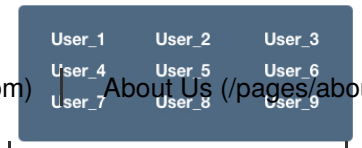
Sharding a Database

Bookmark

Let's design a shar ding scheme for key-value stor age.



Blog (https://blog.inter viewbit.com) | About Us (/pages/about_us/) | FAQ (/pages/faq/) |
Contact Us (/pages/contact_us/) | Ter ms (/pages/ter ms/) | Pr ivacy Policy (/pages/pr ivacy/)

System Design Inter view Questions (/cour ses/system-design/) |
Google Inter view Questions (/google-inter view-q uestions/) |
Facebook Inter view Questions (/facebook-inter view-q uestions/) |
Amazon Inter view Questions (/amazon-inter view-q uestions/) |
Micr osoft Inter view Questions (/micr osoft-inter view-q uestions/) | Puzzles Questions (/puzzles/)

Like Us (https://www.facebook.com/inter viewbit) | Follow Us (https://twitter .com/inter view_bit) | Email (mailto:hello@inter viewbit.com)

Features:

> *This is the first part of any system design interview, coming up with the features which the system should support. As an interviewee, you should try to list down all the features you can think of which our system should support. Try to spend around 2 minutes for this section in the interview. You can use the notes section alongside to remember what you wrote.*

Q: What is the amount of data that we need to stor e?

A: Let's assume a few 100 TB.

Q: Will the data keep gr owing over time? If yes, then at what r ate?
A: Yes. At the r ate of 1TB per day.

Q: Can we make assumptions about the stor age of machines available with me?
A: Let's assume that machines have a RAM of 72G and a har d disk capacity of 10TB.

Q: How many machines do I have to begin with?
A: Let's assume we have 20 machines to begin with. Mor e machines will be available on r equest if need be.

Q: Ar e all key value entr ies independent?
A: Yes. A typical quer y would ask for value cor r esponding to a key.

Estimation:

> *“ This is usually the second part of a design interview, coming up with the estimated numbers of how scalable our system should be. Important parameters to remember for this section is the number of queries per second and the data which the system will be required to handle.*
> *Try to spend around 5 minutes for this section in the interview. ”*

Total stor age size : 100 TB as estimated ear lier
Stor age with ever y machine : 10TB
Q: What is the minimum number of machines r equir ed to stor e the data?

A: Assuming a machine has 10TB of har d disk, we would need minimum of 100TB / 10 TB = 10 machines to stor e the said data. Do note that this is bar e minimum. The actual number might be higher.
In this case, we have 20 machines at our disposal.

()

Q: How fr equently would we need to add machines to our pool ?

A: The data gr ows at 1TB per day. That means that we gener ate data that would fill the stor age of 1 machine (10TB) in 10 days. Assuming, we want to keep a stor age utilization of less than 80%, we would need to add a new machine ever y 8 days.

()

Deep Dive:

> *Lets dig deeper into every component one by one. Discussion for this section will take majority of the interview time(20-30 minutes).*

> Note : In questions like these, the interviewer is looking at how you approach designing a solution. So, saying that I'll use a distributed file system like HDFS is not a valid response. It's okay to discuss the architecture of HDFS with details around how HDFS handles various scenarios internally.

Q: Can we have a fixed number of shar ds?

A: One qualification for a shar d is that the data within a shar d should fit on a single machine completely.
As in our case, the data is gr owing at a fast pace, if we have a fixed number of shar ds, data within a shar d will keep gr owing and exceed the 10TB mar k we have set per machine. Hence, we cannot have a fixed number of shar ds. The shar ds will have to incr ease with time.

Q: How many shar ds do we have and how do we distr ibute the data within the shar d?

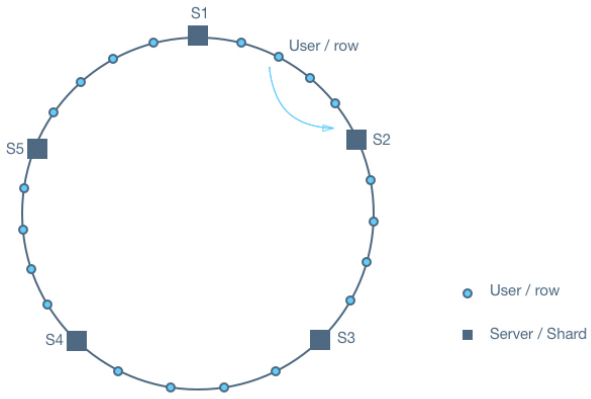
A: Lets say our number of shar ds is S. One way to shar d is that for ever y key, we calculate a numer ic hash H, and assign the key to the shar d cor r esponding to H % S.
Ther e is one pr oblem her e though. As we discussed ear lier , the number of shar ds will have to incr ease. And when it does, our new number of shar d becomes S+1.
As, such H%(S+1) changes for ever y single key causing us to r elocate each and ever y key in our data stor e. This is extr emely expensive and highly undesir able.

()

Q: Can we think of a better shar ding str ategy?

Hint: Consistent Hashing.

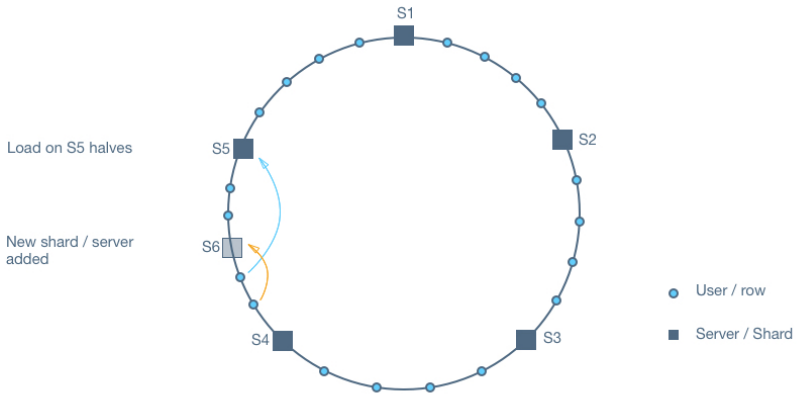
A: Consistent hashing is ideal for the situation descr ibed her e. Lets explor e consistent hashing her e.
Let's say we calculate a 64 bit integer hash for ever y key and map it to a r ing.
Lets say we star t with X shar ds. Each shar d is assigned a position on the r ing as well. Each key maps to the fir st shar d on the r ing in clockwise dir ection.



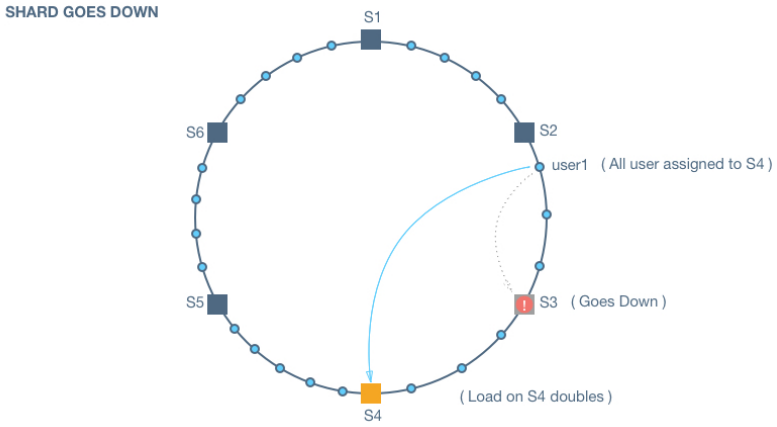
What happens if we need to add another shar d ? Or what if one of the shar d goes down and we need to r e-distr ibute the data among r emaining shar ds?

Got suggestions ? We would love to hear your feedback.

Loved InterviewBit ? Write us a testimonial. (http://www.quora.com/What-is-your-review-of-InterviewBit)

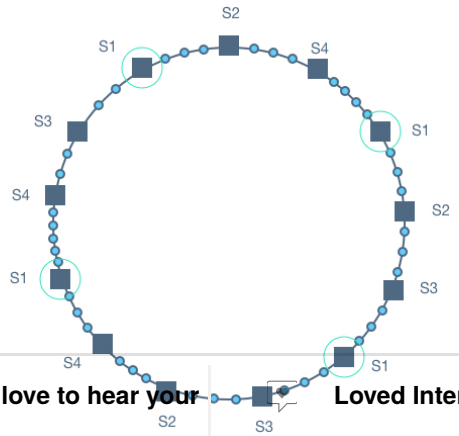


Similar ily, ther e is a pr oblem of cascading failur e when a shar d goes down.

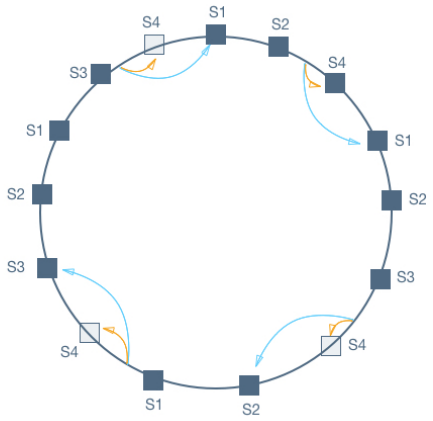


Modified consistent hashing

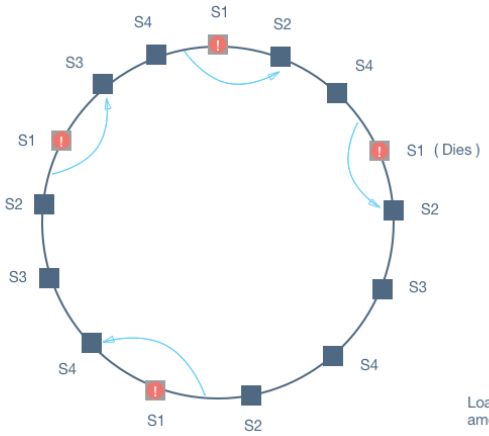
What if we slightly changed the r ing so that instead of one copy per shar d, now we have multiple copies of the same shar d spr ead over the r ing.



Case when new shar d is added :



Case when a shar d goes down : No cascading failur e. Yay!



You have now mastered this problem!

Got suggestions ? We would love to hear your feedback.

Loved InterviewBit ? Write us a testimonial.

(http://www.quora.com/What-is-your-review-of-InterviewBit)

Discussion