Today

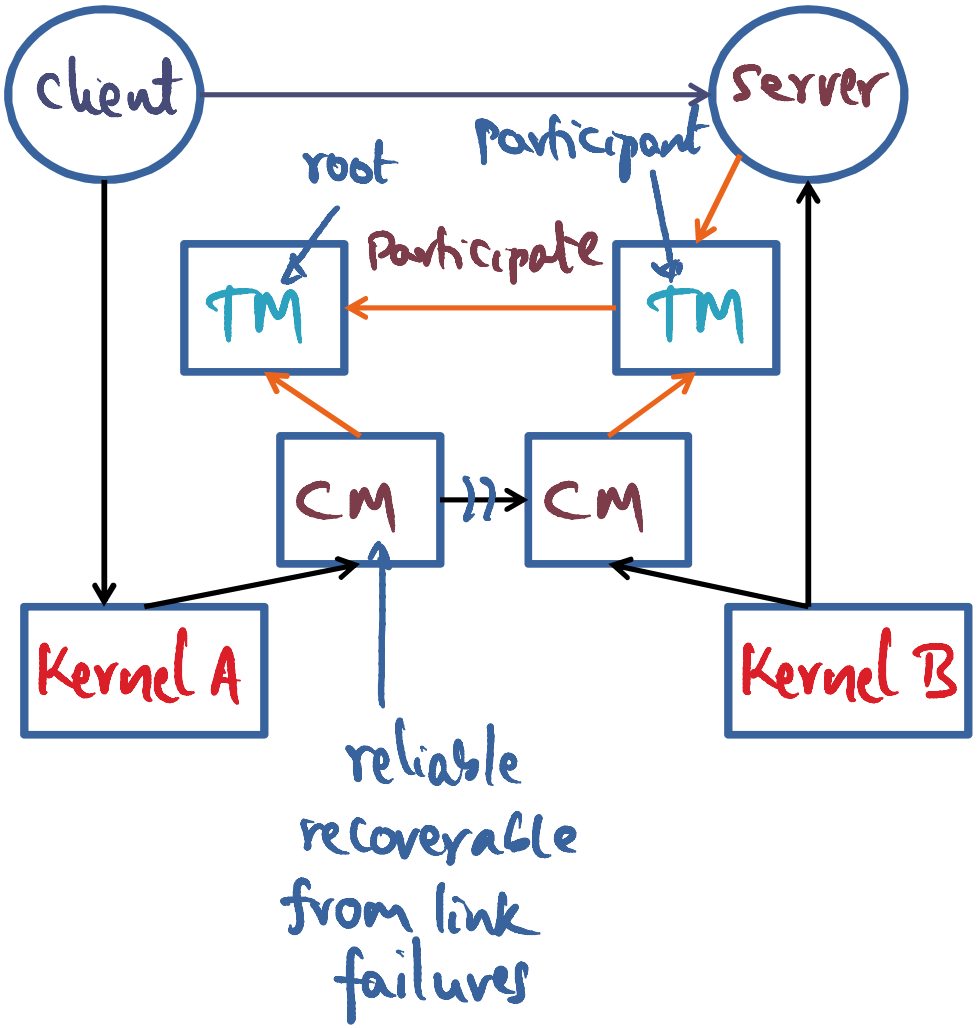
* Quicksilver
Internet scale computing
(Lesson 9)

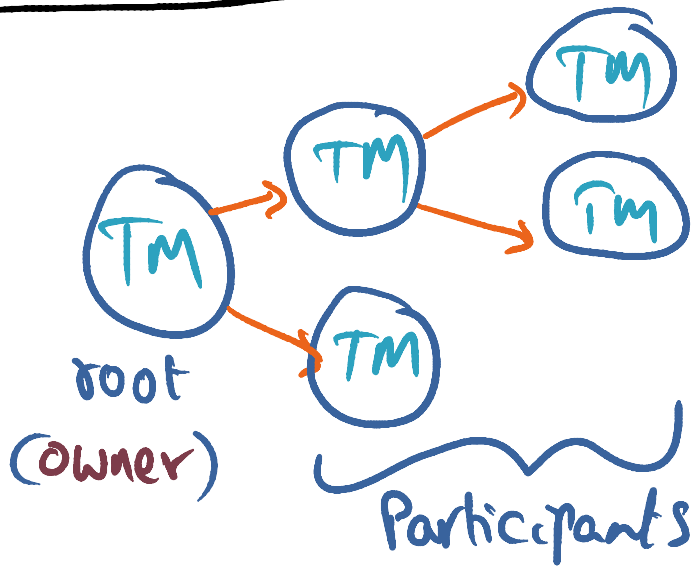
⇒ * giant scale services
* Map-reduce programming model
* CDN (Coral)

* Please watch videos before class *

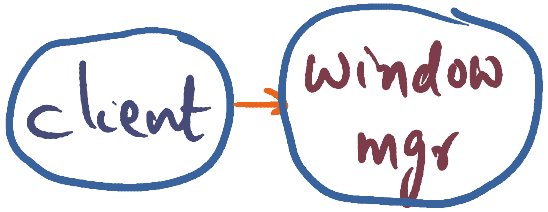
Bundling Distributed IPC & xactions

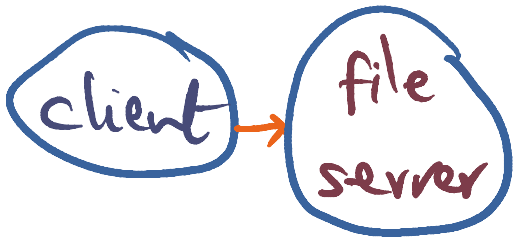
Transaction => Secret Sauce for recovery management





Examples

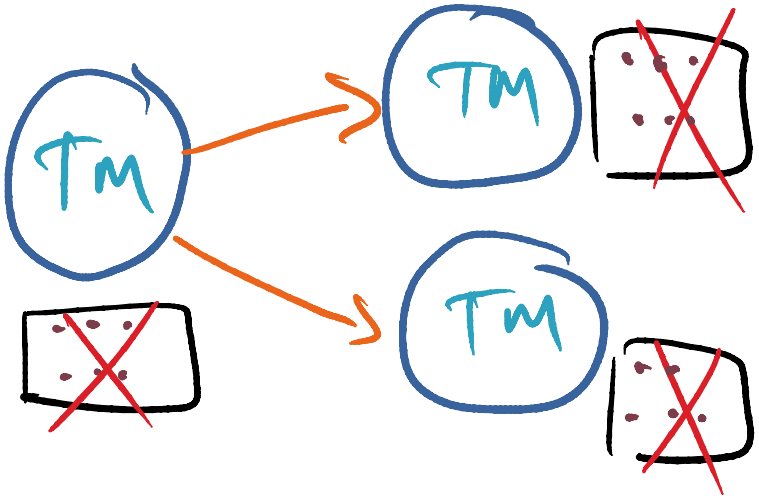




upshot of bundling IPC + Recovery

Reclaim Resources

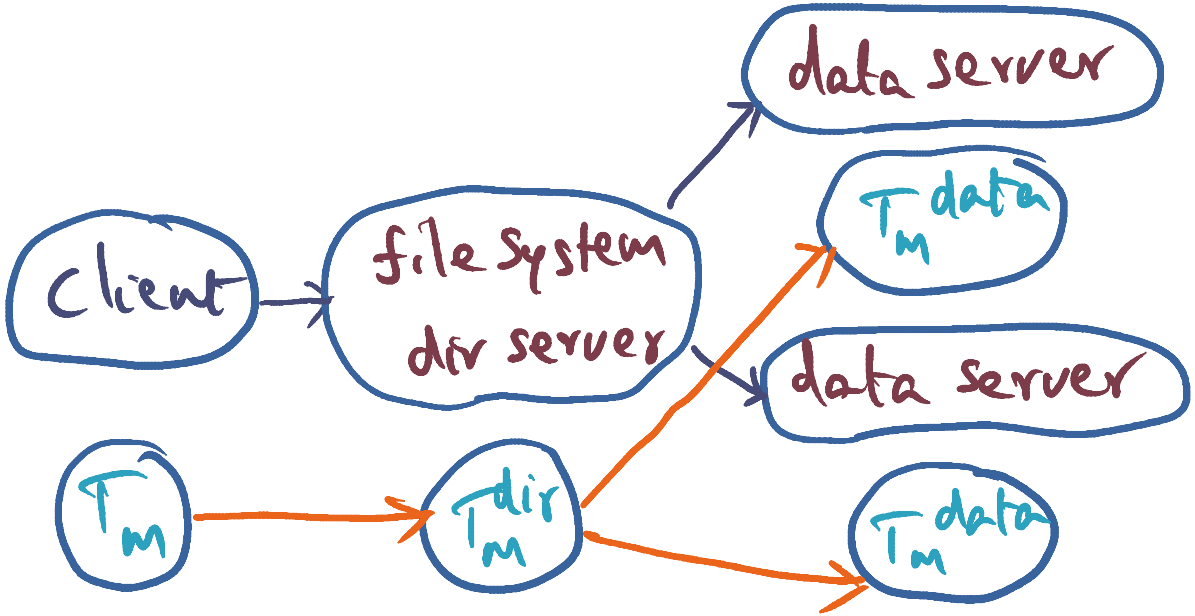
- breadcrumbs left behind by failed clients/servers



Examples:
memory, filehandles,
Communication handles,
orphan windows, ...

Transaction management

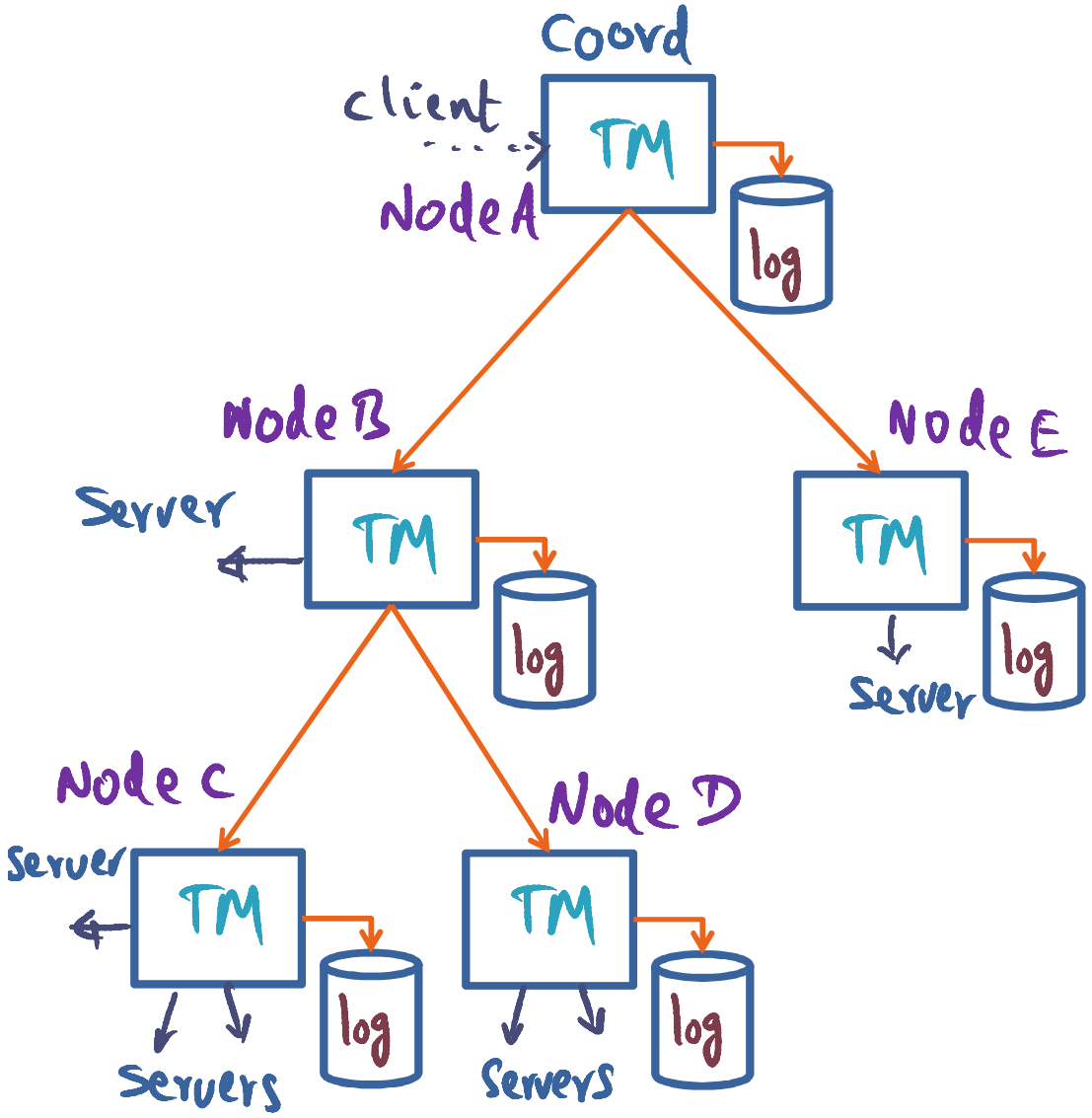
Coordinator can be different from owner



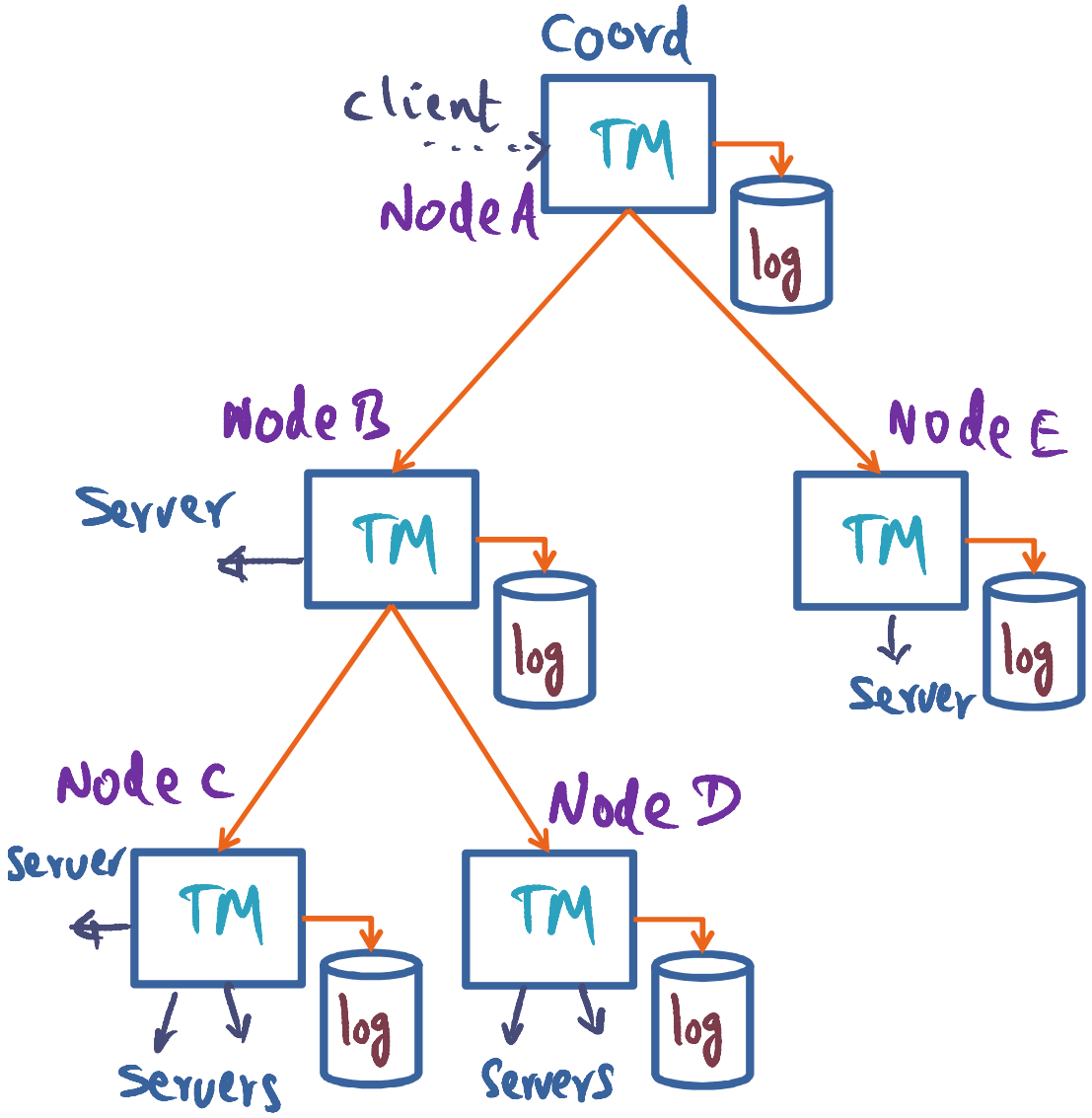
Owner

Participants

Distributed Transaction

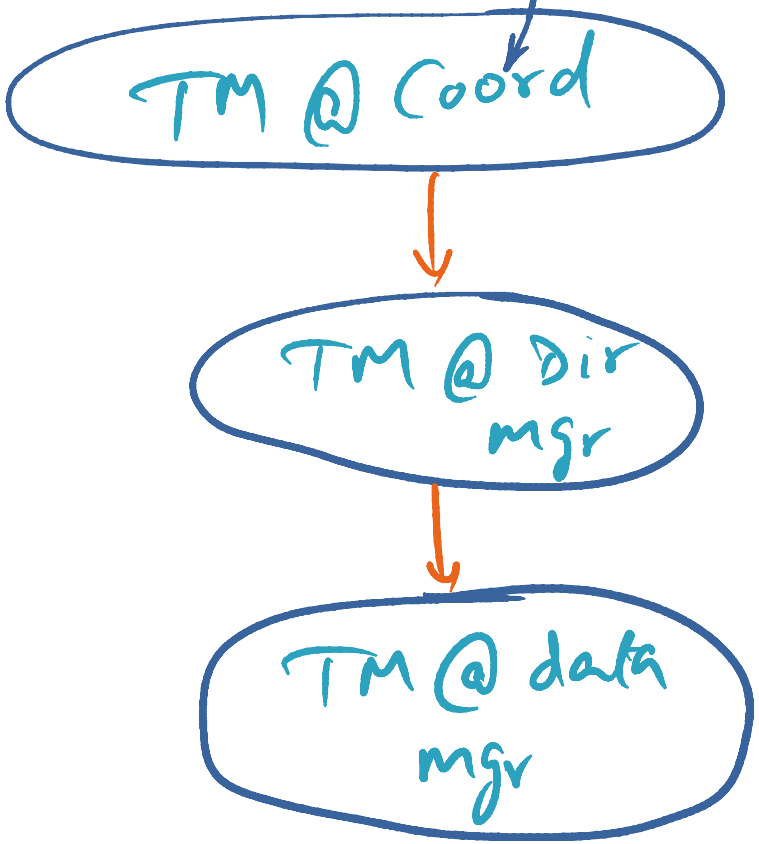


Distributed Transaction

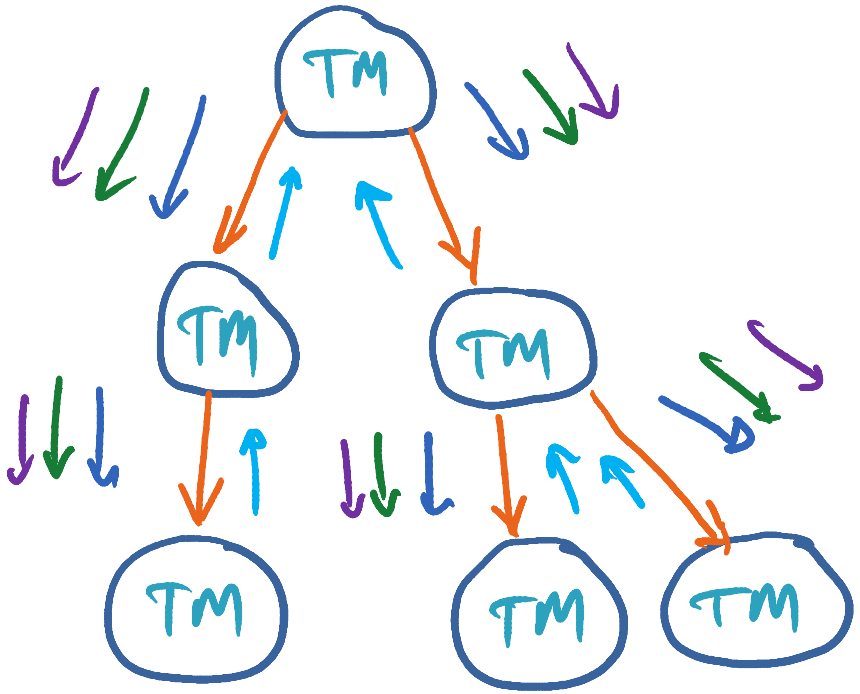


All TMs not equal

- e.g. file server



Commit initiated by Coordinator



Down the tree

→ Vote request

→ Abort request

→ End Commit/abort

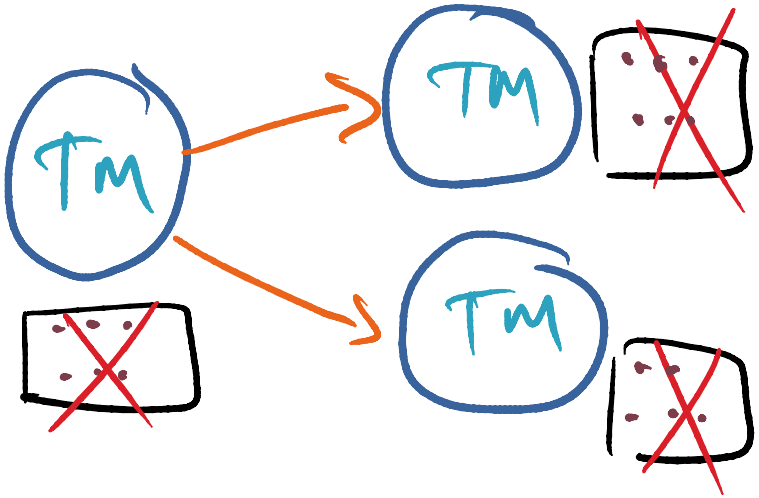
Up the tree

→ response commensurate with request

upshot of bundling IPC + Recovery

Reclaim Resources

- breadcrumbs left behind by failed clients/servers

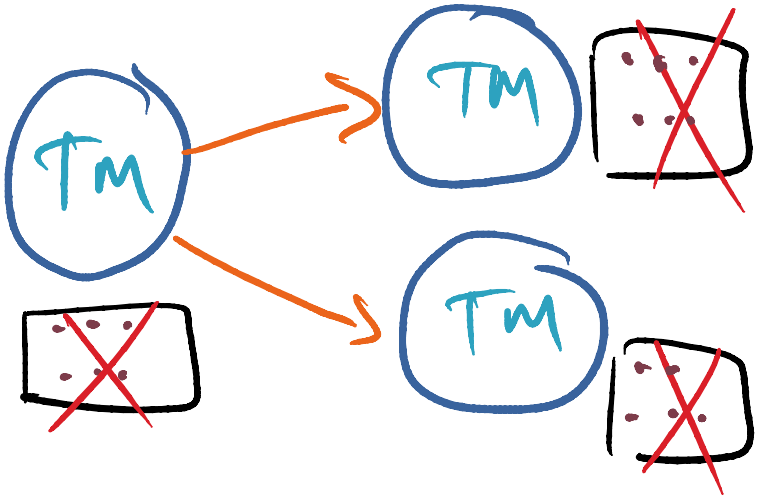


Examples:
memory, filehandles,
Communication handles,
orphan windows, ...

upshot of bundling IPC + Recovery

Reclaim Resources

- breadcrumbs left behind by failed clients/servers



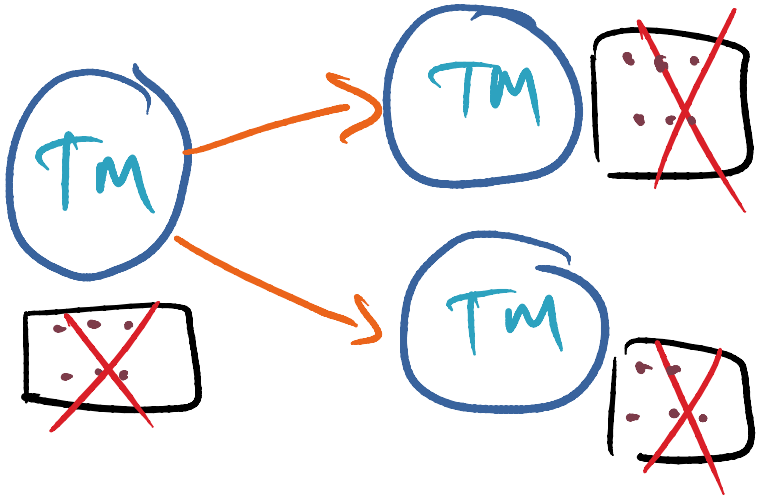
Examples:
memory, filehandles,
Communication handles,
orphan windows, ...

NO Extra communication for recovery

upshot of bundling IPC + Recovery

Reclaim Resources

- breadcrumbs left behind by failed clients/servers



Examples:
memory, filehandles,
Communication handles,
orphan windows, ...

NO Extra communication for recovery

only mechanism in OS, policy up to each service

- low overhead mechanisms for simple services
- weighty mechanisms for services such as FS

Implementation Notes

Log maintenance

- TMs write log records for recovering persistent state
- Frequency of "log force" impacts performance

⇒ Services have to choose mechanisms commensurate with their recovery requirements

Key Takeaways

- Transaction as a fundamental OS mechanism to bundle in state recovery of OS services…resurgence in the 90's in LRVM for providing persistence.
- Again in 2010, resurgence in the form of providing safeguard against system vulnerability to malicious attacks in another research OS called TxOS…a later module on system security.
- Computer industry still focused on performance…reliability always takes a backseat…nasty things go on under cover in OS to gain performance!
  - You write a file and you think it is on the hard disk…think again! It is a while before your file write actually gets persisted on the disk…what if the system crashes in the meanwhile?…too bad!

Future?

- New kinds of memories called "storage class memories" that have latency properties similar to DRAM and yet non-volatile.

Will this new technology result in a resurgence of exploring transactions in OS again? Only time will tell…