्रैफिक को शेष उपलब्ध सर्वरों पर पुनर्निर्देशित करता है। जब कोई नया सर्वर कॉन्फिगरेशन में जुड़ जाता है, तो अनुरोध स्वचालित रूप से उस पर पुनर्निर्देशित हो जाते हैं। लोड बैलेंसर के लाभ निम्नलिखित हैं:

- वे अनुरोधों को अस्वस्थ या अनुपलब्ध सर्वर पर जाने से रोकने में मदद करते हैं।
- संसाधनों के ओवरलोडिंग को रोकने में मदद करता है।
- विफलता के एकल बिंदु को खत्म करने में मदद करता है क्योंकि जब भी कोई सर्वर डाउन होता है तो अनुरोध उपलब्ध सर्वरों पर भेज दिए जाते हैं।
- सर्वर को भेजे गए अनुरोधों को एन्क्रिप्ट किया जाता है और प्रतिक्रियाओं को डिक्రిप्ट किया जाता है। यह एसएसएल समाप्ति में सहायता करता है और X.509 प्रमाणपत्रों को स्थापित करने की आवश्यकता को दूर करता है हर सर्वर।
- लोड बैलेंसिंग सिस्टम की सुरक्षा को प्रभावित करता है और सिस्टम में बदलाव को समायोजित करने के लिए निरंतर सॉवर अपडेट की अनुमति देता है।

4. किसी सिस्टम की लेटेंसी, थ्रूपुट और उपलब्धता से आप क्या समझते हैं?

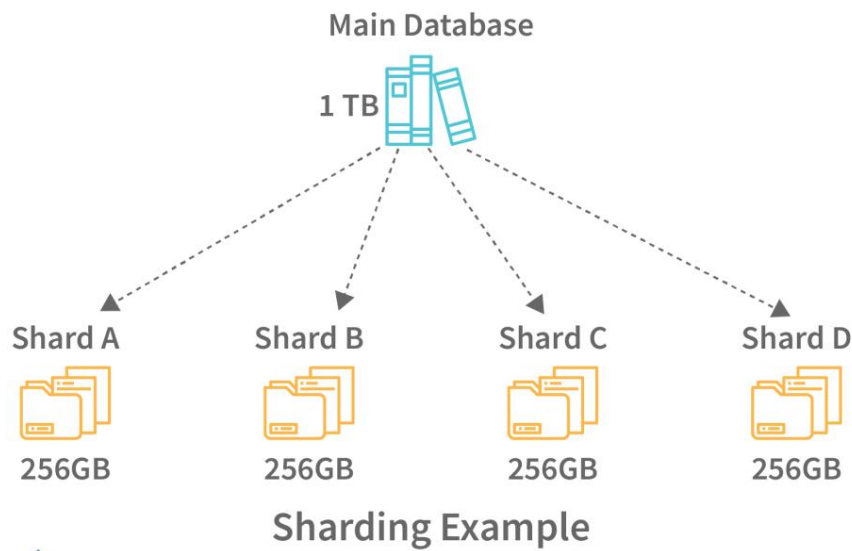
सिस्टम डिजाइन में प्रदर्शन एक महत्वपूर्ण कारक है क्योंकि यह हमारी सेवाओं को तेज़ और विश्वसनीय बनाने में मदद करता है। प्रदर्शन को मापने के लिए तीन प्रमुख मीट्रिक निम्नलिखित हैं:

- विलंबता: यह एक संदेश देने के लिए मिलीसेकंड में लिया गया समय है।
- थ्रूपुट: यह एक निश्चित समय में सिस्टम के माध्यम से सफलतापूर्वक प्रेषित डेटा की मात्रा है। इसे बिट्स प्रति सेकंड में मापा जाता है।
- उपलब्धता: यह निर्धारित करता है कि अनुरोधों का जवाब देने के लिए सिस्टम कितना समय उपलब्ध है। इसकी गणना की जाती है: $\text{सिस्टम अपटाइम} / (\text{सिस्टम अपटाइम} + \text{डाउनटाइम})$ ।

5. शेयरिंग क्या है?

शेयरिंग बड़े तार्किक डेटासेट को कई डेटाबेस में विभाजित करने की एक प्रक्रिया है। यह डेटा के क्षेत्रीय विभाजन को भी संदर्भित करता है क्योंकि इसे कई मशीनों पर संग्रहीत किया जाएगा।

ऐसा करने से, एक शार्ड डेटाबेस एक बड़ी मशीन की तुलना में अधिक अनुरोधों को संभालने में सक्षम हो जाता है। एक उदाहरण पर विचार करें - निम्न छवि में, मान लें कि हमारे पास डेटाबेस में लगभग 1TB डेटा मौजूद है, जब हम शार्डिंग करते हैं, तो हम बड़े 1TB डेटा को 256GB के छोटे टुकड़ों में विभाजित करते हैं, जिन्हें शार्प कहा जाता है।



शेयरिंग बड़े हुए थ्रूपुट, भंडारण क्षमता प्रदान करके और उच्च उपलब्धता सुनिश्चित करके बड़े हुए भार को संभालने में मदद करके डेटाबेस को स्केल करने में मदद करता है।

6. NoSQL डेटाबेस SQL डेटाबेस से कैसे अलग है?

| श्रेणी | एसक्यूएल | नोएसक्यूएल |
|-------------|---|---|
| नमूना | इस प्रकार है रिलेशनल नमूना। | गैर संबंधपरक मॉडल का अनुसरण करता है। |
| जानकारी | के साथ सौदें संरचित डेटा। | अर्ध संरचित डेटा के साथ डील करता है। |
| FLEXIBILITY | SQL इस प्रकार है a सख्त स्कीमा। | NoSQL से संबंधित है गतिशील स्कीमा और बहुत लचीला है। |
| लेनदेन | एसिड का पालन करता है (परमाणुता, संगतता, एकांत, स्थायित्व) गुण। | आधार का अनुसरण करता है (मूल उपलब्धता, सो-स्टेट, अंतिम संगतता) गुण। |

यहां अधिक अंतर देखें ।

7. शार्डिंग विभाजन से किस प्रकार भिन्न है?

- **Database Sharding** - Sharding is a technique for dividing a single dataset among many databases, allowing it to be stored across multiple workstations. Larger datasets can be divided into smaller parts and stored in numerous data nodes, boosting the system's total storage capacity. A sharded database, similarly, can accommodate more requests than a single system by dividing the data over numerous machines. Sharding, also known as horizontal scaling or scale-out, is a type of scaling in which more nodes are added to distribute the load. Horizontal scaling provides near-limitless scalability for handling large amounts of data and high-volume tasks.
- **Database Partitioning** - Partitioning is the process of separating stored database objects (tables, indexes, and views) into distinct portions. Large database items are partitioned to improve controllability, performance, and availability. Partitioning can enhance performance when accessing partitioned tables in specific instances. Partitioning can act as a leading column in indexes, reducing index size and increasing the likelihood of finding the most desired indexes in memory. When a large portion of one area is used in the resultset, scanning that region is much faster than accessing data scattered throughout the entire table by index. Adding and deleting sections allows for large-scale data uploading and deletion, which improves performance. Data that are rarely used can be uploaded to more affordable data storage devices.

The following table lists the differences between sharding and partitioning:

शेयरिंग

शेयरिंग एक प्रकार का विभाजन है और इसे क्षैतिज विभाजन भी कहा जाता है।

साझाकरण को स्कीमा की प्रतिकृति के रूप में भी परिभाषित किया जा सकता है और फिर एक शार्प कुंजी के आधार पर डेटा को विभाजित किया जा सकता है।

विभाजन

एक विभाजन एक तार्किक डेटाबेस का अलग, स्वतंत्र भागों में विभाजित है। डेटाबेस विभाजन का उपयोग आमतौर पर लोड संतुलन, प्रबंधनीयता, प्रदर्शन और उपलब्धता के लिए किया जाता है।



8. प्रदर्शन और मापनीयता एक दूसरे से कैसे संबंधित है?

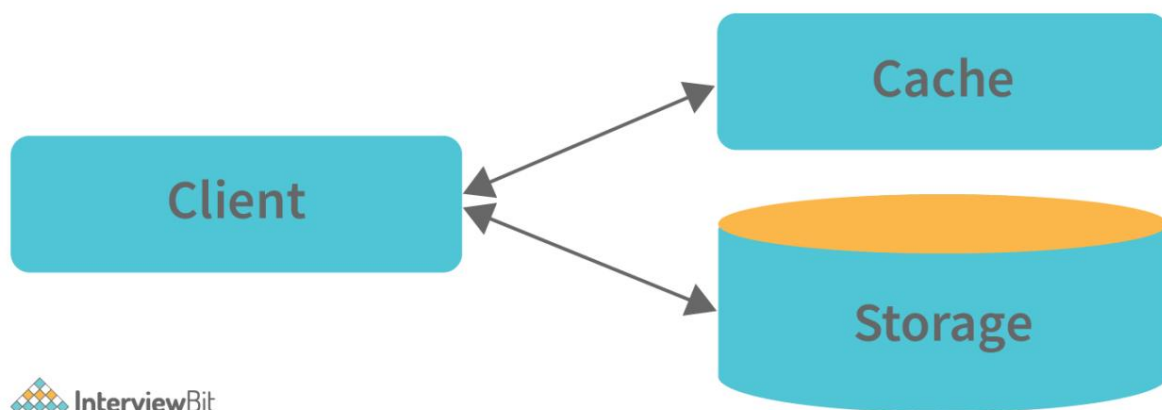
एक प्रणाली को स्केलेबल कहा जाता है यदि प्रदर्शन में वृद्धि हुई है जो संसाधनों के लिए आनुपातिक है। आम तौर पर, मापनीयता के संदर्भ में प्रदर्शन में वृद्धि का अर्थ अधिक कार्य इकाइयों की सेवा करना है। लेकिन इसका मतलब यह भी हो सकता है कि डेटासेट बढ़ने पर बड़ी कार्य इकाइयों को संभालने में सक्षम हो। यदि एप्लिकेशन में कोई प्रदर्शन समस्या है, तो सिस्टम केवल एक उपयोगकर्ता के लिए धीमा होगा। लेकिन अगर स्केलेबिलिटी की समस्या है, तो सिस्टम एकल उपयोगकर्ता के लिए तेज़ हो सकता है लेकिन एप्लिकेशन पर भारी उपयोगकर्ता लोड के तहत यह धीमा हो सकता है।

9. कैशिंग क्या है? कैशिंग में उपलब्ध विभिन्न कैशे अद्यतन रणनीतियाँ क्या हैं?

कैशिंग एक अस्थायी भंडारण स्थान में फ़ाइल प्रतियों को संग्रहीत करने की प्रक्रिया को संदर्भित करता है जिसे कैश कहा जाता है जो डेटा को अधिक तेज़ी से एक्सेस करने में मदद करता है जिससे साइट विलंबता कम हो जाती है। कैश केवल सीमित मात्रा में डेटा संग्रहीत कर सकता है। इसके कारण, कैशे अपडेट रणनीतियों को निर्धारित करना महत्वपूर्ण है जो व्यावसायिक आवश्यकताओं के लिए सबसे उपयुक्त हैं। उपलब्ध विभिन्न कैशिंग रणनीतियाँ निम्नलिखित हैं:

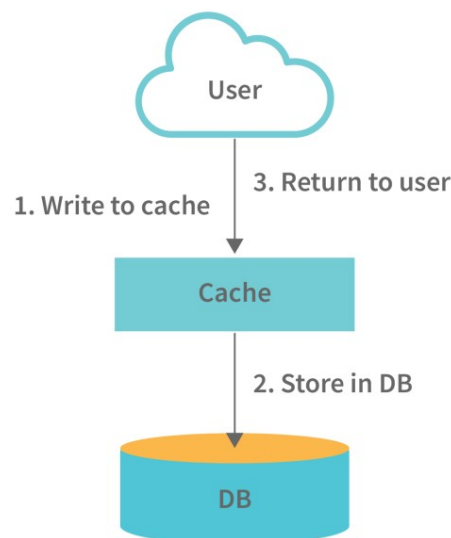
- कैश-असाइड: इस रणनीति में, हमारा एप्लिकेशन स्टोरेज से डेटा लिखने और पढ़ने के लिए जिम्मेदार है। भंडारण के साथ कैश इंटरैक्शन प्रत्यक्ष नहीं है। यहां, एप्लिकेशन कैश में एक प्रविष्टि की तलाश करता है, यदि परिणाम नहीं मिलता है, तो प्रविष्टि डेटाबेस से प्राप्त की जाती है और आगे के उपयोग के लिए कैश में जोड़ दी जाती है।

Memcached इस अद्यतन रणनीति का उपयोग करने का एक उदाहरण है।



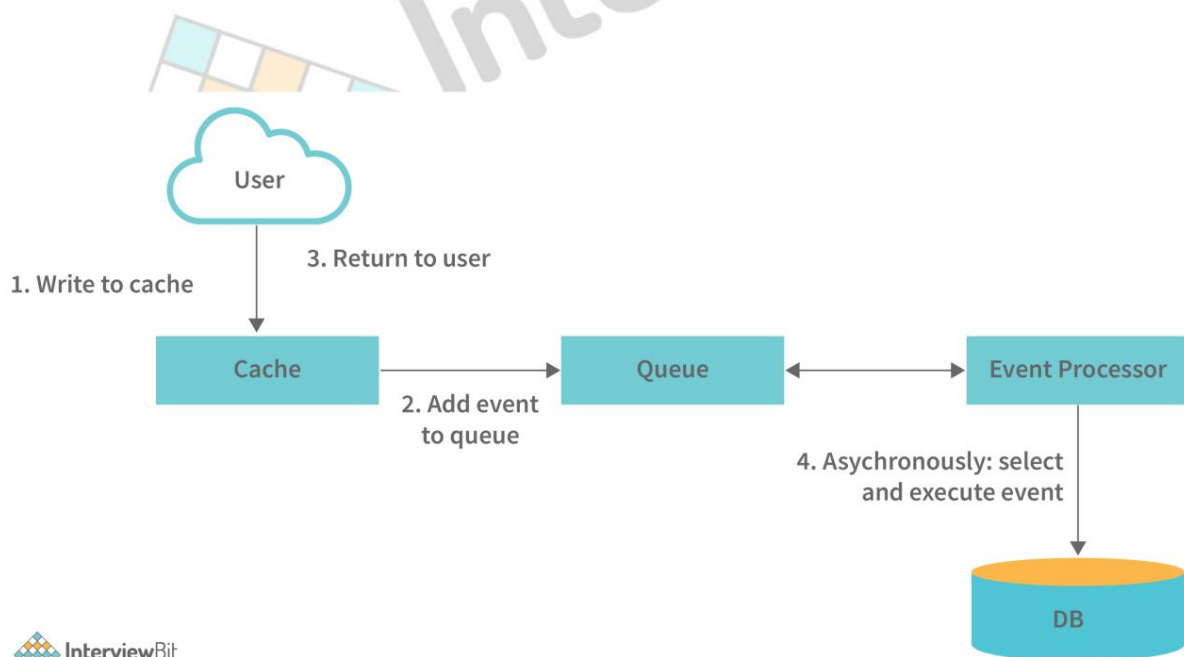
Cache-aside strategy is also known as lazy loading because only the requested entry will be cached thereby avoiding unnecessary caching of the data. Some of the disadvantages of this strategy are:

- In cases of a cache miss, there would be a noticeable delay as it results in fetching data from the database and then caching it.
- The chances of data being stale are more if it is updated in the database. This can be reduced by defining the time-to-live parameter which forces an update of the cache entry.
- When a cache node fails, it will be replaced by a new, empty node which results in increased latency.
- **Write-through:** In this strategy, the cache will be considered as the main data store by the system and the system reads and writes data into it. The cache then updates the database accordingly as shown in the database.



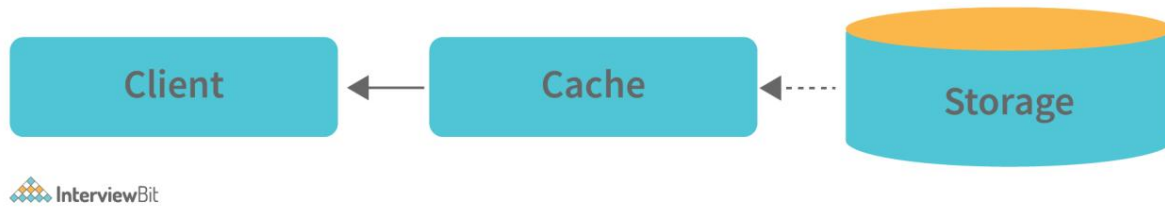
- सिस्टम कैश में प्रविष्टि जोड़ता या अद्यतन करता है।
- कैश समकालिक रूप से डेटाबेस में प्रविष्टियाँ लिखता है। सिंक्रोनस राइट ऑपरेशन के कारण यह रणनीति कुल मिलाकर एक धीमी गति से चलने वाला ऑपरेशन है। हालाँकि, हाल ही में लिखे गए डेटा के बाद के पठन बहुत तेज़ होंगे। यह रणनीति यह भी सुनिश्चित करती है कि कैश पुराना नहीं है। लेकिन, ऐसी संभावना है कि कैश में लिखा गया डेटा कभी पढ़ा न जाए। उपयुक्त टीटीएल प्रदान करके इस मुद्दे को कम किया जा सकता है।

- राइट-बैक (राइट-बैक): इस रणनीति में, एप्लिकेशन निम्न चरणों का पालन करता है: कैश में एक प्रविष्टि जोड़ें या अपडेट करें लेखन प्रदर्शन में सुधार के लिए डेटा स्टोर में प्रविष्टि को अतुल्यकालिक रूप से लिखें। यह नीचे दी गई
 - छवि में प्रदर्शित किया गया है:
 -



इस पद्धति का मुख्य नुकसान यह है कि यदि कैश की सामग्री को डेटाबेस में लिखे जाने से पहले कैश नीचे चला जाता है तो डेटा हानि की संभावना होती है।

- रीफ्रेश-फ़ॉरवर्ड: इस रणनीति का उपयोग करके, हम कैश की समाप्ति से पहले कैश प्रविष्टि को स्वचालित रूप से ताज़ा करने के लिए कैश को कॉन्फ़िगर कर सकते हैं।



इस कैश रणनीति के परिणामस्वरूप विलंबता कम हो जाती है यदि यह सटीक रूप से भविष्यवाणी कर सकती है कि भविष्य में किन वस्तुओं की आवश्यकता है।

10. सिस्टम डिजाइन में उपलब्ध विभिन्न संगति पैटर्न क्या हैं?

सीएपी प्रमेय से संगति बताती है कि प्रत्येक पढ़ने के अनुरोध को सबसे हाल ही में लिखित डेटा मिलना चाहिए। जब कई डेटा प्रतियां उपलब्ध होती हैं, तो उन्हें सिंक्रनाइज़ करने की समस्या उत्पन्न होती है ताकि ग्राहकों को लगातार ताजा डेटा मिल सके।

निम्नलिखित संगति पैटर्न उपलब्ध हैं:

- कमजोर संगति: यदि कोई डेटा लिखता है, तो रीड अनुरोध नया डेटा प्राप्त करने में सक्षम हो भी सकता है और नहीं भी। वीओआईपी, वीडियो चैट, रीयल-टाइम मल्टीप्लेयर गेम इत्यादि जैसे रीयल-टाइम उपयोग के मामलों में इस प्रकार की स्थिरता अच्छी तरह से काम करती है। उदाहरण के लिए, जब हम फोन कॉल पर होते हैं, अगर हम कुछ सेकंड के लिए नेटवर्क खो देते हैं, तो हम इसके बारे में जानकारी खो देते हैं उस दौरान क्या बोला गया था।
- अंतिम संगति: डेटा लिखने के बाद, रीड अंततः मिलीसेकंड के भीतर नवीनतम डेटा देखेंगे। यहां, डेटा को अतुल्यकालिक रूप से दोहराया जाता है। ये DNS और ईमेल सिस्टम में देखे जाते हैं। यह अत्यधिक उपलब्ध प्रणालियों में अच्छी तरह से काम करता है।
- मजबूत संगति: यदि कोई डेटा लिखता है, तो बाद के पढ़ने में नवीनतम डेटा दिखाई देगा। यहां, डेटा को समकालिक रूप से दोहराया जाता है। यह आरडीबीएमएस और फाइल सिस्टम में देखा जाता है और डेटा के लेनदेन की आवश्यकता वाले सिस्टम में उपयुक्त हैं।

11. सामग्री वितरण नेटवर्क से आप क्या समझते हैं?

सामग्री वितरण नेटवर्क या संक्षेप में सीडीएन एक विश्व स्तर पर वितरित प्रॉक्सी सर्वर नेटवर्क है जो अंतिम उपयोगकर्ताओं के लिए नजदीकी स्थानों से सामग्री परोसता है। आमतौर पर, वेबसाइटों में, सीडीएन से HTML, CSS, JS फाइलें, चित्र और वीडियो जैसी स्थिर फाइलें परोसी जाती हैं।

सामग्री वितरित करने में सीडीएन का उपयोग करने से प्रदर्शन में सुधार करने में मदद मिलती है:

- चूंकि उपयोगकर्ता अपने नजदीकी केंद्रों से डेटा प्राप्त करते हैं जैसा कि नीचे दी गई छवि में दिखाया गया है, उन्हें लंबे समय तक प्रतीक्षा करने की आवश्यकता नहीं है।



- सर्वर पर लोड काफी कम हो जाता है क्योंकि कुछ जिम्मेदारी सीडीएन द्वारा साझा की जाती है।

सीडीएन दो प्रकार के होते हैं, वे हैं:

- पुश सीडीएन: यहां, जब भी सर्वर में परिवर्तन होता है, तो सीडीएन द्वारा सामग्री प्राप्त की जाती है। सीडीएन पर सामग्री अपलोड करने की जिम्मेदारी हम पर है। सामग्री को सीडीएन में तभी अपडेट किया जाता है जब इसे संशोधित या जोड़ा जाता है जो बदले में यातायात को कम करके भंडारण को अधिकतम करता है। आम तौर पर, कम ट्रैफिक या सामग्री वाली साइटें पुश सीडीएन का उपयोग करके अच्छी तरह से काम करती हैं।
- पुल सीडीएन: जब पहला उपयोगकर्ता साइट से सामग्री का अनुरोध करता है तो यहां सर्वर से नई सामग्री ली जाती है। यह पहली बार धीमी गति से अनुरोध करता है जब तक कि सामग्री सीडीएन पर संग्रहीत/कैश नहीं हो जाती। ये सीडीएन सीडीएन पर उपयोग की जाने वाली जगह को कम करते हैं, लेकिन जब समाप्त हो चुकी फाइलों को बदलने से पहले खींच लिया जाता है तो वे अनावश्यक यातायात का कारण बन सकते हैं। पुल सीडीएन के साथ उपयोग किए जाने पर भारी ट्रैफिक वाली वेबसाइटें अच्छी तरह से काम करती हैं।

12. नेता चुनाव से आप क्या समझते हैं?

एक वितरित वातावरण में जहां कई सर्वर हैं जो एप्लिकेशन की उपलब्धता में योगदान दे रहे हैं, ऐसी स्थितियां हो सकती हैं जहां केवल एक सर्वर को तीसरे पक्ष के एपीआई को अपडेट करने के लिए नेतृत्व करना पड़ता है क्योंकि विभिन्न सर्वर तीसरे पक्ष के एपीआई का उपयोग करते समय समस्याएं पैदा कर सकते हैं। इस सर्वर को प्राथमिक सर्वर कहा जाता है और इस सर्वर को चुनने की प्रक्रिया को नेता चुनाव कहा जाता है। वितरित वातावरण में सर्वर को पता लगाना होता है कि लीडर सर्वर कब विफल हो गया है और लीडर बनने के लिए दूसरे को नियुक्त करें। सर्वसम्मति एल्गोरिथ्म का उपयोग करके उच्च उपलब्धता और मजबूत स्थिरता आधारित अनुप्रयोगों में यह प्रक्रिया सबसे उपयुक्त है।

13. आप सिस्टम डिज़ाइन साक्षात्कार प्रश्नों का उत्तर कैसे देते हैं?

- स्पष्टीकरण के लिए साक्षात्कारकर्ता से प्रश्न पूछें: चूंकि प्रश्न उद्देश्यपूर्ण रूप से अस्पष्ट हैं, यह सुनिश्चित करने के लिए साक्षात्कारकर्ता से प्रासंगिक प्रश्न पूछने की सलाह दी जाती है कि आप और साक्षात्कारकर्ता दोनों एक ही पृष्ठ पर हैं। प्रश्न पूछना यह भी दर्शाता है कि आप ग्राहकों की आवश्यकताओं की परवाह करते हैं।
- आवश्यकताओं को इकट्ठा करें: उन सभी सुविधाओं की सूची बनाएं जो आवश्यक हैं, सामान्य समस्याएं और सिस्टम प्रदर्शन पैरामीटर क्या हैं जिन्हें सिस्टम द्वारा संभालने की अपेक्षा की जाती है। यह कदम साक्षात्कारकर्ता को यह देखने में मदद करता है कि आप कितनी अच्छी योजना बनाते हैं, समस्याओं की अपेक्षा करते हैं और उनमें से प्रत्येक के समाधान के साथ आते हैं। सिस्टम डिजाइन करते समय हर पसंद मायने रखती है। प्रत्येक विकल्प के लिए, सिस्टम के कम से कम एक पक्ष और विपक्ष को सूचीबद्ध करने की आवश्यकता है।
- एक डिजाइन के साथ आओ: तय की गई प्रत्येक आवश्यकताओं के लिए एक उच्च-स्तरीय डिज़ाइन और निम्न-स्तरीय डिज़ाइन समाधान के साथ आओ। डिजाइन के पेशेवरों और विपक्षों पर चर्चा करें। साथ ही, चर्चा करें कि वे व्यवसाय के लिए कैसे फायदेमंद हैं।

सिस्टम डिज़ाइन साक्षात्कार का प्राथमिक उद्देश्य यह मूल्यांकन करना है कि एक डेवलपर किसी समस्या के लिए सर्वोत्तम संभव समाधान चुनने के लिए कितनी अच्छी तरह योजना बना सकता है, प्राथमिकता दे सकता है, विभिन्न विकल्पों का मूल्यांकन कर सकता है।

14. वितरित प्रणालियों में कुछ डिज़ाइन मुद्दे क्या हैं?

वितरित सिस्टम में पाए जाने वाले कुछ मुद्दे निम्नलिखित हैं:

- विषमता: इंटरनेट अनुप्रयोगों को कंप्यूटर और नेटवर्क के विषम संग्रह पर चलाने की अनुमति देता है। विभिन्न प्रकार के नेटवर्क होंगे और अंतर एक दूसरे के साथ संचार करने के लिए मानक इंटरनेट प्रोटोकॉल के उपयोग द्वारा छिपाए जाते हैं। वितरित अनुप्रयोगों को डिजाइन करते समय यह एक मुद्दा बन जाता है खुलापन: खुलापन उस माप का प्रतिनिधित्व करता है जिसके द्वारा एक प्रणाली को विभिन्न तरीकों से बढ़ाया और पुनः कार्यान्वित किया जा सकता है। वितरित सिस्टम में, यह उस डिग्री को निर्दिष्ट करता है जिसमें नई साझाकरण सेवाओं को जोड़ा जा सकता है और क्लाइंट उपयोग के लिए उपलब्ध कराया जा सकता है।
- सुरक्षा: वितरित सिस्टम में रखी गई जानकारी को सुरक्षित रखने की आवश्यकता है क्योंकि वे उपयोगकर्ताओं के लिए मूल्यवान हैं। वितरित प्रणालियों की गोपनीयता, उपलब्धता और अखंडता को बनाए रखना पड़ता है और यह कभी-कभी एक चुनौती बन जाती है।
- मापनीयता: अनुरोध यातायात और संसाधनों में उल्लेखनीय वृद्धि होने पर प्रणाली प्रभावी रहने पर मापनीय होती है। एक वितरित प्रणाली को डिजाइन करने में पहले से अच्छी तरह से योजना बनाना शामिल है कि अलग-अलग उपयोगकर्ता भार के तहत सिस्टम को कितनी अच्छी तरह स्केलेबल बनाया जा सकता है।
- विफलता से निपटने: एक वितरित वातावरण में, विफलताएं आंशिक होती हैं, जिसका अर्थ है कि यदि कुछ घटक विफल हो जाते हैं, तो अन्य अभी भी कार्य करेंगे। इन विफलताओं को संभालना चुनौतीपूर्ण हो जाता है क्योंकि इसमें सही घटकों की पहचान करना शामिल होता है जहां विफलताएं होती हैं।

सिस्टम डिजाइन साक्षात्कार प्रश्न

अनुभव

15. व्हाट्सएप या फेसबुक मैसेंजर जैसी वैश्विक चैट सेवा डिजाइन करें।

- कुछ आवश्यक विशेषताएं क्या हैं?
 - उपयोगकर्ताओं को इंटरनेट पर चैट करने की अनुमति दें।
 - आमने-सामने और समूह चैट के लिए सहायता प्रदान करें।
 - बेहतर देखने के लिए संदेशों को संग्रहीत करने की आवश्यकता है।
 - सुरक्षा उद्देश्यों के लिए संदेशों को एन्क्रिप्ट करने की आवश्यकता है।
- कुछ सामान्य समस्याएं क्या हैं जिनका सामना किया जा सकता है?
 - यदि किसी संदेश को बिना इंटरनेट कनेक्शन के भेजा जाता है तो उसका क्या होगा?
 - क्या एन्क्रिप्ट करने और डिक्रिप्ट करने से विलंबता बढ़ेगी?
 - डिवाइस पर संदेश कैसे भेजे और अधिसूचित किए जाते हैं?
- विचार के लिए संभावित सुझाव:
 - डेटाबेस स्कीमा को कई तालिकाओं में विभाजित करें जैसे कि उपयोगकर्ता तालिका, चैट तालिका, मालिश तालिका आदि।
 - डिवाइस और सर्वर के बीच द्वि-दिशात्मक संचार के लिए वेब सॉकेट का उपयोग करें।
 - सदस्यों को ऑनलाइन होने पर भी सूचित करने के लिए पुश नोटिफिकेशन का उपयोग करें।

16. आप TinyURL या . जैसी URL शॉर्टिंग सेवा कैसे डिज़ाइन करते हैं बिट.ली?

TinyURL या bit.ly एक लंबा URL लेता है और एक नया अद्वितीय लघु URL बनाता है। ये सिस्टम संक्षिप्त URL लेने और मूल पूर्ण URL वापस करने में भी सक्षम हैं।

- आवश्यक सुविधाओं में से कुछ क्या हैं?
 - एक छोटा URL बनाएं जिसकी लंबाई मूल URL से कम हो।
 - मूल URL को स्टोर करें और उसे संक्षिप्त में मैप करें।
 - संक्षिप्त URL में रीडायरेक्ट की अनुमति दें।
 - छोटे यूआरएल के लिए कस्टम नामों का समर्थन करें।
 - एक ही समय में कई अनुरोधों को संभालें।
- आम समस्याओं में से कुछ क्या हैं?
 - क्या होगा यदि दो उपयोगकर्ता एक ही कस्टम यूआरएल इनपुट करते हैं?
 - यदि अपेक्षा से अधिक उपयोगकर्ता लोड हैं तो क्या होगा?
 - आप डेटाबेस संग्रहण स्थान को कैसे नियंत्रित करते हैं?
- विचार के लिए संभावित सुझाव:
 - हैशिंग की अवधारणा का उपयोग मूल और नए URL को जोड़ने के लिए किया जा सकता है।
 - REST API का उपयोग उच्च ट्रैफ़िक को संतुलित करने और फ्रंट-एंड संचार को संभालने के लिए किया जा सकता है।
 - एक ही समय में कई अनुरोधों को संभालने के लिए मल्टीथ्रेडिंग अवधारणा।
 - मूल URL को संग्रहीत करने के लिए NoSQL डेटाबेस।

17. Quora, Reddit या HackerNews जैसे फ़ोरम जैसे सिस्टम डिज़ाइन करें।

ये साइटें प्रश्न पोस्ट करने और उनका उत्तर देने, टैग और संबंधित विषयों पर आधारित लोकप्रिय प्रश्नों को हाइलाइट करने वाली न्यूज़फ़ीड दिखाने के लिए हैं।

- **What are some of the Required Features?**

- Users should be able to create public posts and apply tags to them.
- Posts should be sortable based on tags.
- Post comments in real-time by users.
- Display posts on newsfeed based on followed tags.

- **What are some of the Common Problems encountered?**

- Should it be just a web application?
- Where to store the uploaded images and links?
- How can you determine the related tags?
- How can you distribute posts across a server network?

- **Possible tips for consideration:**

- Check on using SQL database for mapping relational data between users, posts, comments, likes, tags, posts etc.
- Incorporate multithreading and load balancer for supporting high traffic.
- Make use of sharding for distributing the data across different systems.
- Incorporate machine learning algorithms for finding correlations between the tags.

18. Design Facebook's newsfeed system.

Facebook's newsfeed allows users to see what is happening in their friend's circle, liked pages and groups followed.

- आवश्यक सुविधाओं में से कुछ क्या हैं?
 - उपयोगकर्ता द्वारा अनुसरण की जाने वाली अन्य सिस्टम संस्थाओं से पोस्ट का उपयोग करके न्यूज़फ़ीड उत्पन्न करें।
 - न्यूज़फ़ीड पोस्ट टेक्स्ट, छवि, ऑडियो या वीडियो प्रारूप के हो सकते हैं।
 - वास्तविक समय के करीब उपयोगकर्ता के न्यूज़फ़ीड में नई पोस्ट जोड़ें।
- आम समस्याओं में से कुछ क्या हैं?
 - क्या होगा यदि नई पोस्ट को समाचार फ़ीड में जोड़ने के लिए बहुत अधिक विलंबता दिखाई देती है?
 - क्या एल्गोरिथ्म अचानक उपयोगकर्ता लोड को संभाल सकता है?
 - समाचार फ़ीड में प्रदर्शित करने के लिए किन पदों को प्राथमिकता देनी चाहिए?
- विचार के लिए संभावित सुझाव:
 - अनुयायियों को पोस्ट प्रकाशित करने के लिए फैनआउट की प्रक्रिया का मूल्यांकन करें जांचें कि भारी
 - उपयोगकर्ता भार को संभालने के लिए शार्डिंग को कुशलतापूर्वक कैसे प्राप्त किया जा सकता है। किसी उपयोगकर्ता का फ़ीड डेटा एकाधिक सर्वरों में नहीं डाला जाना चाहिए। इसके बजाय, उपयोगकर्ता आईडी पर शार्डिंग की जा सकती है।

19. पार्किंग सिस्टम डिजाइन करें?

- आवश्यक सुविधाओं में से कुछ क्या हैं?
 - पार्किंग स्थल में कई स्तर हो सकते हैं जहां प्रत्येक स्तर पर पार्किंग स्थलों के लिए कई पंक्तियाँ होती हैं।
 - पार्किंग स्थल कारों, बसों, मोटरसाइकिलों के लिए पार्किंग का समर्थन कर सकता है इसलिए स्पॉट कई आकारों के हो सकते हैं।
 - सिस्टम डिजाइन करते समय पार्किंग स्थल की क्षमता पर विचार करें।
 - प्रत्येक पार्किंग स्थल के लिए उचित मूल्य निर्धारण डिजाइन करें।
- आम समस्याओं में से कुछ क्या हैं?
 - अगर हर जगह पर कब्जा है तो पार्किंग व्यवस्था का क्या होना चाहिए?
 - बड़े आकार के वाहनों को छोटे आकार के पार्किंग स्थल आवंटित करना।
- विचार के लिए संभावित सुझाव:
 - किसी वाहन के लिए उपयुक्त पार्किंग स्थल निर्दिष्ट करने के लिए एक एल्गोरिथम के बारे में सोचें।
 - सिस्टम को डिजाइन करने के लिए आवश्यक विभिन्न संस्थाओं के बारे में सोचें।

20. आप एक सिफारिश प्रणाली कैसे डिजाइन करते हैं?

अनुशंसा प्रणाली का उपयोग उपयोगकर्ताओं को उनके इतिहास या रुचियों के आधार पर विभिन्न विकल्पों और विकल्पों की पेशकश करके उनकी सहायता करके कुशलता से पहचानने में मदद करने के लिए किया जाता है कि वे क्या चाहते हैं।

- आवश्यक सुविधाओं में से कुछ क्या हैं?
 - चर्चा करें कि किस प्रकार की सिफारिश प्रणाली की आवश्यकता है - चाहे वह फिल्मों, ई-कॉमर्स वेबसाइटों, गीतों आदि के लिए हो।
- आम समस्याओं में से कुछ क्या हैं?
 - रीयल-टाइम में ताज़ा और प्रासंगिक सामग्री की अनुशंसा करने का तरीका जानें।
- विचार के लिए संभावित सुझाव: सिस्टम की
 - कार्यप्रणाली को समझने के लिए इवल घटक का उपयोग कैसे करें, इस पर चर्चा करें।
 - एक सहयोगी फ़िल्टरिंग दृष्टिकोण को प्रशिक्षित करने के तरीके पर चर्चा करें।

21. GitHub या Firebase के लिए API रेट लिमिटर सिस्टम डिज़ाइन करें साइटें

एपीआई दर सीमाएं अनुरोध अधिभार से बचने के लिए एक निश्चित समय अवधि में प्राप्त होने वाली एपीआई कॉल को सीमित करती हैं। यह प्रश्न एक मशीन पर वितरित नेटवर्क पर कोडिंग एल्गोरिथम से शुरू हो सकता है।

- आवश्यक सुविधाओं में से कुछ क्या हैं?
 - आवश्यक अनुरोध संख्या प्रति घंटा या सेकंड क्या है? आइए मान लें कि आवश्यकता प्रति सेकंड 10 अनुरोध हो सकती है।
 - अनुरोध अवरुद्ध होने पर क्या लिमिटर उपयोगकर्ता को सूचित करना चाहिए?
 - सीमक को पैमाने के अनुसार उपयुक्त यातायात को संभालना चाहिए।
- आम समस्याओं में से कुछ क्या हैं?
 - प्रति दिए गए समय के अनुरोधों को कैसे मापें?
 - स्थानीय सिस्टम की तुलना में वितरित सिस्टम के लिए रेट लिमिटर कैसे डिज़ाइन करें?
- विचार के लिए संभावित सुझाव:
 - घंटे के रीसेट से बचने के लिए स्लाइडिंग टाइम विंडो के उपयोग का मूल्यांकन करें।
 - स्थान बचाने के अनुरोध के बजाय काउंटर पूर्णांक का उपयोग करने का प्रयास करें।

22. आप ग्लोबल फाइल स्टोरेज और फाइल शेयरिंग सेवाओं जैसे गूगल ड्राइव, ड्रॉपबॉक्स आदि को कैसे डिजाइन करते हैं?

- आवश्यक सुविधाओं में से कुछ क्या हैं?
 - उपयोगकर्ता वेब पर फ़ाइलों को अपलोड करने, हटाने, साझा करने और डाउनलोड करने में सक्षम होना चाहिए।
 - फ़ाइल अपडेट को कई उपकरणों में समन्वयित किया जाना चाहिए।
- आम समस्याओं में से कुछ क्या हैं?
 - फाइलों को कहाँ स्टोर करें?
 - आप अपडेट कैसे संभाल सकते हैं? क्या फाइलों को फिर से अपलोड किया जाना चाहिए या सिर्फ संशोधित संस्करण को अपडेट करना है?
 - एक ही समय में दो दस्तावेज़ों के अपडेशन को कैसे हैंडल करें?
- विचार के लिए संभावित सुझाव:
 - संपूर्ण फ़ाइल के बजाय किसी विशेष अनुभाग के पुनः अपलोड का समर्थन करने के लिए फ़ाइलों को एकाधिक अनुभागों में विभाजित करने के लिए चंकिंग का उपयोग करने पर विचार करें।
 - फ़ाइलों को संग्रहीत करने के लिए क्लाउड स्टोरेज का उपयोग करें।

23. टाइप-फ़ॉरवर्ड सर्च इंजन सेवा डिज़ाइन करें।

यह सेवा उपयोगकर्ता द्वारा खोजी जाने वाली क्वेरी को पूरा करने के लिए सुझावों की संख्या प्रदर्शित करके आंशिक रूप से खोज क्वेरी को पूरा करती है।

- आवश्यक सुविधाओं में से कुछ क्या हैं?
 - सेवा को लोकप्रिय खोजी गई क्वेरी के साथ आंशिक क्वेरी का मिलान करना होगा।
 - सिस्टम को लिखित प्रश्न के आधार पर n सुझावों की संख्या (उदाहरण के लिए 5 कहेँ) प्रदर्शित करनी होती है।
 - सुझावों को क्वेरी अपडेशन के आधार पर अपडेट करना होगा।
- आम समस्याओं में से कुछ क्या हैं?
 - बिना अधिक विलंबता के सुझावों को कैसे अपडेट करें?
 - सबसे संभावित सुझाव कैसे निर्धारित करें?
 - क्या सुझाव उपयोगकर्ता के खोज परिणामों के अनुकूल हो रहे हैं?
 - सुझाव कब प्रकट होते हैं? क्या यह फ्लाय पर अपडेट होता है या एक बार उपयोगकर्ता लिखना बंद कर देता है?
- विचार करने के लिए संभावित सुझाव: अगले
 - वर्णों का अनुमान लगाने के लिए प्राकृतिक भाषा प्रसंस्करण के उपयोग का मूल्यांकन करें।
 - शीर्ष प्रश्नों की संभावनाओं की रैंकिंग के लिए मार्कोव श्रृंखला नियम।

24. नेटफ्लिक्स डिजाइन करें।

नेटफ्लिक्स एक वीडियो स्ट्रीमिंग सर्विस है।

- **What are some of the Required Features?**

- Uninterrupted video streaming to be made available for the users.
- Likes and reviews of videos.
- Recommend new videos.
- Support high traffic of users.

- **What are some of the common problems encountered?**

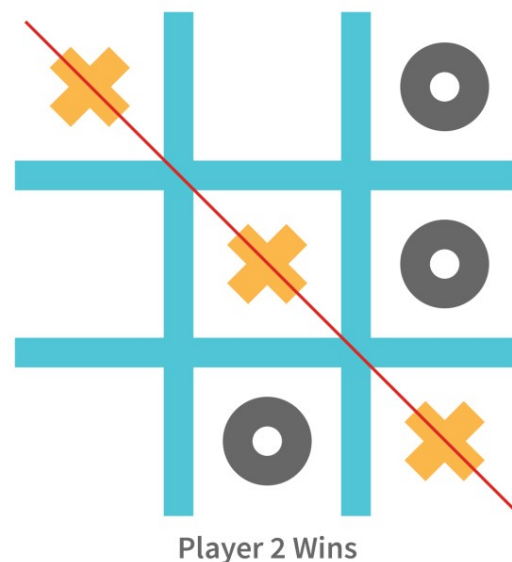
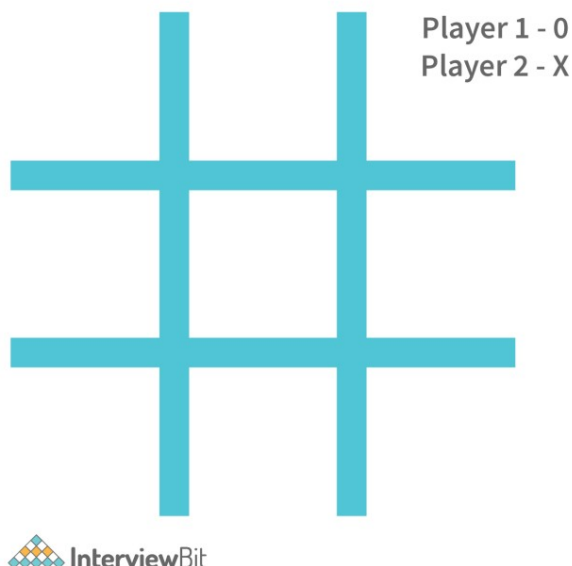
- Is it acceptable to have lags while uploading videos?
- What happens if many users are accessing the same video concurrently?

- **Possible tips for consideration:**

- Make use of cloud technology to store and transmit video data
- There are three components of Netflix: OC (Content Delivery Network), Backend database, Client device for accessing the application.

25. Design Tic-Tac-Toe game.

Tic-tac-toe game involves two players where one player chooses 0 and the other player chooses X for marking the cells. The player who fills a row/column/diagonal with their selected character wins.



- आवश्यक सुविधाओं में से कुछ क्या हैं?
 - 2 प्लेयर गेम का समर्थन करें जहां एक खिलाड़ी कंप्यूटर हो सकता है।
 - जीत और हार के परिणामों की गणना करने के लिए डिजाइन एल्गोरिदम।
- आम समस्याओं में से कुछ क्या हैं?
 - यदि दोनों खिलाड़ी बेहतर तरीके से खेलते हैं तो क्या होगा?
 - जीतने की रणनीति कैसे तय करें?
- विचार के लिए संभावित सुझाव:
 - यदि एक खिलाड़ी कंप्यूटर है, तो यह सुनिश्चित करने के लिए रैंड () विधि का उपयोग करें कि चाल पूरी तरह से यादृच्छिक है।

26. एक यातायात नियंत्रण प्रणाली डिजाइन करें।

आम तौर पर, यातायात नियंत्रण प्रणाली में, हम देखते हैं कि रोशनी लाल से हरे, हरे से नारंगी और फिर लाल से परिवर्तित होती है।

- आवश्यक सुविधाओं में से कुछ क्या हैं?
 - सम्मेलनों के आधार पर ट्रांजिशन ट्रैफिक लाइट।
- आम समस्याओं में से कुछ क्या हैं?
 - उस समय अंतराल का निर्धारण करें जिसके लिए ट्रैफिक लाइट की स्थिति बदलनी है।
 - सबसे खराब स्थिति में क्या होता है जहां राज्य को गलत तरीके से दिखाया जाता है?
- विचार के लिए संभावित सुझाव:
 - एक रंग से दूसरे रंग में राज्य के संक्रमण के लिए राज्य डिजाइन पैटर्न और शेड्यूलिंग एल्गोरिदम का उपयोग करें।

27. वेब क्रॉलर डिज़ाइन करें।

वेब क्रॉलर Google, DuckDuckGo जैसी एक खोज इंजन से संबंधित सेवा है और इसका उपयोग इंटरनेट पर वेबसाइट सामग्री को हर परिणाम के लिए उपलब्ध कराने के लिए अनुक्रमित करने के लिए किया जाता है।

- आवश्यक सुविधाओं में से कुछ क्या हैं?
 - संपूर्ण वेब से जानकारी एकत्र करने और लाखों वेब दस्तावेज़ प्राप्त करने के लिए एक स्केलेबल सेवा डिज़ाइन और विकसित करें।
 - प्रत्येक खोज क्वेरी के लिए ताजा डेटा लाया जाना है।
- आम समस्याओं में से कुछ क्या हैं?
 - जब उपयोगकर्ता बहुत तेजी से टाइप कर रहे हों तो अपडेट कैसे संभालें?
 - गतिशील रूप से बदलते वेब पेजों को प्राथमिकता कैसे दें?
- विचार के लिए संभावित सुझाव:
 - इस प्रणाली को लागू करने के लिए URL फ्रंटियर आर्किटेक्चर देखें।
 - जानिए कैसे रेंगना स्क्रीपिंग से अलग है।

28. एटीएम प्रणाली डिजाइन करें।

एटीएम का उपयोग ग्राहकों से पैसे जमा करने और निकालने के लिए किया जाता है। यह खाते की शेष राशि की जांच के लिए भी उपयोगी है।

- कुछ आवश्यक विशेषताएं क्या हैं?
 - प्रत्येक उपयोगकर्ता के पास कम से कम एक बैंक खाता होना चाहिए जो लेनदेन करने के लिए कार्ड से जुड़ा हो।
 - कार्ड से जुड़े 4 अंकों के पिन के आधार पर उपयोगकर्ता को प्रमाणित करने के लिए एटीएम।
 - उपयोगकर्ता एक निश्चित समय में केवल एक लेनदेन करने के लिए।
- आम समस्याओं में से कुछ क्या हैं?
 - लेन-देन समयबाह्य के दौरान क्या होता है?
 - क्या होता है यदि पैसा बैंक खाते से काट लिया जाता है लेकिन उपयोगकर्ता को मशीन से प्राप्त नहीं होता है?
- विचार के लिए संभावित सुझाव:
 - समस्या को विभिन्न संस्थाओं जैसे कार्ड, कार्ड रीडर आदि में विभाजित करें और प्रत्येक संस्था के बीच संबंध स्थापित करें।

29. Uber, Ola या Ly प्रकार के सिस्टम डिज़ाइन करें।

ये प्लेटफॉर्म यूजर रिक्वेस्ट राइड में मदद करते हैं और ड्राइवर उन्हें लोकेशन से उठाकर यूजर द्वारा चुने गए डेस्टिनेशन पर छोड़ देता है।

- कुछ आवश्यक विशेषताएं क्या हैं?
 - सवारी की बुकिंग के लिए रीयल-टाइम सेवा में सवारी
 - निर्दिष्ट करने की क्षमता होनी चाहिए जिससे उपयोगकर्ता गंतव्य तक तेजी से पहुंच सके।
 - सवारी की बुकिंग करने वाले चालक का ईटीए (आगमन का अनुमानित समय) दिखाएं और सवारी शुरू होने के बाद, गंतव्य पर पहुंचने वाले वाहन का ईटीए दिखाएं।
- आम समस्याओं में से कुछ क्या हैं?
 - हमेशा चलने वाले ड्राइवरों के लिए भौगोलिक स्थानों को कैसे स्टोर करें?
 - ग्राहकों को कुशलता से ड्राइवर कैसे असाइन करें?
 - आप ड्राइवर के आगमन या गंतव्य आगमन के ईटीए की गणना कैसे करते हैं?
- विचार के लिए संभावित सुझाव:
 - तेजी से बुकिंग के लिए तेजी से डेटाबेस के साथ माइक्रोसर्विसेज अवधारणा का उपयोग करें।
 - उपयोगकर्ताओं को ड्राइवर असाइन करने के लिए डिस्पैच सिस्टम का मूल्यांकन करें।

निष्कर्ष:

इस लेख में, हमने सिस्टम डिजाइन पर अक्सर पूछे जाने वाले साक्षात्कार प्रश्नों को शामिल किया है। सिस्टम डिजाइन इंटरव्यू को क्लियर करने का मुख्य तत्व यह है कि आपको उस दृष्टिकोण की स्पष्ट समझ होनी चाहिए जो आप किसी विशेष सिस्टम को डिजाइन करते समय अपना रहे हैं। उदाहरण के लिए, किसी सिस्टम में, यदि आप किसी SQL डेटाबेस में डेटा संग्रहीत करना चुनते हैं, तो आपको उस कारण के बारे में स्पष्ट होना चाहिए जिसके कारण आपने SQL डेटाबेस पर कोई SQL डेटाबेस नहीं चुना। आपको SQL और No SQL डेटाबेस के बीच अंतर के बारे में स्पष्ट होना चाहिए। दूसरे शब्दों में, आपका प्रत्येक प्रस्ताव किसी न किसी तार्किक तर्क द्वारा समर्थित होना चाहिए। इससे आपको इंटरव्यू में बढ़त मिलेगी।

उपयोगी संसाधन:

- [डेटा मॉडलिंग](#)
- [सिस्टम डिजाइन पुस्तकें](#)
- [सॉफ्टवेयर इंजीनियरिंग एमसीक्यू](#)

अधिक साक्षात्कार के लिए लिंक प्रश्न

ग साक्षात्कार सवाल

पी एच पी सम्बंधित इन्टरव्यू के सवाल

सी तीव्र साक्षात्कार प्रश्न

वेब एपीआई साक्षात्कार
प्रश्न

हाइबरनेट साक्षात्कार
प्रश्न

नोड जेएस साक्षात्कार प्रश्न

सीपीपी साक्षात्कार प्रश्न

उफ़ साक्षात्कार प्रश्न

देवोप्स साक्षात्कार प्रश्न

मशीन लर्निंग इंटरव्यू
प्रश्न

डॉकर साक्षात्कार प्रश्न मैसकल साक्षात्कार प्रश्न

सीएसएस साक्षात्कार प्रश्न

लारवेल साक्षात्कार प्रश्न एसपी नेट साक्षात्कार प्रश्न

Django साक्षात्कार प्रश्न डॉट नेट साक्षात्कार प्रश्न कुबेर्नेट्स साक्षात्कार

प्रश्न

ऑपरेटिंग सिस्टम साक्षात्कार
प्रश्न

प्रतिक्रिया मूल निवासी साक्षात्कार
प्रश्न

एडब्ल्यूएस साक्षात्कार प्रश्न

गिट साक्षात्कार प्रश्न

जावा 8 साक्षात्कार प्रश्न मोंगोडब साक्षात्कार
प्रश्न

डीबीएमएस साक्षात्कार सवाल

स्प्रिंग बूट साक्षात्कार
प्रश्न

पावर बीआई साक्षात्कार प्रश्न

PI Sql साक्षात्कार प्रश्न

झांकी साक्षात्कार
प्रश्न

लिनक्स साक्षात्कार प्रश्न

उत्तरदायी साक्षात्कार प्रश्न जावा साक्षात्कार प्रश्न

जेनकींस साक्षात्कार प्रश्न