Lec1

Q) what is a distributed system?

→ a system where I can't get my work as somewhere any computer has crashed which I never heard of (Joke)

] by father of DS

→ It's about failure.

Martin Kleppmann's defn. author of Data Intensive Applicat

→ running on several nodes connected by a network

"an instance of some software running on a computer in order to accomplish some task"

→ characterized by partial failure.

(It means → some parts of the system might be broken while other parts of it are okay

and furthermore it's often impossible to (atleast impossible for the software) to know what exactly is broken.

Partial failures can be:

→ Either a computer failed

→ or Connection b/w two computer broke / failed

This is different from the kind of failure that we are often used to thinking

or a single machine if part of OS fail. then all fails. and there is no way to operate.

but if you have many machines then the part of keep operating. Even an failure is very imp

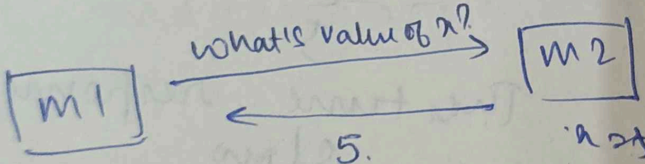
(restarting all machines is not a good option)

" The more machines you have its more inevitable that some systems will fail "

Cloud Computing v/s HPC Computing

Cloud Computing	HPC Computing
→ working around partial failures.	→ treats partial failures as total failures.
	→ checkpoints involved for long run.

Case 1- We have two machines.

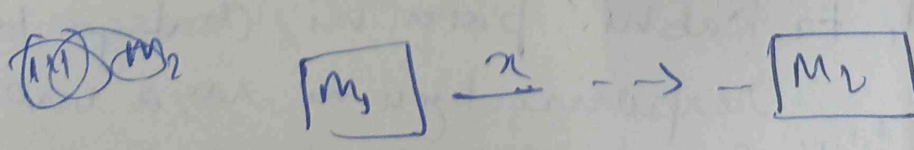


Is simple req-response model mai kya kya dikkate/failures aa sakte hi?

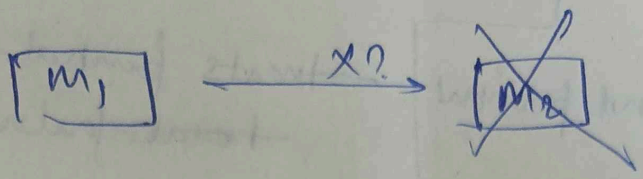
(i) Req from M_1 gets lost

(ii) Req from M_1 is slow (M_2 hamesha ke liye wait thori kar sakta h)

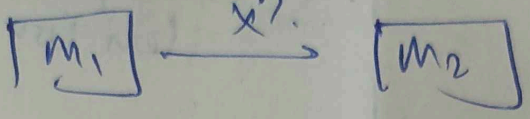
↳ Reason can be "congestion" or "may be a managing queue"



(3) M_2 crashed

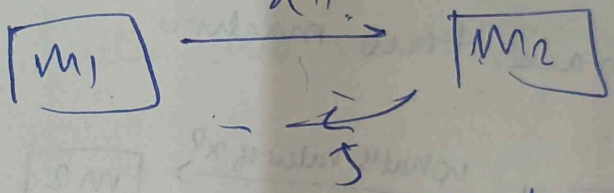


(4)



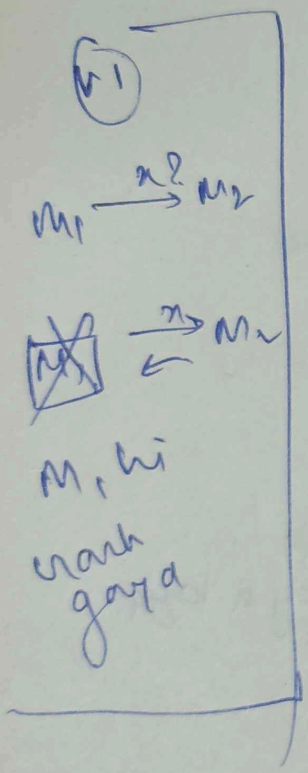
M_1 sends, but M_2 is slower than M_1 expects, so M_1 assume I would not get a response.

(5)

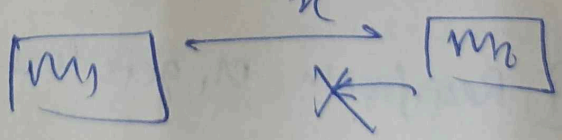


This time response is slow

(6)



(7) response is lost



All these conditions are different but they are same from M_1 perspective.

M_1 to kabhi pata hi chalega ki response kyu ni aya unke liye saare situations same hi h.

"So, if you send a req to another node and you don't receive a response it is impossible to know why"
(without global knowledge of the system)

So, how do real systems deal with this type of situations?

If one machine wants to send a message to another machine and it doesn't know for sure if it's going to get a response.

So what one should do when sending a req and expecting a response.

In reality M1 should have some sort of timeout.

Ye kya h?

after sometime when M1 has sended req and if there is no response then assume failure.

It is a one of the best approach in reality.

PTO

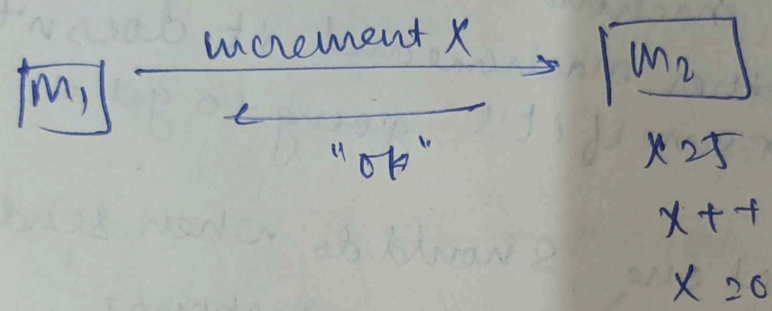
But why It might be a mistake to assume failure

↳ Abhi to dekh sirf

M1 n kyah whi puch raha tha,

par are situation ka kya jaha.

M1 ka req M2 ko ~~effect~~ affect kar raha?



Agar M2 ok ni bhg paya (ya timeout se phele w aaya), to ye mana ki n++ hua galat aur naa mana bhi risky h.

and this sort of uncertainty is (or inevitable) fundamental characteristic of distributed system.

Distributed System = partial failure + unbounded latency

So in Distributed System we have to deal with partial failures and unbounded latency and that is what makes Distributed systems so hard.

By why you like a system like this that is cleanly very terrible, hard to debug, to build, why would you want it?

→ data too big to fit on one machine.

→ You want things to be faster. So you use more computers.

→ for redundant copies to fail independently.

Time and clocks

What are clocks for?

(i) Mark points in time.

"This class starts at 9:20 am"

(ii) durations or intervals of time.

"This class is 65 minutes long".

time-of-day clocks

- Sync'd with NTP (Network time protocol)
- bad for measuring duration & intervals. because here time can jump forward & backward.
- Don't use for implementing timeouts
- ok for timestamps but not great

monotonic clocks

- only go forward
- bad for timestamps
- good for durations, intervals & timeouts

Lec-08 :- GBS

Ye dono physical clocks h. (Peeche page wale)

Physical clocks.	logical clocks.
Time of day, monotonic clocks.	only measure ordering of events. which event happened before another.

Suppose A happended before B

↳ A ⟶ B

Inference / Causality -
- 'A' could have caused 'B'
- 'B' could not have caused 'A'

"My style of teaching might be influenced by Sir teaching but Sir teaching was not influenced by my teaching" not sure afterwar