Hello everyone, our tonight’s class is on Zookeeper and Kafka. These two are some of the most important systems that you would see being used across systems that work at scale. We are going to dig deeper into these and understand their foundations. Make it a point to attend the class live. This is going to be thoroughly rewarding.

Also, we are going to start the class exactly on time. The content will start from 9:03pm. Join on time.

Batch Name: Academy Nov22 Intermediate 1

Topic to be covered: HLD: Zookeeper + Kafka.

Class Time: 09:00 pm (IST)

Link to join: Classroom link

**Following are the topics we shall be covering in the upcoming class.**

Topic 1: Problem Statement 1 (State tracking)

Topic 2: Naive approach to the state tracking problem

Topic 3: Zookeeper as a solution to state tracking problem

Topic 4: ZK Nodes

Topic 5: ZK Node for consistency Master Election

Topic 6: ZK Setting a watch

Topic 7: ZK Architecture

Topic 8: ZK Master dies

Topic 9: Problem Statement 2 (Async tasks)

Topic 10: Solution using Persistent queue

Topic 11: PubSub

Topic 12: Topics

Topic 13: Using kafka as a solution

Topic 14: Problem statements

Topic 15: Doubt Resolution

First half will talk about a system called zookeeper. 2nd Kafka

Zookeeper: a popular system in world of scalable system. Even I other system people use it. These are built system we just use. Will look at internal of them how do they do things internaly and they solve it. How can you understand the things the system can do for us. Optimal and non OPt things they do.

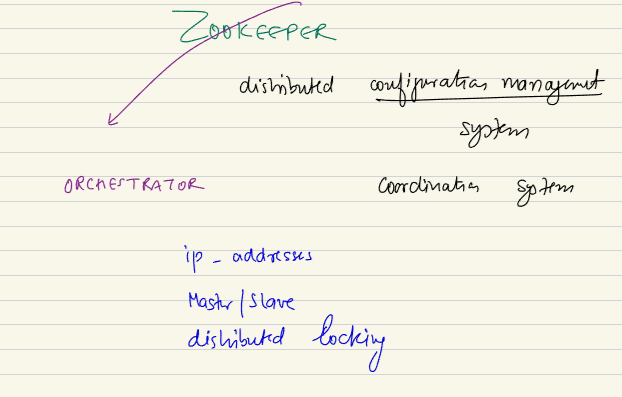
Zookeeper: a distributed configuration management system. Coordination system.

What does it mean? In high level scalable system, there are many component involves, DB, master and slave machine. Each machine will interact with bunch of other machine. How does a machine know who is master, who is salve, how do I know a master is aliev or dead. How do such a complicated system works as one. How does a machine know which IP I go to query/wwrite?

Master can go down and a slave go down. Point is each of the machine can go down and dynamic in nature. How does a system keep a track of all of such complicated system. This is where zookeeper comes in picture.

Knowing the IP of a shard, or a master or a slave IP, IP of these machines zookeeper is solution. Hwo is mater or who is slave, ZK is solution.

Distributed system me multiple people, machine. Thread try to access and get a lock over something. zK can help in districuted locking as well.



Orchestrator: who orchestrate the bunch of thing that play in front of us. In such system zookeeper plays role of orchtrator.. zookeeper is a product. Will talk on foundation how does a product like zookeeper is working.

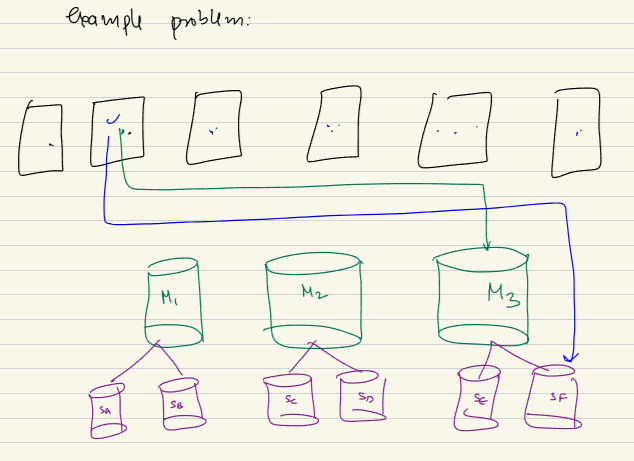
Zookeeper is a poster boy, it will have its own machines. It will have diff system itself which will help to

Lets look at a real life problem.. and see how ZK help to solve the problem..

We have a app server cluster, there might be few shards (entire data cant fit in one and I devide in 3 machine.. m1, m2, m3)… sharding can also work in replica. Each of shard will have replication. Master and 2 slaves.. Sa, Sb, Sc, Sd, Se, Sf.. every masdter has 2 slaves.

All of these app servers might have to interact with these DBs. To write will go tomaster. For read we can allow read from slaves. All these machine will need the knowledge of who is master and who is slave.. tomorrow if I wantto add a new app server machine that machine also need to know this. Same way if I add a enw shard and new slaves. Same way these app server needs to know about this entry. Also if Sd die and new machine take places. Then app also need to know about the new DB.

If a master dies and I need to write then unless master selection happens, once the change happens all the app server ip needs to know the new master.



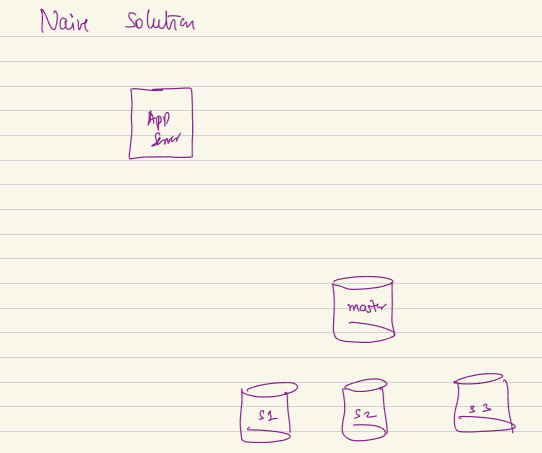
Service discovery also can be a use case:

A payment server need to intercat with order serveice. So it need to know ip and port to make HTTP call to… so not only depend on app-DB. It can also happen in one serveice to another service or oen app to another app server. Zookeeper can help me on these…

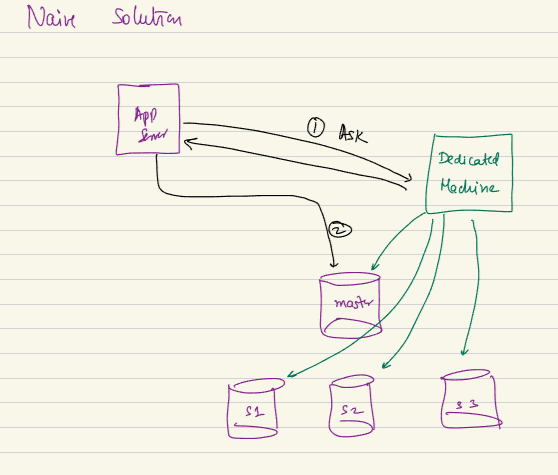
Multiple serveice and these need to know about existance about other services service to service talking in micro services.

Best way to learn is build it ourselves. Lest see how such system can be build..

App server need to know the ip of the aster and salve 1,2,3..



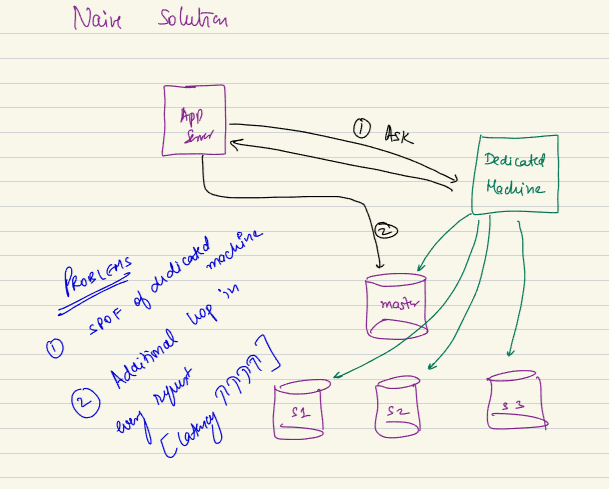
This will keep on changing. Master machine can die and new will come. Most naïve sol is lets have a dedicated machine whose IP everyone knows, every component of a system know s the ip of this machine. This machine will send IP heart beat of all system. DM keeps interacting with all machine. When app server needs to know. App s will go to dm and ask the ip of the master or this particular salve. DM will reply. App will go to particular machine.



If we have a dedicated machine. We can assume that Load balancer is doing this task. I can give this responsibility to load Balnor any machine.

Here we have 2 problems:

1. Single point of failure: if the dedicated machine dies every communication fails. Like DNS system. If DNS down every resource in internet not reachable.
2. If every color first need to dedicated amchine and ask then go to correct machine there is a additional HOP in every request. In other words.. Latency increases.



How can I solve the single point of failure in system design?

By introducing replicas. By havinga bucnch of machine that together keeps the state.

Additional hop. I have to go that machine and go to correact machine. I ahev to go on same machine again and again. To solve it I can do caching. Caching cankeep me out of sync. I want my machine to have correct master always. It should have write-through cache..

So we should think about all these edge cache..

**Lets look at system, that is zookeeper:**

We have naïve/ possible solution.. the nicely introcuced solution is zookeeper… lest understand idea system behind this product. ZK acts/ behaves like a file system.

Given I wantto keep a track of state, ip, port who is master hwo is salve, I would want myself to exists as a file systema and in lunux we have

Root /

Many next leverl folder etc,.

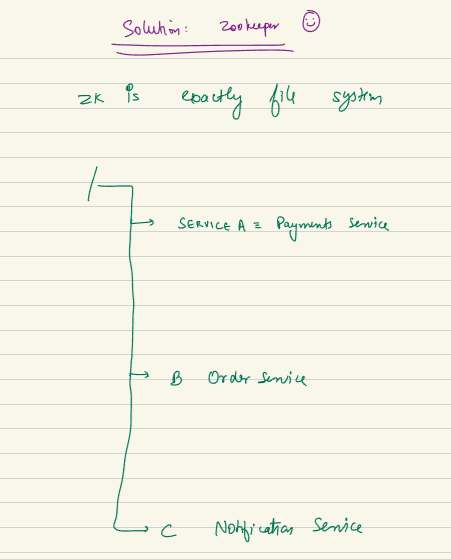
For every service or cluster I might want to have different folder. Global cahche, DB cluster. They are diff component.. there can be multiple other… we can thing of this arch as a combo of hirercical stu=ructure.

In service 1 I have sub folder like that.

Subfolder represents every service. Lest say I have a first level folder which has payment service.

Another B service has order service..

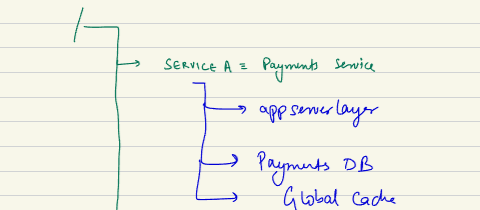
C can have notification service.



Folder str helps to go to correct place.

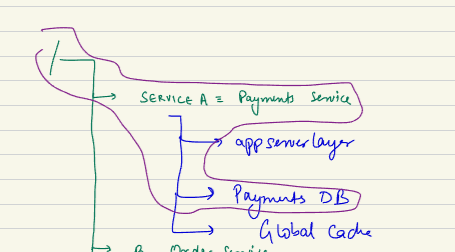
In service I can have app server layer and another I can have payment DB and global cache.. they are second level.. inside payment service

ZK acts as a file system. This folder structure will be maintained inside ZK.



Lest go inside one folder structure and zoom in..

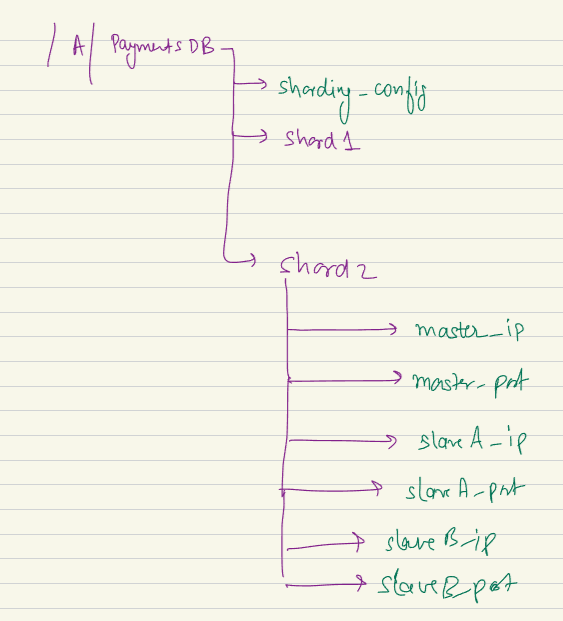
Root/service -A: payment service/ payment DB



Inside a DB of payment there will be multiple shards and I mght have a file which says sharding\_configuration. A file keeps the config info of how many shards are there.. parrelel to this.. folder inside this shards.

Inside shard-2 I will have master and slaves.. green color = file, purple = folder

In shard -2 I might have master ip, master port.. slave A ip, slave A port etc…



All these are inside service A..

Q: can we not create xml file structure instead of multiple files?

This is nothing buta keep a value in structured folder structure.. green colors are files.. purple is folders. Inside these files.. master-ip\_file.. content will be just ip address of the machine which is acting as master for second shard of payment DB.

slaveA-ip: ip of slave of second shard of payment DB.

Elaborate folder structure: I want to i=ensure that the independenct files which might have answer can to them. Lock can have on dile. They might have race condition. I want creal folder so lock can be taken and they are consistence.

Why folder? Individual so I can take lock. Why lock: a master and 4 slave, if master die one slave will be master. But multiple can be want to master. But I want one can be master. I want very consistence value. I want to have a lock ..

In xml I will have all entity written inside same file. All ip, port will eb part of same file. Difficult for me to .

When I create a flder structure, I am not taking anything extra…not difficult. So that independenct locks, or owner of these file can be assign. No issue of synchronization.

Having independenct entities to take lock.

Lets assume there si asingle ZK machine. Don’t worry about multiple copies… inside ZK has this folder structure..

Vishal [43: ]: More like mutex, I kep entity diff no sync issue. I kep separate because if master go down, one salve become new master. The slave ip will change, no longer a salve. I want independent entity.. can take lock and no need to clutter and sepearte

When I have payment DB, there might be a kep associated with DB, authentication systm.

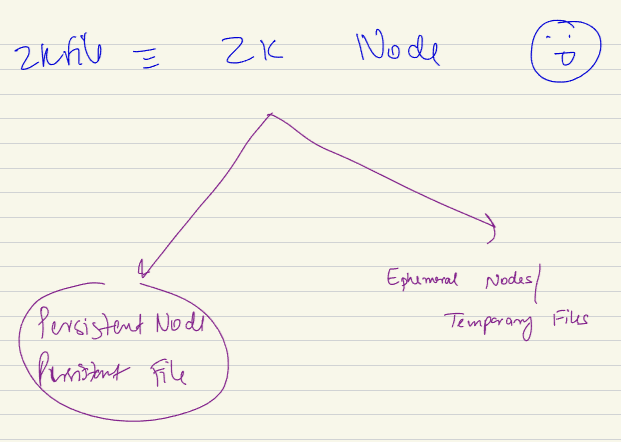
If DB based on AWS rds. As a system we might also want to know whats the aws key fo rthsi entire DB. There might a file on top of them a sinple aws key.. other machine can read after they authenticate themselves. Top level config that is true for all these below.

A common info will be on top. Aws key to acess the DB. A token I might wnatto store which allows to access certain things.

ZK is internally maintains a folder structure. They give funny name. zk calls each of these files nodes. Usually node means a server/ machine. In Zk world every file they call ZK node. ZK not means one ZK machne. Means a file inm ZK hierarchy.

2 kinds of ZK nodes… persistant node or file… Ephemeral node. (temporary files)

Things like sharding\_config, aws keep they store info like which is not going to change very often.



The master if, ip of a master machine, it it dies some other machine will be new master. Master ip will change. When that happens immediately master file neds to change..

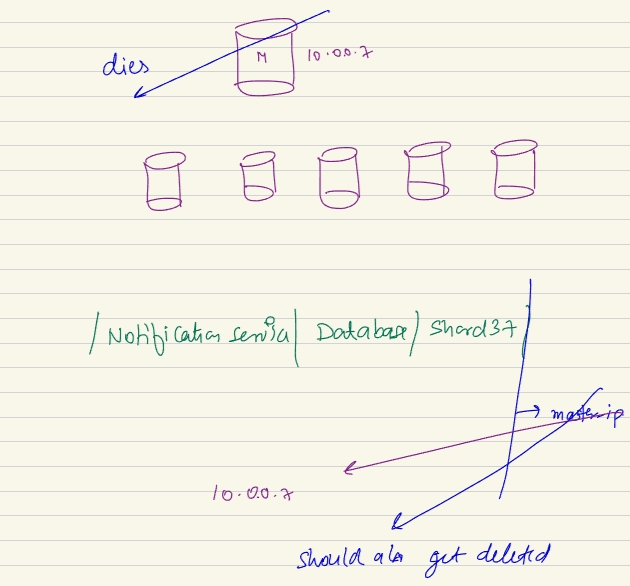
So 2 kind of file.. normal persisted file.

Another are **Ephemeral**: info about which are probability to immediately change.

Lets go background how thing work:

Imagine a master has few slaves and I have a structure.

/Notificationservice/ DB/ shard37/ in it bunch of files.. one of them is master IP file. This IP has an ip of 10.0.0.7.. this is ip of current master. But the moment the machine dies this master IP file also need to deleted as the info insdie is wrong now. As everybody other than this DB is going to use this info inside the file to know which is master ip of the shard37.



When this master dies my responsibility to delete the file. Hese kind of files store temporary info. Ndoes that are need to delete which lives as long as the node lives are knows as Ephemeral nodes.

Nodes which have to live only till their underlying owner machine is active.

On the other hands, nodes which has static info like aws token known as persistent nodes.

how were things handled before a zookeeper ? I mean can we have a distributed system without zookeeper

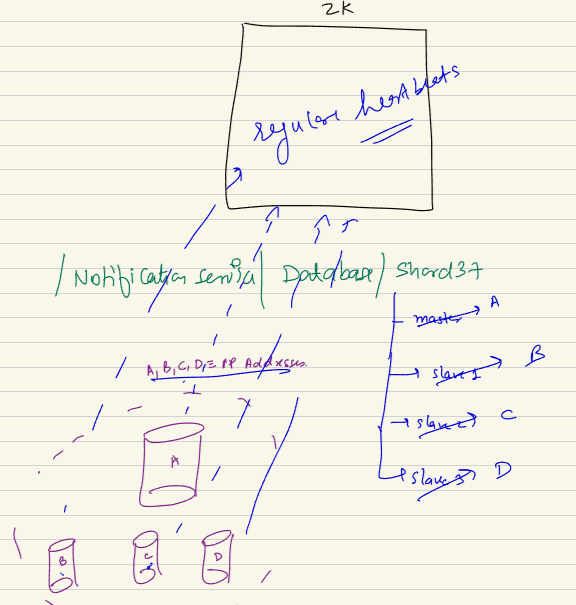
lest say I have a ZK system. I have a master and there are few slaves. At this moment I am worried about this folder structure. This is shard37.



A,B,C,D are my IP address..

At the beginning A is master A which has 3 slaves.. the value inside slaves are B,C,D

Each machine will send regular heartbeat to my ZK. Saying I am alive



I have these files, logical file inside ZK. All machine will send I am alive to ZK. The moment ZK know im ok. It knows. The master or lock is alive till the time owner is alive.

All machine are sending me heartbeat. I will keep the file intact and not allow nayone. These physical machine are owner.

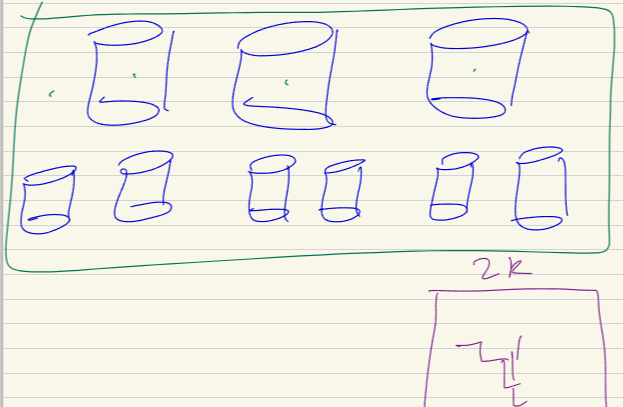
I have ohycial machine , master-slave amchine. I have a ZK. It is manager. Who own the machiens. Inside itself it has a regiuster, marks attendance, I have files for the master, slave 1,2,3. Now each of shard and machine tells I am alive, I will keep that file of that kmachine sacrosaned(not going to temper/ sacred). No let anyone update the file. If after sometime master stop sending me heartbeats. If I have 5 heartbest as a threshold. I can assume that master has died, I will looka t the file tear it and throw it awsy. I don’t have the file with me anymore. Only after few missed mesdsage I delete the file. It now measn every other component in system who were replying on zk. I can tell we do not have a master now…

Given I torn the file apart no master exists. Which measn that.. now I am manager I torn the file. Slaves will know master is dead as ZK just informed the master is dead. Immediately will try to be master. One person can take the lock. So onely one of them will be master and only one ..

ZK is mamanger who is keeping the files every entiry in real system.

There is a gateway.. inside it we have app servers. You have these shards, every shard have some slaves. If I am ZK machine acting as mamager/ orchestrator of the system. Inside my folder structure there will be a ephemeral file for each of these entities,a nd each of the file will conatin IP of the relevant amchine.

Inside storage machine I will have file which will have info about these machine…



Salve A file for master 1.

Each of these entities who is a part of orchestration. I am keeping file and keeping it sacrosand. Once I have these sacrosand file for the entity. If I stp getting update I will tear that file. If someone asks about [I ahev storage layer, what the master ip] the ip I wil tell sorry I don’t have it.

In steady state I read and tell this is ip.

If entity don’t send me heartbeat I will tear it up..

A file with me, for every entry, the moment entry dies I will tear it..

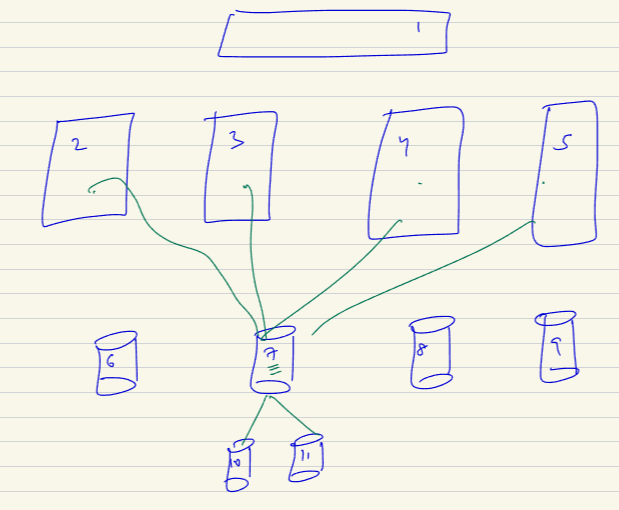
**Zookeeper Watch:**

Given ZK has maintain folder and files. ZK allows other member to

Machine id 1,2,3, …. 6.7.8,9,10,11….I have gwy.. . I have a file for each machine. I will allow each of these enties to keep

The file in 7 which people want to watch the file.. 10, 11. Should want to learn. If master dies.

Same way these app server want to know.. other entity can keep a watch on these files. Like observer DP.



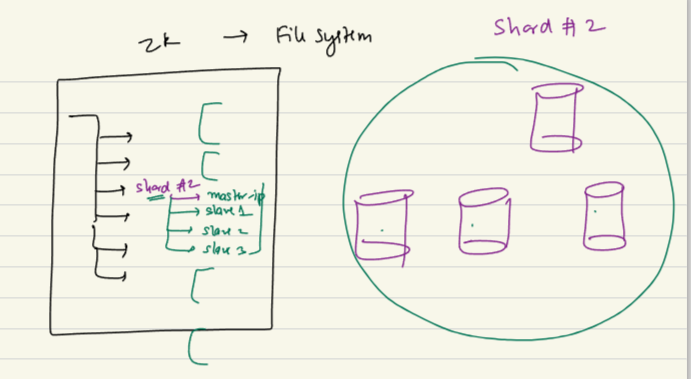
I enrolled myself if something happen to the file please letme know. Slave and app server will want to know.

Now if 7 dies, ZK deletes its file, ZK will look at the machine wants to know about 7. ZK will preemtively will tell the file yu are watching is no more. ZK will notify.

So the salve of machine will made aware that master is no more. So slave 10, 11 will try to rush and take a lock my master is dead make me master. Slaves has own behavior. Try to take alock. Sucecssfull oen will be master.

Everyone will be told this is master now.

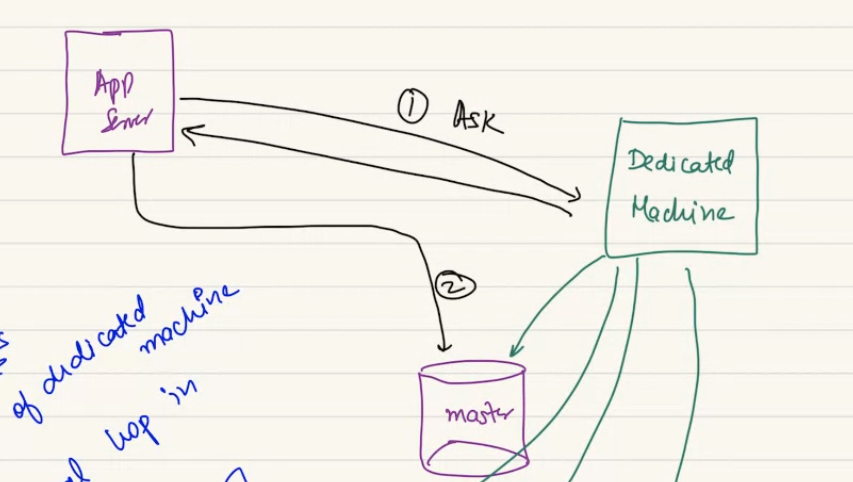
Each of these below represent shard 2..



For every node I will have list of watcher. They register themselves “hey I am interested knowing what happens the master..” salves, other shards, or load balancer of the DB who does consistent hashing.. app server also sets the watch. I read the value and keep using it. But when dies you lemme know.

Other member can set up watch.

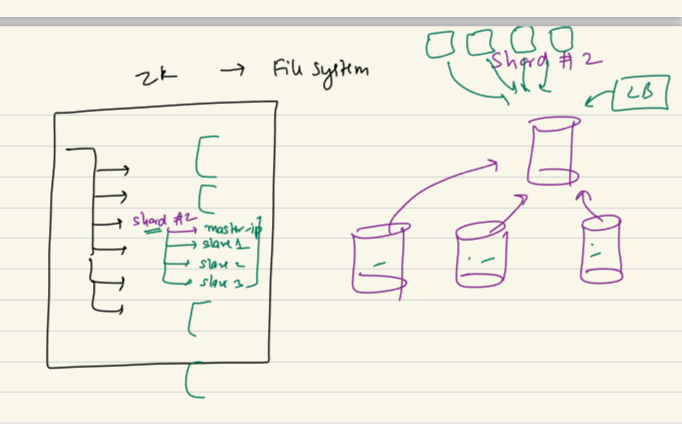
This allows the dependent entity who is master who is master. ZK responsibility to le them know.



We have changed this idea.

Here new system has a dedicated machine.. which is ZK. I want to set up watch on a file. When it changes letme know. Zk allows register myself to watch. Becomes a member of watchlist. If nothing happens no need to ask or ZK will inform.

If something happens.. ZK will let them know. No latency..

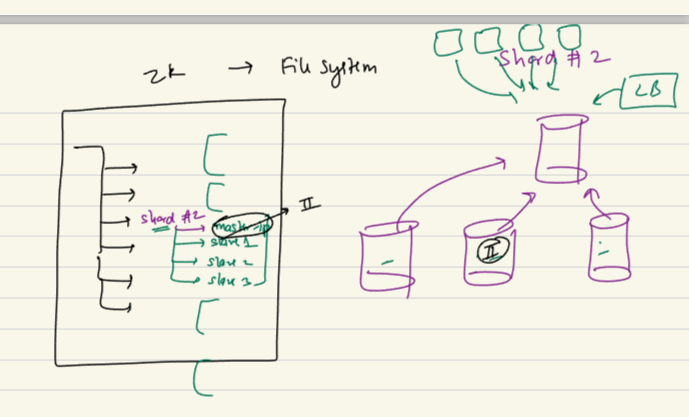


When master dies. ZK let slave know.

Leader electon algo. They aprticiapte. One become master. He will have lock over ephemeral. File Updates as 2.

When any file change ZK will inform all wastchcer. When a new file is created ZK will let them know.

I saved the time for watcher to come again and again.



1. ZK is nothing but a file system.
2. ZK files are called ZK nodes.
3. ZK nodes of 2 types. Normal persistant ZK nodes. Other is Ephemeral ZK node.
4. Idea of having an entity…

Each of the ephemeral file will have a real physical owner. When k heart beat miss I go delete it. Once I delete it I notify all watcher.

if i delete the file.. for a new master a new file will be created? A new file will be carted is owned/ locked by new master. Also that slave who became amster that salve file si deleted.

all slaves are watchers in case of master.. also other machine who interested.

All config is held

Single point of failure this problem is not solved. What if ZK dies..

Abhishek: 1.29: zk provide watch all dependent machine, app server/ salves. That case when slave try to have lock and write own ip. If watch is there other things can try to be master: no only slave. I configure them as. acts as their behaviour.. we set up ZK, configure it. We also configure. Slave machioen and app server will run ZK client. In app server I will bnot configure that they will not try to complete.

1.32: here master is aalso a controller? Controller election? We didn’t use controller term. Not kafka controller. These software use their term.. product wise its different. Kafka for state management or election. It piggybeg on ZK.

@Tarun: the slave takes a lock and try to register them as master.. when elected do they create a new file. or takes lock on an existing file.. please clear this out.. "takes a lock on the file" meaning?

**He will share the content of whole class of ZK..**

Pramod: when master goes down.. all slave reaches to ZK. Zk allows one to master. Can be fastest finger fast or an algo.. to ideantiy best master. Only one will have lock on newly created file.

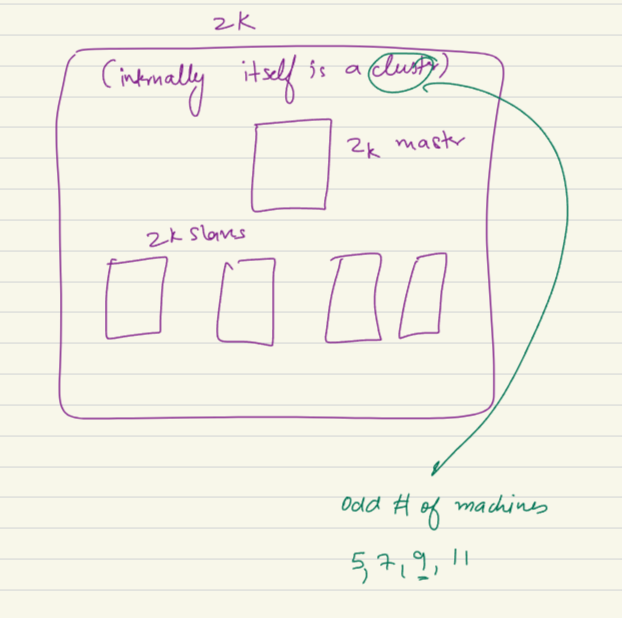
I can throw and build my solution.. is not zk we need some solution to do this..

Will all slave rush to be a master? If a slave is dead he cant go an compete.

Prateek: how hashing is implemented? App server has IP addes, don’t know which ip I want to go. CH can be in client side.

ZK created in an manner it can handle SOF very well. ZK doesn’t have just one machine. ZK itself internally is a Cluster.

The way the cluster is created there si a ZK master node. There is some ZK slave nodes. ZK is a cluster which is created an idea that the cluster will always have odd number of machine. 5,7,9,11 etc. usually 7 or 9 are most popular set up.



In ZK master same structure wil ebmaintained in slaves. When files needs to be written/ modify it comes to ZK master. ZK make an update on all slaves. When

Zk gets a write, tryies to write on all. When a corum of slave replies back about the slave. More than a threaosld of amchie. It is called Quarm.

Reason for having is even if oen machine dies other machine can keep the job done.

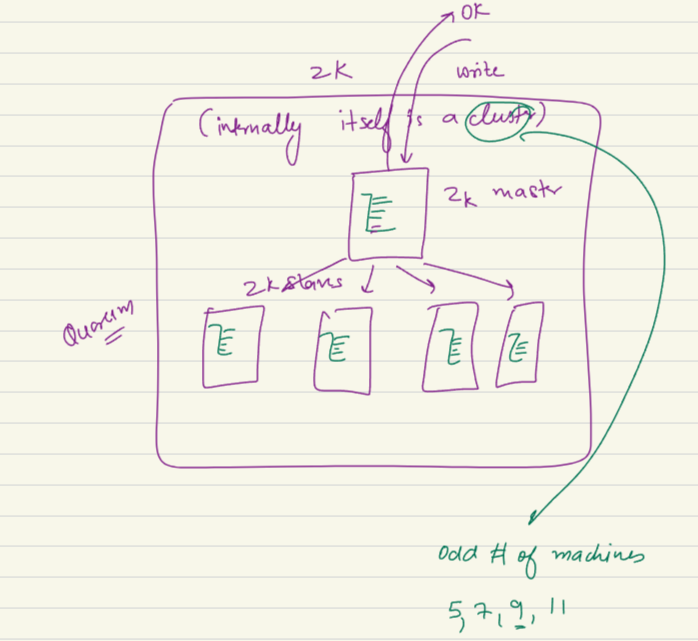
Quarm:

If a I have amchie 1 and have slave machine. I try to wriote on all of them. Unless more than half reply positively.. I will not tell job is donw. Only when more thnhalf reply back..

Master will wait for slave to give confirmation. When they give confirmation..

Reason is.. even if someobudy dies. Still majority will agress on same set of answer. Realibility of system increase a as I am consistent.

Whatever update on folder saructure. For ZK the file/ folder update that come to me..



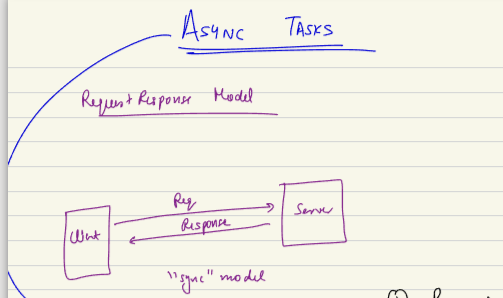
To manage a cluster I created ZK. ZK has itself master slave..config. this is still fien reason is one ZK manages entire world. All layer being handled by ZK. If ZK master dies, all of my communication update will stoop. ZK will have again a ZK master elctrion algo. They will use Gossip Protocol. Talk to each other, gossip to each other. A horse in ashamed jogya. That idea they used..

Each slave will gossipe. And one slave will be master. Once the new master is up everything else will function properly. No need to worry ZK master election algo.

**ASYNC Task:**

As of now we saw request -response model..

A server, I have a client… app server can be a clientto DB. An entiry who is goingto maker a request and an entity who gives response..



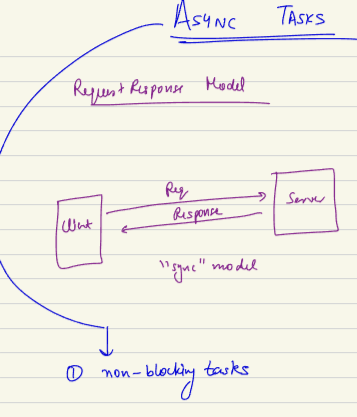
This is request response model or sync model. In sync manner sending a req and getting response.

This model not possible always.i cant make the requester waiting till a job gets done.

This idea not possible always to block myself and wait and send response in sync manner.

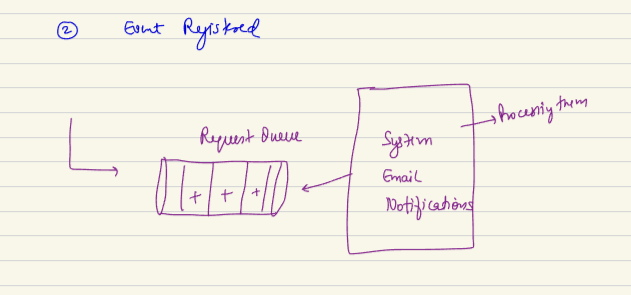
Hence there is alternative model async model .. idea of not blocking a task. Not going to wait on eresponse, I will register myself and wait to get a response.

There is a event that I registered, that will be handled later I will not wait.



Email sending system need to be cognizant to something. There is a limit, service under some constraint. Rather system to design in a manner client send a request and response right away instead imagine in a menner I have a request queue in front of me. Whoever want to send a request they add a request in my queue. In my good time I kep on reading these request and keep on processing them.

Rather than a req-res system which is sync manner rather I have a non sync manner…I create a qeueu I take it up. Not at that point of time necessarily.



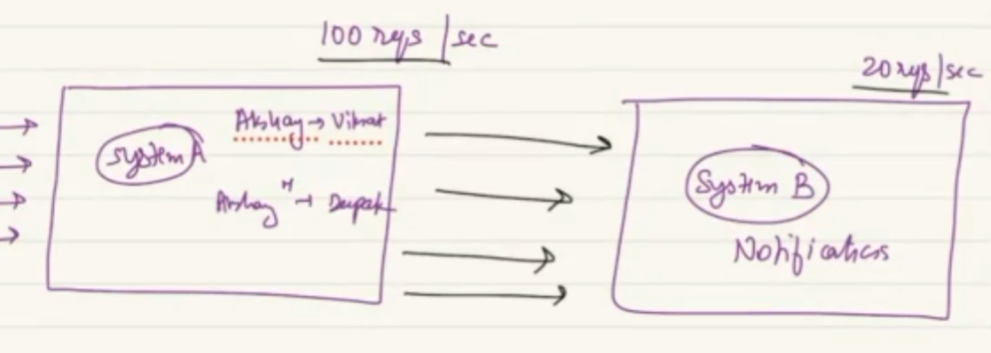
If I miss something in slack, I get a digest email.. any notification system..

**NOTIFICATION SYSTEM:**

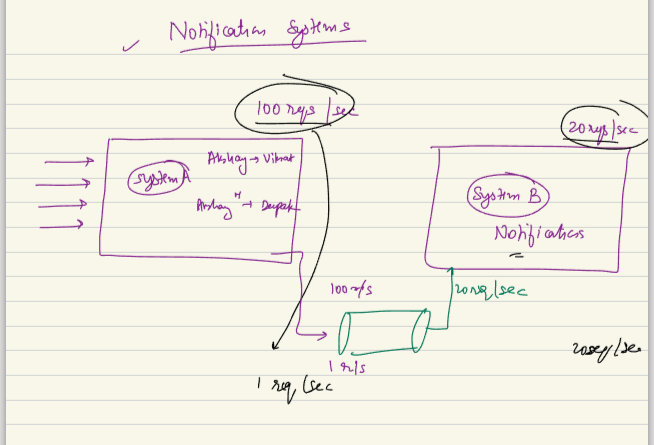
Lets say I have 2 systems. A and B.

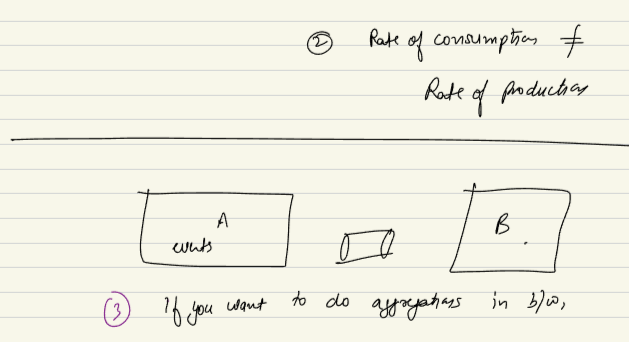
In A I have 100 request coming per second. System A is the gettihg 100 request. B is sytem which send snotifictaion for these updates.. B can handle 20 req/ second..

If each of them need a notification. 80 of req will fail..



If it is not the rate throughout the day. 100 req/sec can be 1 req/second.. than rather than making it direct call I can create a queue in between where there is a a event I populate. At times I can populate 100 req/sec. throughout the day B can keep consuming 20req/second. Heavy load in beginning can be handled in rest o fthe day when load is less.





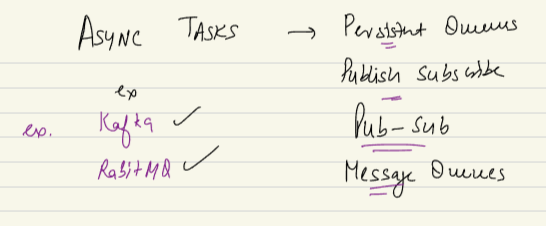
3rd use case: system A registers all events done throughout day. B need to keep entry/record throughout day. And send 1 email at end of the day. This email cant send in req-respon/ sync manner. Email need to send async manner. All event need to be collect and send at end of day.

If you want to do some agggregations, in between aynced task can be used. I am collection whatever event user doing throughoutthe day. This async task we decided, it is done using the idea of Persistance Queue.

Aka Publisher-subscriber,

Pub-sub, message-queue etc..

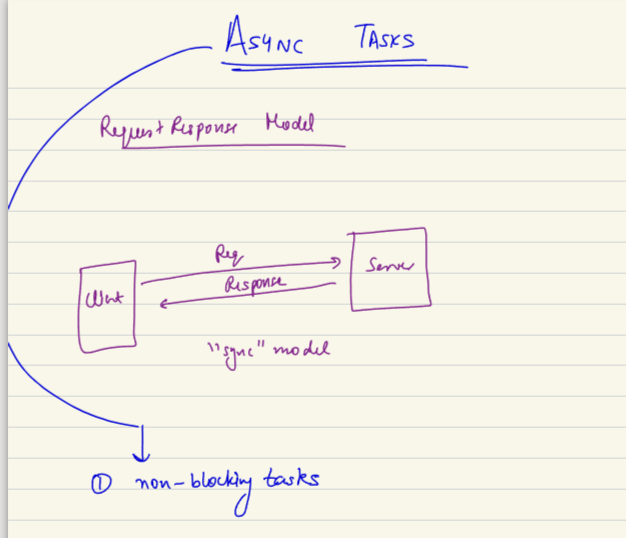
System which behaves like this is Kafka, Rabbit MQ.. main 2 things



Redis can also behave like this..

Redis is a db, acts as a DB. Also acts as a Cache. Redis also acts as a Message Queue. A system can be tweaked to do these things..

kafkaa and rabbitmq dffrnt messaging model. Underlying idea of async task is same..



No need to wait, will get done.

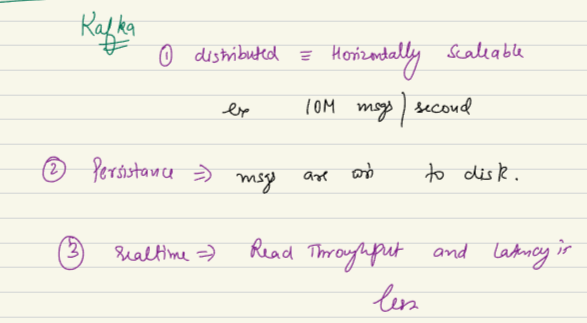
You register on a evernt. Will will happen later . you don’t need to wait oin.

rabbit mq is point to point but kafka pub sub model as per me

**Kafka: just a message queue. Work as pub sub model.**

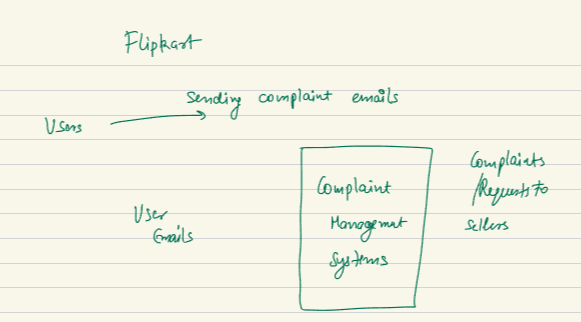
Have following principle:

1. Kafka is a product which is distributes in nature, means I can scale them horizontally. I can scale my Message queue horizontally. Means Kafka can support upto 10M message in a second. A event that are qritten in kafka queue.
2. It is a persisted queue. Persistence menas, once the message are written cannot be deleted. Messages are written in deisk. Not only in primory memory.
3. Kafka has the feature of real time. Read throuput and latency is less. We can read a lot of messages.



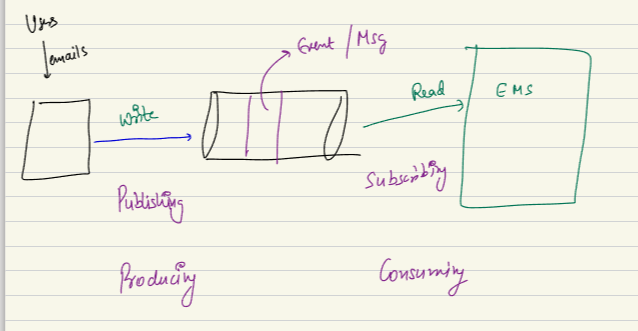
Example :

I am flipkart, I have the idea where my users who are sending complaint emails.. or raise request. There is a system in between which is complaint management system who process the emails or use human intervention. Does process, to resolve them it sends request to its sellers. Req or complaint forwarded to sellers.



The complaint management system, the email that come to it may come at any rate. At some point thousand or very few. The complaint management system cannot scale up or scale down. So its not possible to do in sync manner. So we will process in async manner.

Suer send a email. Received to the system it adds message to the queue, pushes to queue. EMS can read the message at its own pace. .. the idea of writing to the queue is called as publishing an event. And idea to reading from queue called as subscription of message, or producing an event or consuming n event. Entities can called event or a message.



Machine who produces : producer

Who read: consumer.

The thing in between is Kafka queue.

Kafka is a distributes system, I can employ diff machine. I can achieve a lot higher parallel process. Distributed and scalable system

**How this work actually?**

When I have a kafka queue or any message queue. It means I have a queue from a physical point of view, collection/ cluster of machine. Its actually not built on single machine.

Redis also cluster of redis machine same for mongodb.

The entire cluster is called kafka, same cluster can be solved multiple problem simultaneously. I can have same cluster in my company and can work on different use case.

If there is different Topic (come later)

A queue handles order, another queue which manages all refund. Imagine there is a user. Who creates an order which a message should get inserted in order queue. Same way a refund message should go in.

When multiple people are interested who want to know if the order is created. Logistic service might want to know if order is placed. Acts as consumer. Same way the refund.

Same way seller service will also be interested in knowing whats going in order queue. Also analytical service wants to know about order.

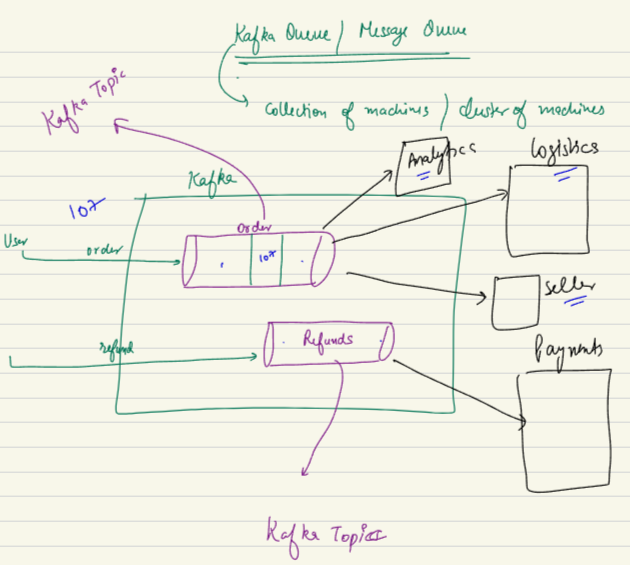
Logistic, seller analytics want to know.

So when order 107 is created the same message whould be read by all 3. Each one of them should get the message one time and should all of them read it. Also there is no duplicacy..

How to make this happen??

Each of these independent queue are called Kafka Topic. Order is kafka topic. Independent queue inside kafka is topic.

A queue name in kafka cluster. There can be 1000 queue are called independent kafka topic.



**How to Read from Queue:**

I have 2 topics/ queue.

When a producer write a message to a topic. Once I write to a topic I basically create a new message id. That message gets stored inside disk. Say message id are 107,103 etc. in FIFO manner they are going to be stored in queue. When producer makes a message I will put in topic and persisted in disk

Some consumers they can register themselves. I want to register myself for this topic. Please register me I am analytics I want to rerad from this queue/ topic.

So analytics, seller can register.

Kafka will create a state to know what is the last message analystics, seller from me.. till tyhsi point he read this message from me. In the beginning the offset set os 0.

When someone wants to read, FIFo queue I see offset. I will give him 1st message. And make offset as 101.

So when someone comes I check offset and return 103 and make offset 103.

Whenever a service comes say seller service I want to read. I see offset as 0. So I give 101…

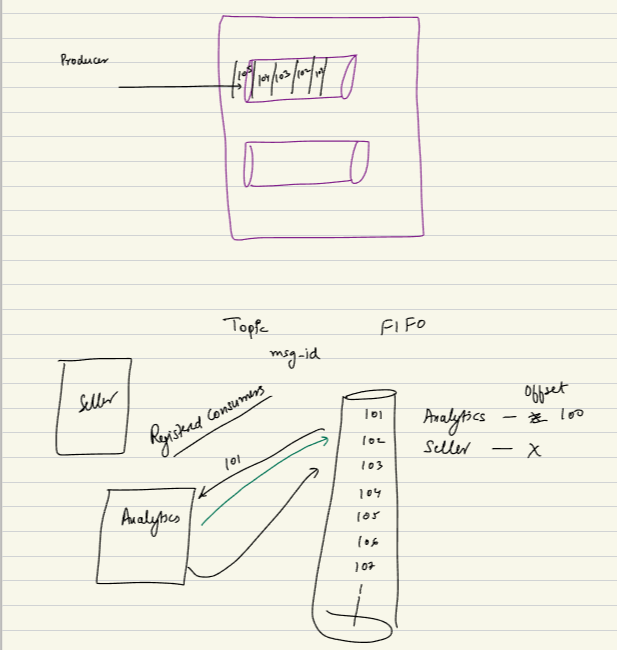
Measn the messages are not deleted when someone reads from me. Gonna stay. Which means my queue will always going to keep growing. To solve this. Every message inside queue has a TTL.

Say I have 4 consumer. The slowest reads once in 4 hours. Person who reads sloest. If I keep an TTL of 12 hours and slowest reads 5 hours . I am confident that there is enough mergin. After that even if someone don’t read it, I will not wait and delete. So kafka machine manages offset for each machine and present next data point. The offset keep gorowing. I keep TTL and after that time message s gets deleted.

Means devops team has to change the TTL or increase temporarily. Keep higher TTL gives devops team a time.

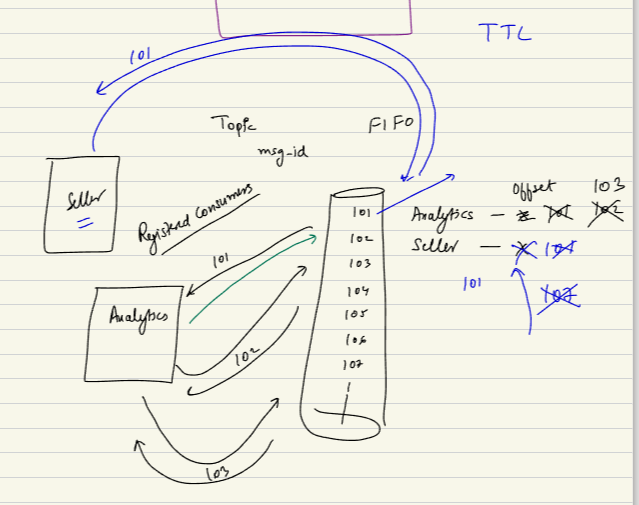
Pople misunderstand that you do not delete the message..when somebody has read it.

You only delete when TTL has expired even if someone is yet to read it.



Given I have queue and seller service offset if 107. Devops team relizes that last 5 message seller read they didn’t process them properly or made mistake.. while processing them

Seller team can come and revise the offset.. set it from 107 to 101. only if ttl has not expired till then



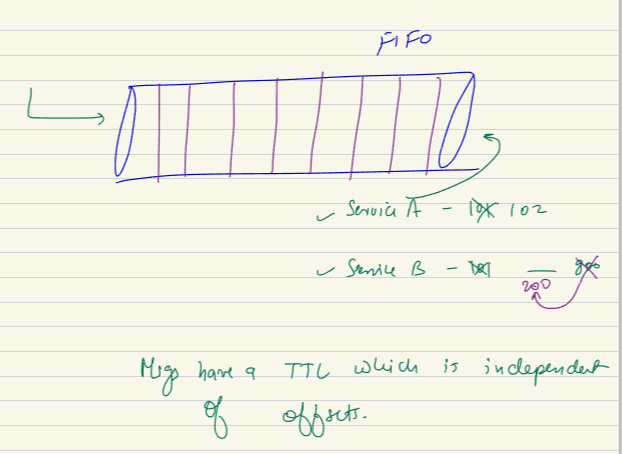
Akfka q is cluster of machine.. inside it we have many small topic. They can have multiple consumer. They can come and register themselves as consumer. Kafka queue as a FIFO queue. You can have multiple consumer who can register itself as a consumer..for eah of the consumer I can mainataina offset. Which means this is last message they read..

All the message will get deleted even if no one read me or everyone read me. Only get deleted based on retaintion policy. The TTL is indepent of offset.

Kafka stores offset of all service. Its configuration system. So kafka uses ZOOKEEPER (a confighuration management system)

As a service whenever I consume I give next message as offset and it will be consumed. Guarentee of every message will be read once.

If I made a mistake I want to red a message once, I can reset the offset again if the TTL is allowed.



ZK maintain confugration for me..

For evry topic I will set up service level configuration.. service A read till this far..

what can message or event contain exactly in order example?

If I have a queue, every message has some detail that my consumer will read. An order will have order id, timestamp, order details, user id. Conatin some info that I want my consumer to read from it.

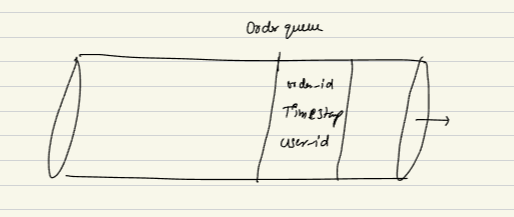
Overload message and store all info

Another school of thought that to make message lighter.. and kkepp only id. And theat can call from DB.

If I store everything in message, bulky message, but no extra hop.

If Only ID, light m,essage, an additional hop to get infor from SQL.

A message can store only info.. but API model not possible. I will not put PAI. But info so later PAI call can happen.



How will producer know there message si read.. as of now no ack.. if I want my producer to know that their message consumed. I can create a request-resposnse set up..

I can have a different message queue that can put in ack queue and servciec can read…

Its fir-forget model.. producer fier and forget, if I want producer toknwo…

API can be called or a queue to push the info and producer can read..

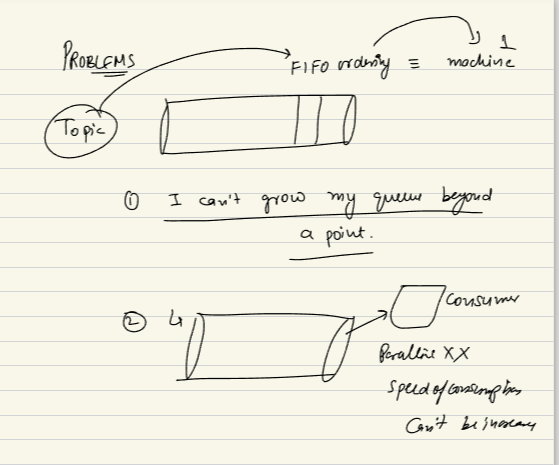
Message Queue out of box do not give any ack to producer. Producer onlyu know they produced but don’t know..

Can you tell us any other real time example of Kafka?

Dominos uses it

**PROBLEM to this solution:**

Problem are two folder, given I have a quue, and it needs FIFO ordering, means I want to store all mesaage in one place in order. This means that I cant grow my queue beyond a point.



For a topic isngle mach im writing.. and one consumer who is reading..

Speed of consumption cant be increase, a single place where I am reading one by one..

A service an read oen by one as FITO manner. I cant have thread. As I have offset.. it will grow one by one.

Single underlying machine, when a topic is FIFO queue:

1. Read one by one, I cant parrellize, yes many service can parelly read.. I need to write all message in single machine.
2. As writing entire topic in single machine. Cant store so many message..

Will solve using partitioning.. a new concept..

Kafka is async

Replication like anyotehr rather storing in one..

No one will ask zookeeper, kafka will very important..

Yes if you metion inresume they will ask

kafka can be configured to be not FIFO like SQS ? yes u can.

Any resources for Kafka you will provide to learn from ? he will give video links…. Today..

Lot info coming in 2 days

**QnA:**

1. But files are stored on hard disk and not in main memory. Will it not add to latency to read from disk.

U r right. File are stored in 2nd dary. Adv is persistent. Wont loose out. As there is name but writing will increase. We can ||ize read. Also have some adv.

how were things handled before a zookeeper ? I mean can we have a distributed system without zookeeper?

A districuted system can without ZK. ZKis produced system to get job done. We can manage usually LB acts a s config machine.

So will this dedicated machine do sharding?? ZK internnaly cluster. M-S. rather than doing sharding as data not big. No sharding but replication.

Shouldn't the master node files be persistent nodes as master going down every now and then is not very frequent activities? They need to handle in sofication manner. When heartbeat don’t come, fille will be deleted. Ephemeral nodes will alow to do all management.

Protocoal deiofned by ZK, ZK clients will send ping.

So basically every system will subscribe for each file maintained by zookeeper? No every system will not watch every ZK file. Not sub to file of every sngle mach.

how is hashing implemented if ZK hop is skipped and app server is directly accesing the master of shard ? hashing will done in client/ LB side. Zk tells IP.

zookeper is after orchestrator and consistent hashing , is my understanding correct? No. ZK itself is my corchester. CH happens before it.

I didn't get the spof of zk, zk cluster and how these zk cluster get managed ..

sir are kafka and rabbitMQ related please throw some light? Both arte pub-sub, both are MQ. Just diff priority are set. 2 diff products.

API Gateway works in sync or in async manner? Synmc manner.

Topic ihave will have limitation..

TTL set at topic level

If TTL does not allow to configure offset how the service is going to take those messages. If TTL expired and I want to reset offset, I cant do it. TTL says messge will kept for this time. Set it offset before deletion. You cant reread it.

when consumer's pace is slower than producers, queue can become full then how the new produced messages can be handled? will they be dropped??? Yesa, q filled, no space means it will be deletd. But producer rate is higher for some amount of time. But iof always higher than consumer. Then there will be a pointto drop message from QUEUE.

Tarun ZK is a cluster internally, so now whichever client interacts with ZK, will it go thru a Load Balancer? No, not multiple shard of ZK. Just M-S. for a write go to master. For read go to slave directly, no need for LB. donthave multiple shards.

Zookeeper Qtn master dies does it mean master can become unresponsive because of some deadlock ? yes. Master can unresponsive, didn’t send heartbeat ans assume dead.

does every shard have one ZK each or one ZK can manage more than one shards? One ZK canmanage every shard, also every layer. Everryting end to end in my system

when you say Kafka can handle 10M mesages in a second. does it mean entire kafka cluster has a max capacity of 10M.? can be more than that. With a reasonalble cluster. Topic Can handle 10M.

any good reason kafka is more popular than zeroMQ / RabbitMQ?kafka has big connunity . gives guarentee at one time reading. Great throughput.

Over weekend: keep a track of announcement grup. He will share.. revise kafka.. video link…