
DRAIN: Distributed Recovery Architecture for Inaccessible Nodes in Multi-core Chips

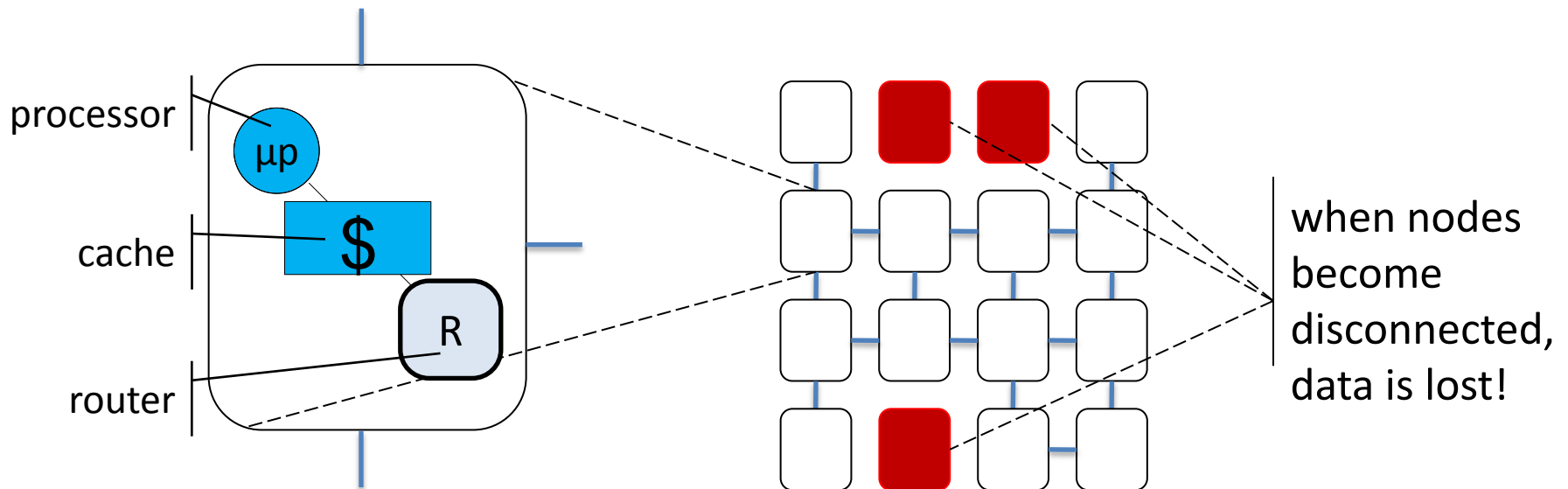
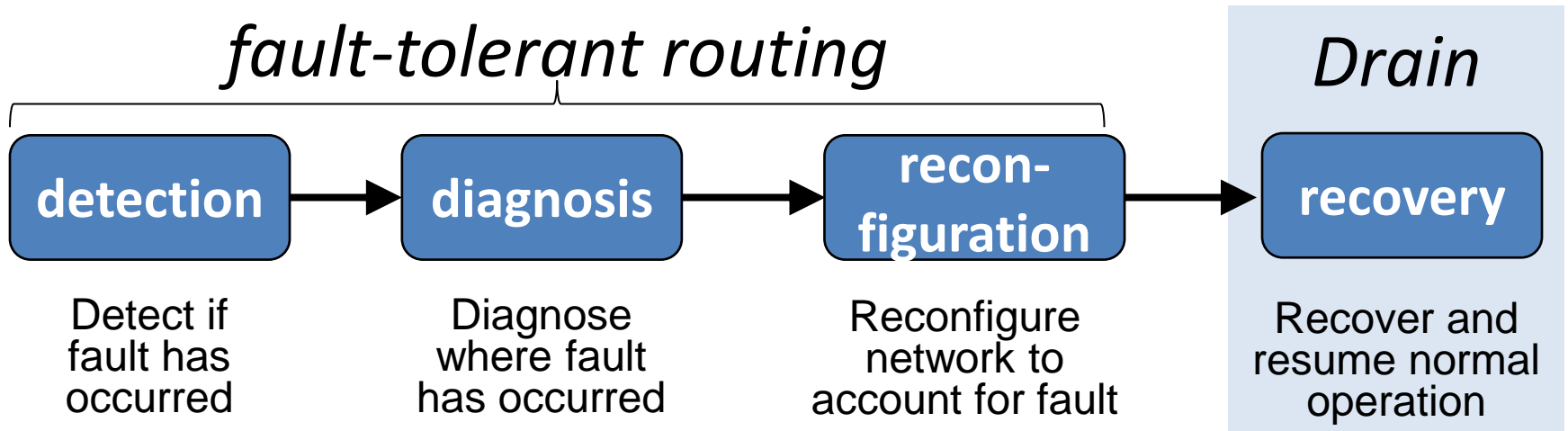
Andrew DeOrio[†], Konstantinos Aisopos[‡]§
Valeria Bertacco[†], Li-Shiuan Peh[§]

[†]University of Michigan

[‡]Princeton University

[§]Massachusetts Institute of Technology

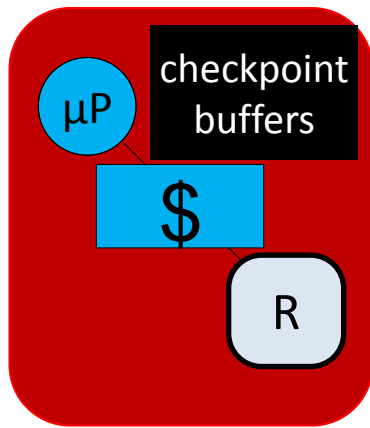
Reliable Networks on Chip



Previous Recovery Solutions

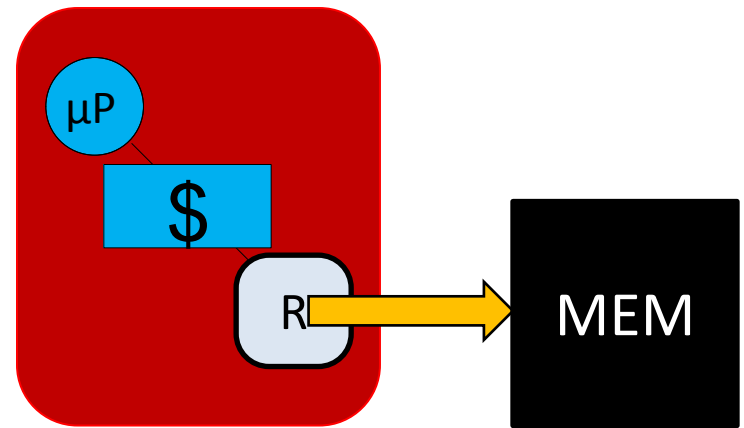
- Checkpoint approaches

ReVive [Prvulovic et. al'02]



data stuck in
checkpoint buffer!

SafetyNet [Sorin et. al'02]

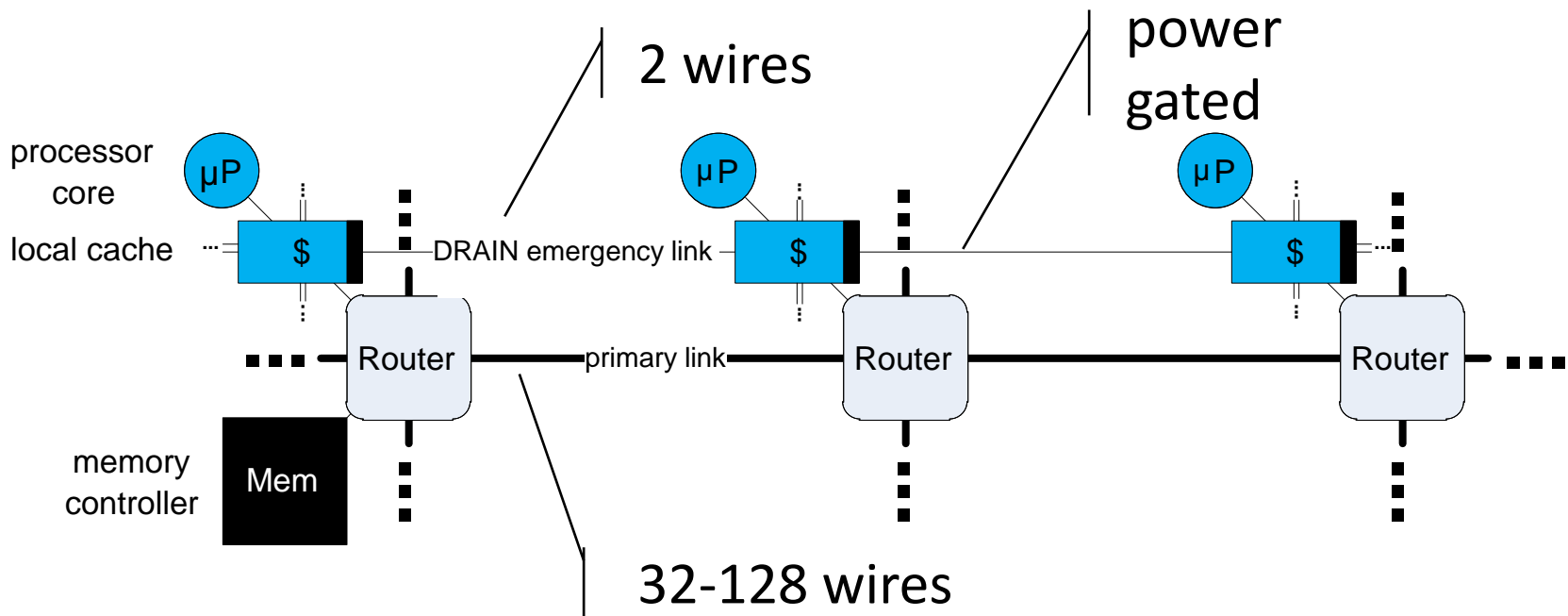


high performance
overhead!

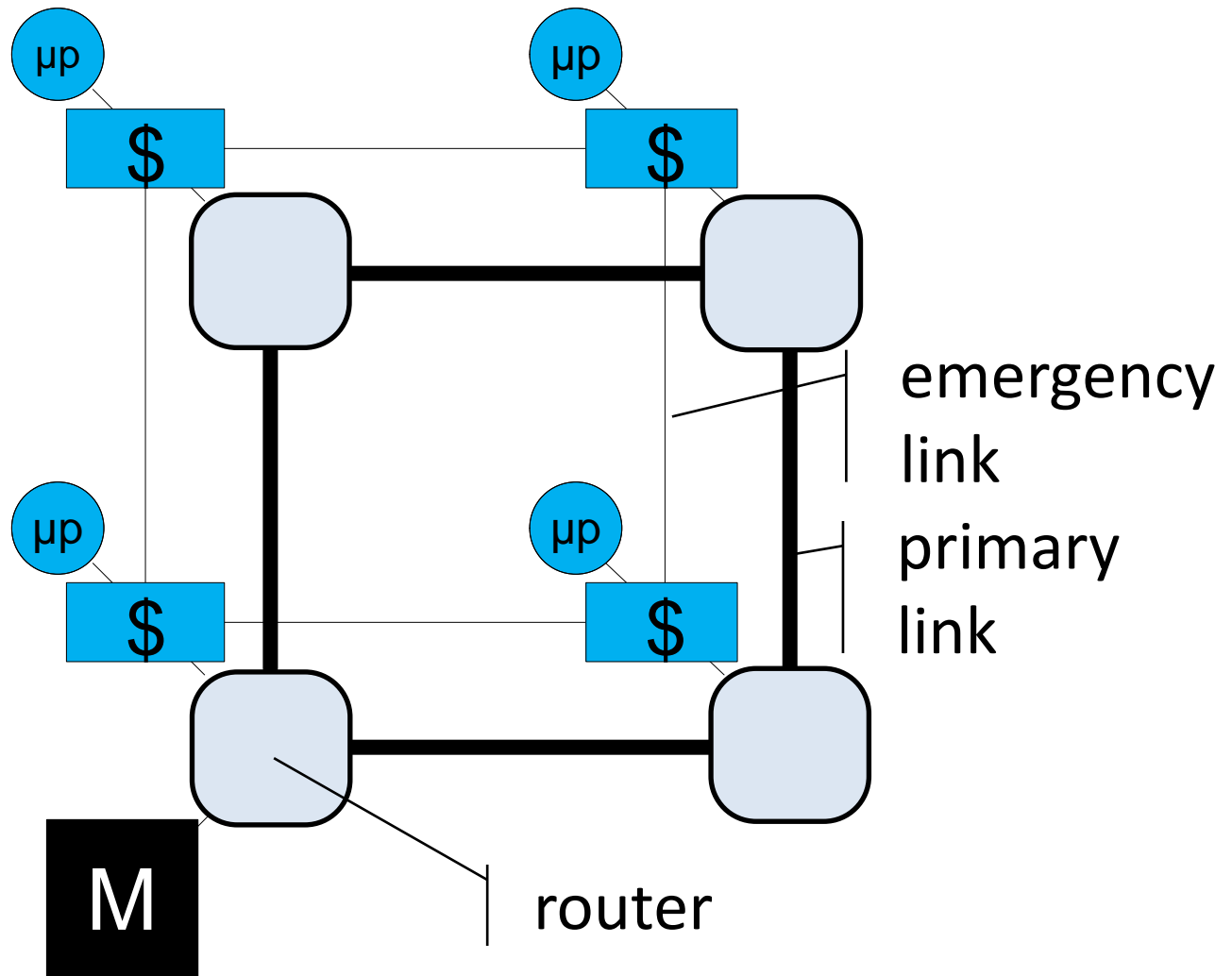
- Drain takes a **reactive approach**, incurring performance overhead only when errors occur

Data Recovery with Drain

- Recover data lost during reconfiguration
 - Emergency links provide alternate path
 - Transfers cache contents and architectural state

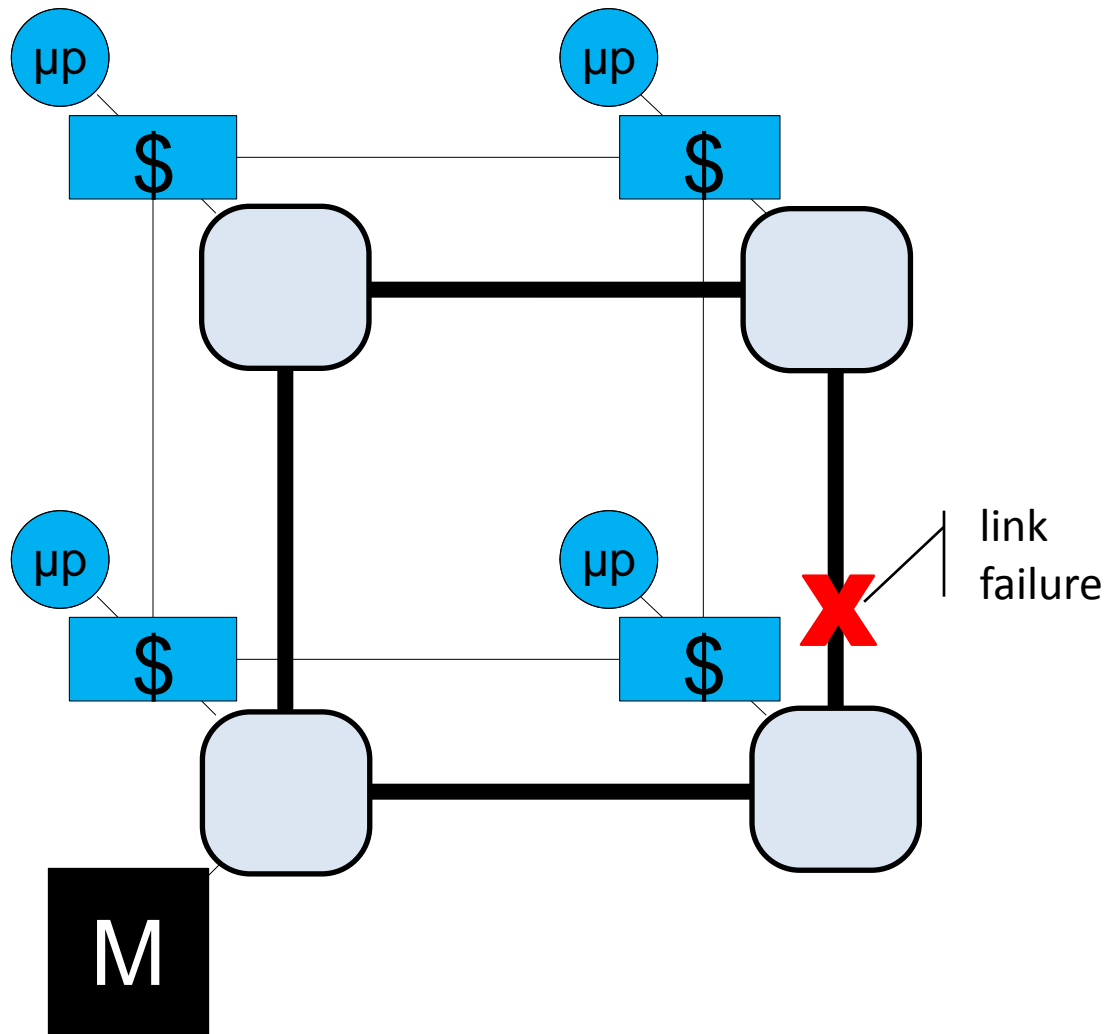


Drain Example

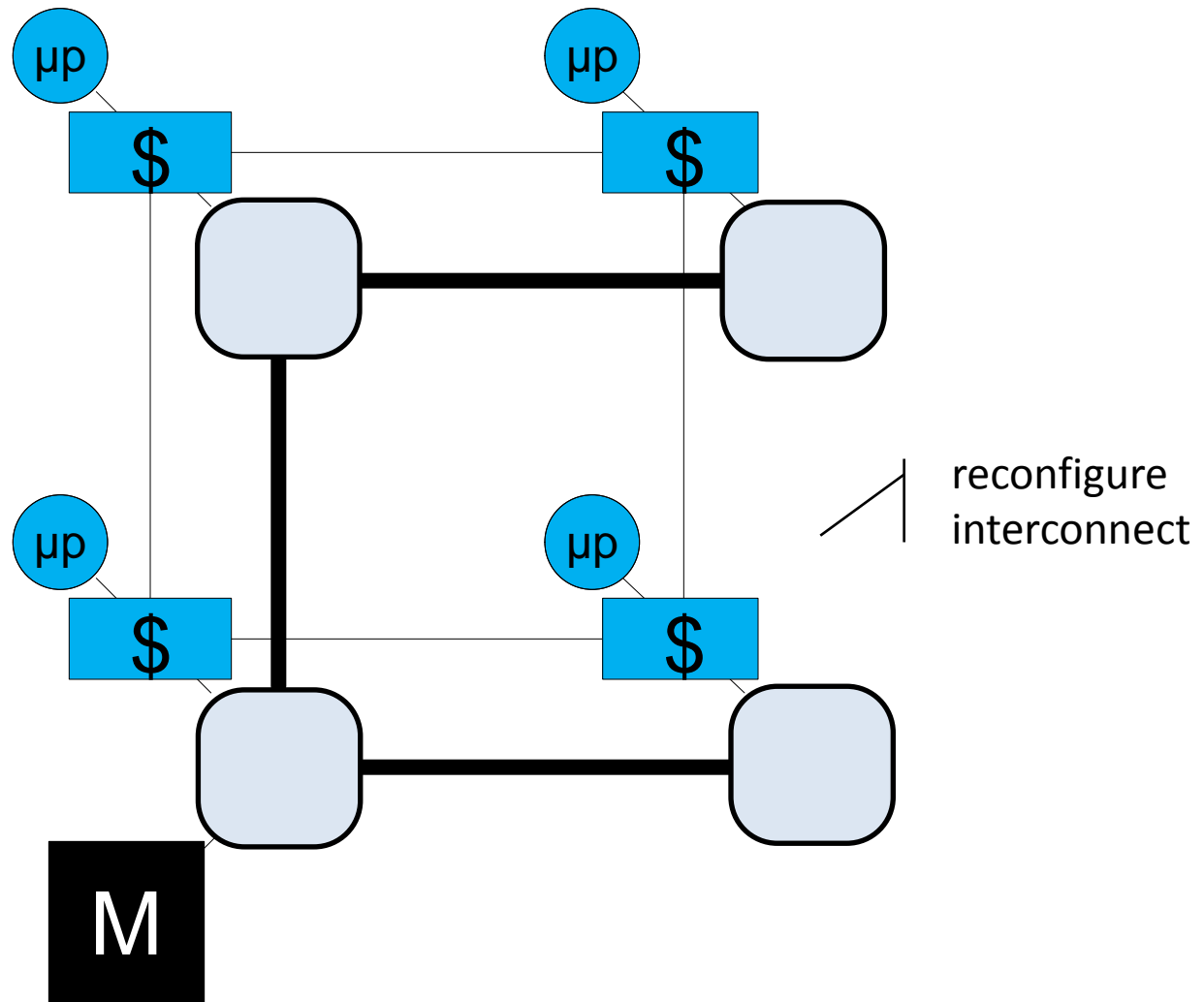


Drain Example

Fault model:
faults accumulate
one at a time.

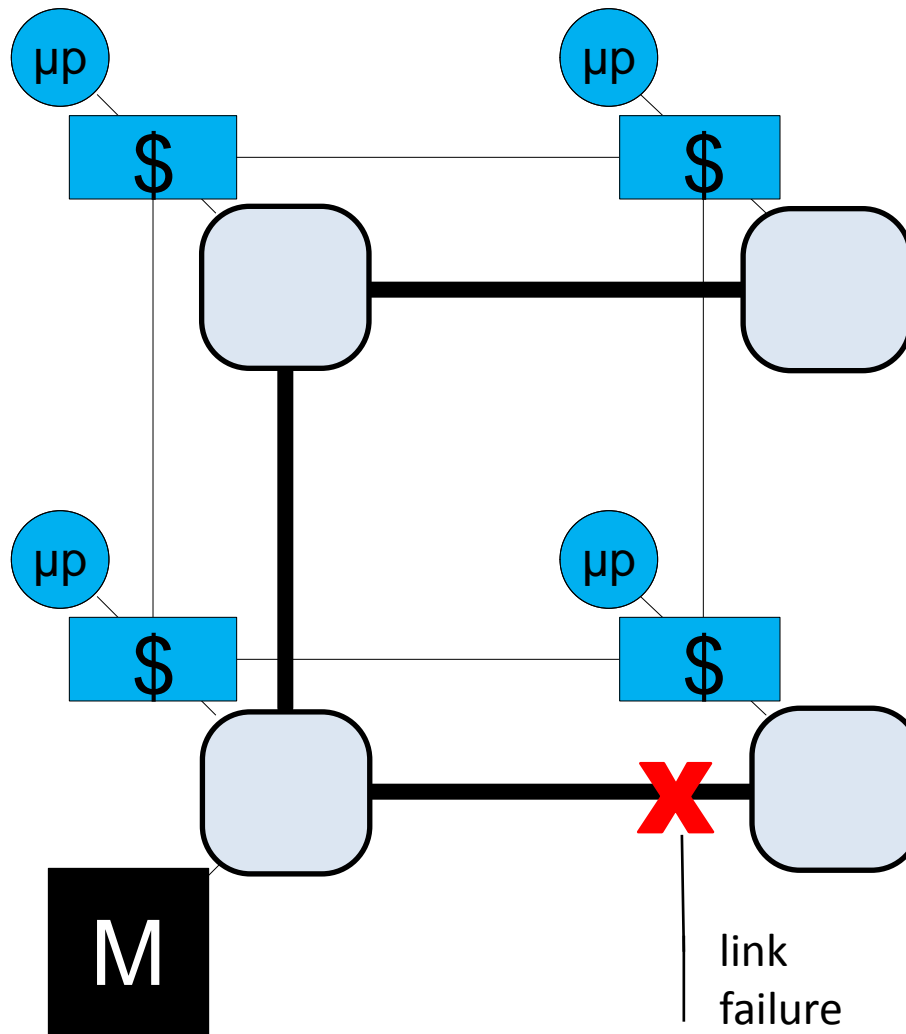


Drain Example

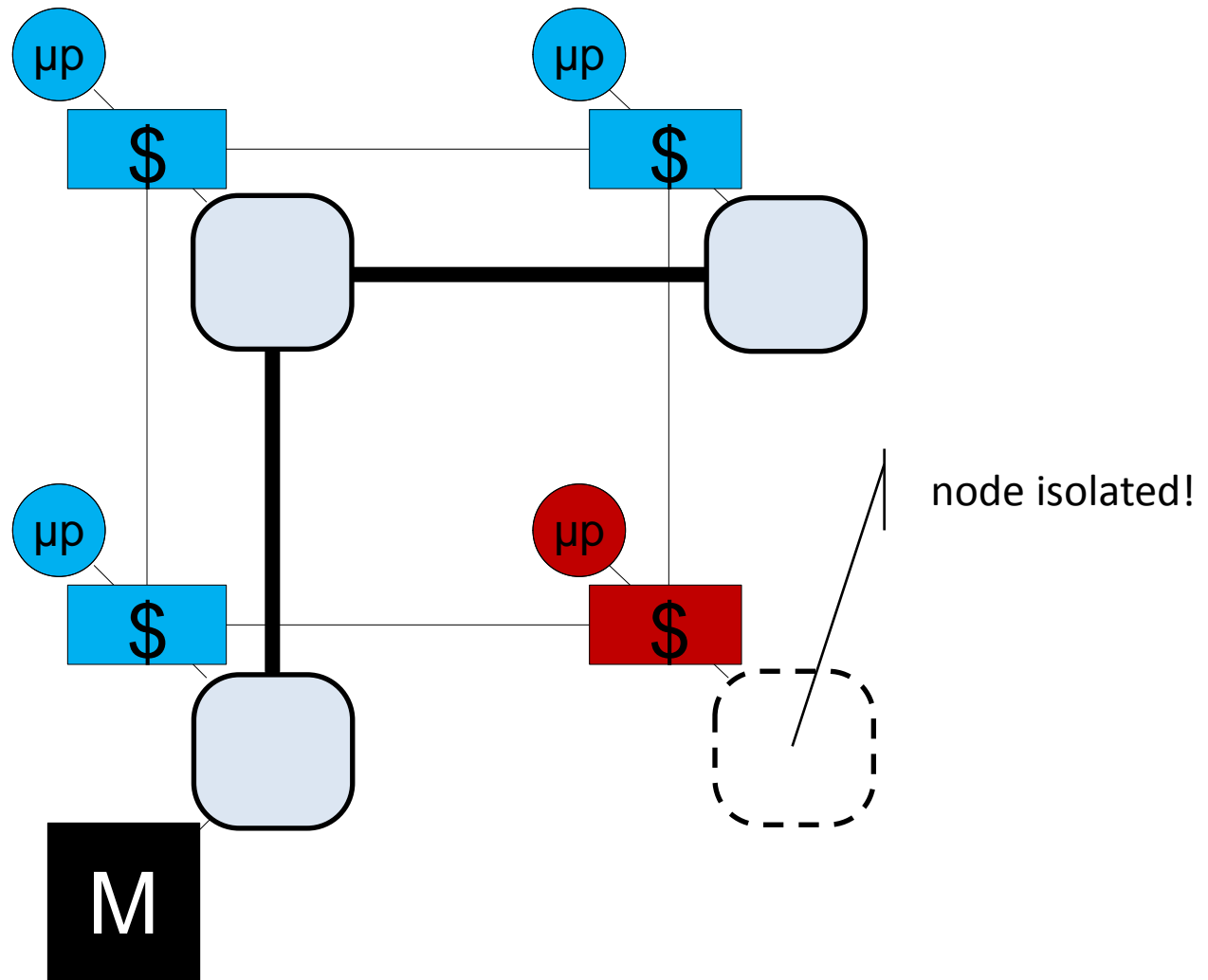


Drain Example

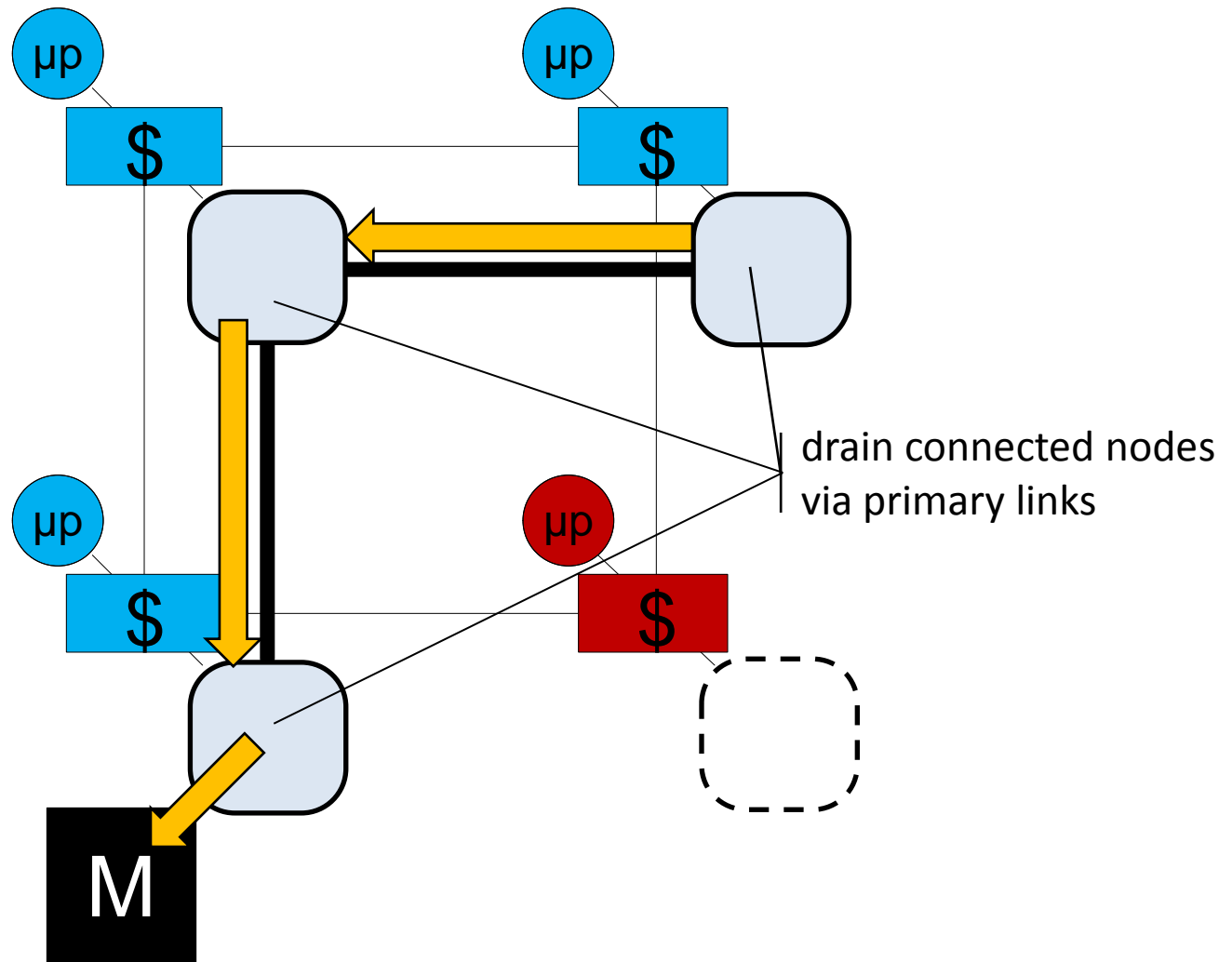
Fault model:
initiate Drain
recovery when a
single additional
fault causes a
node to become
isolated



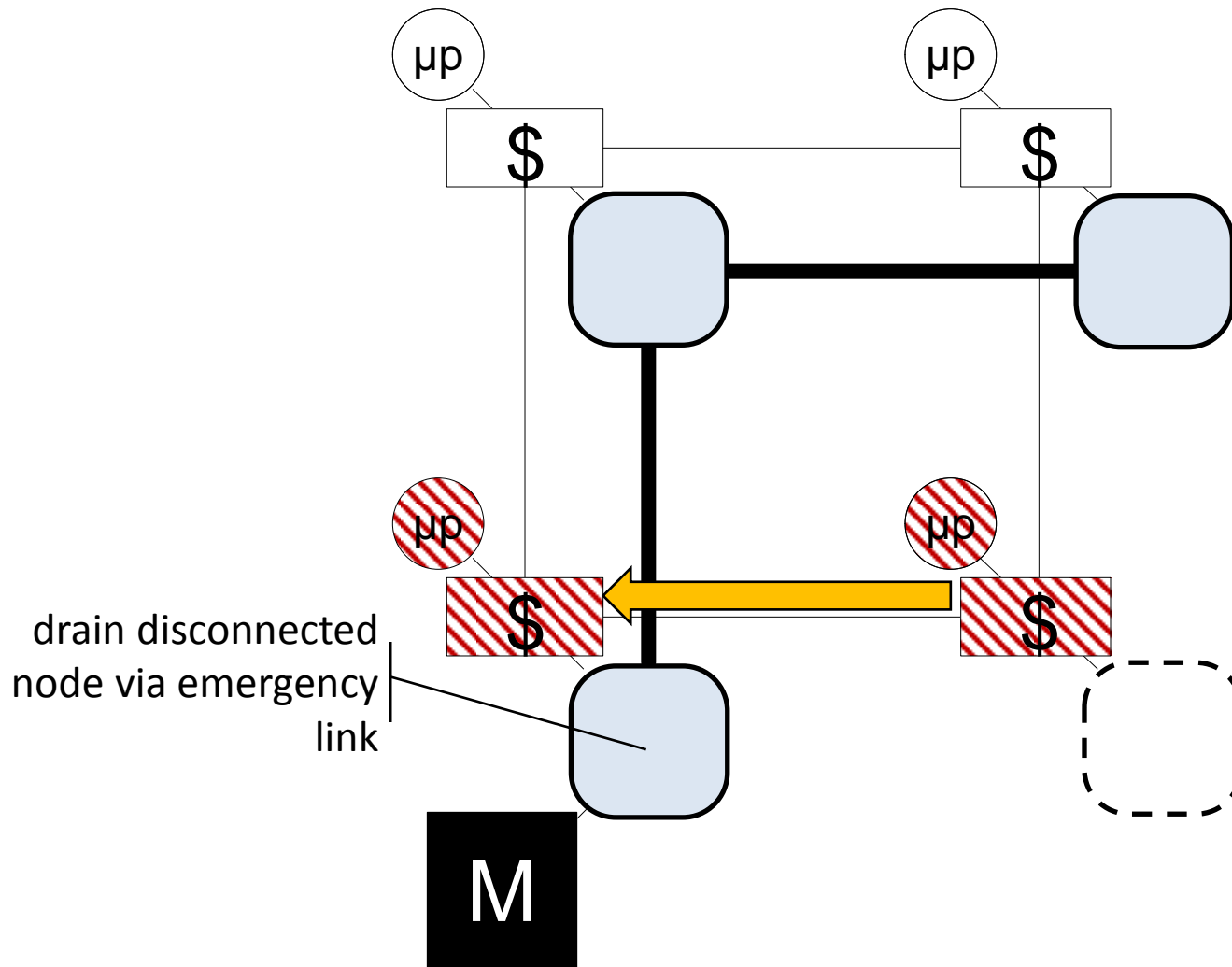
Drain Example



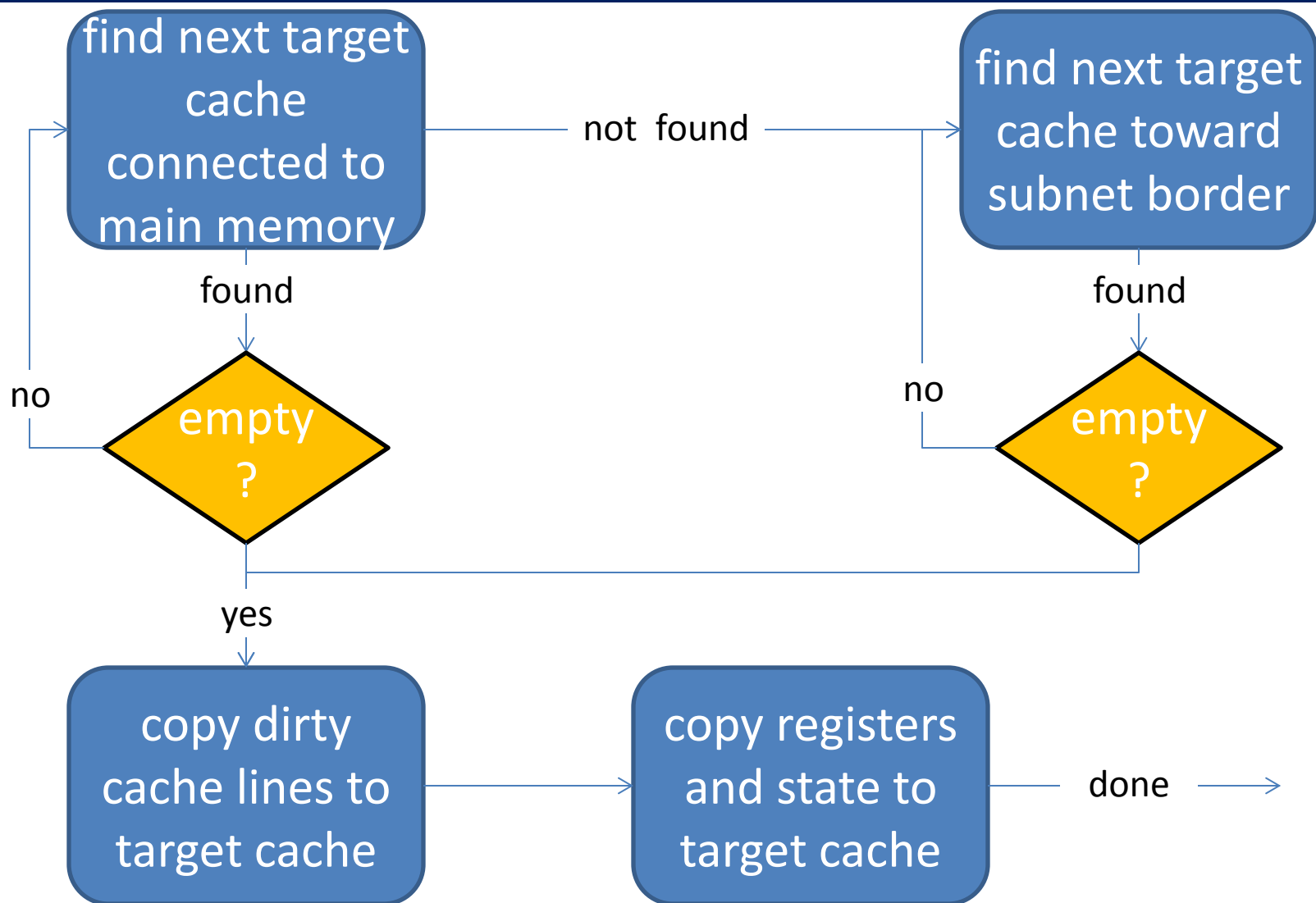
Drain Example



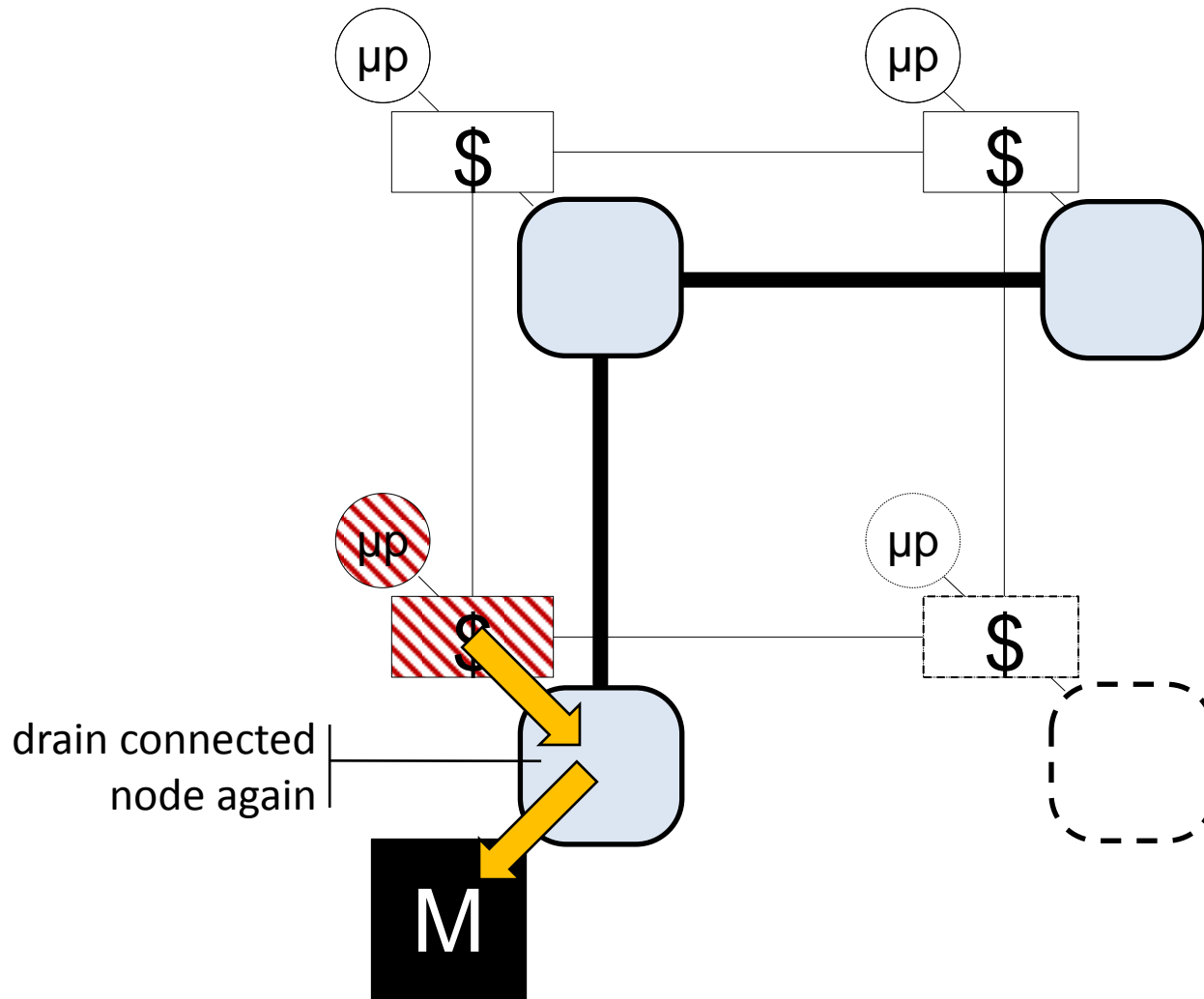
Drain Example



Emergency Link Algorithm

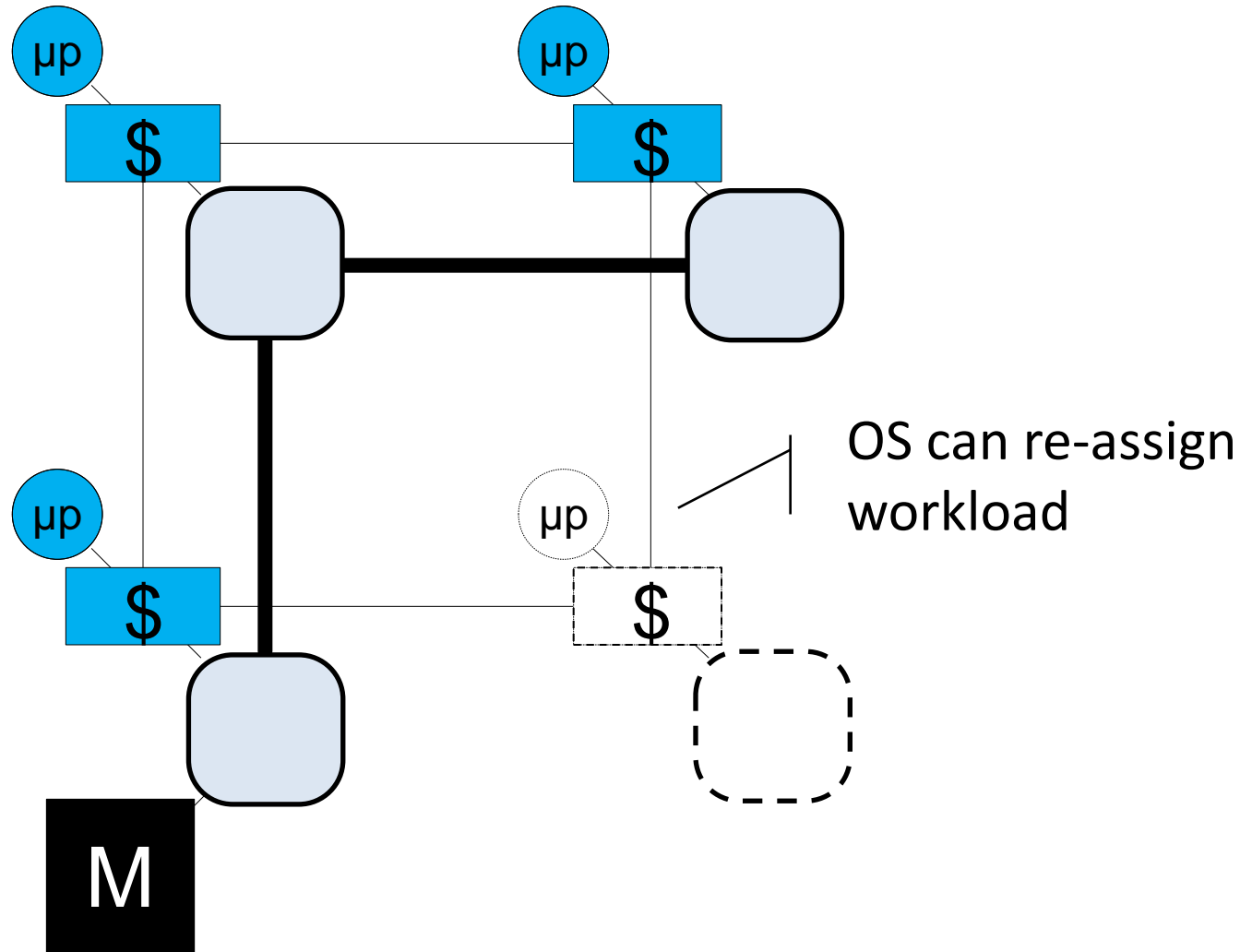


Drain Example

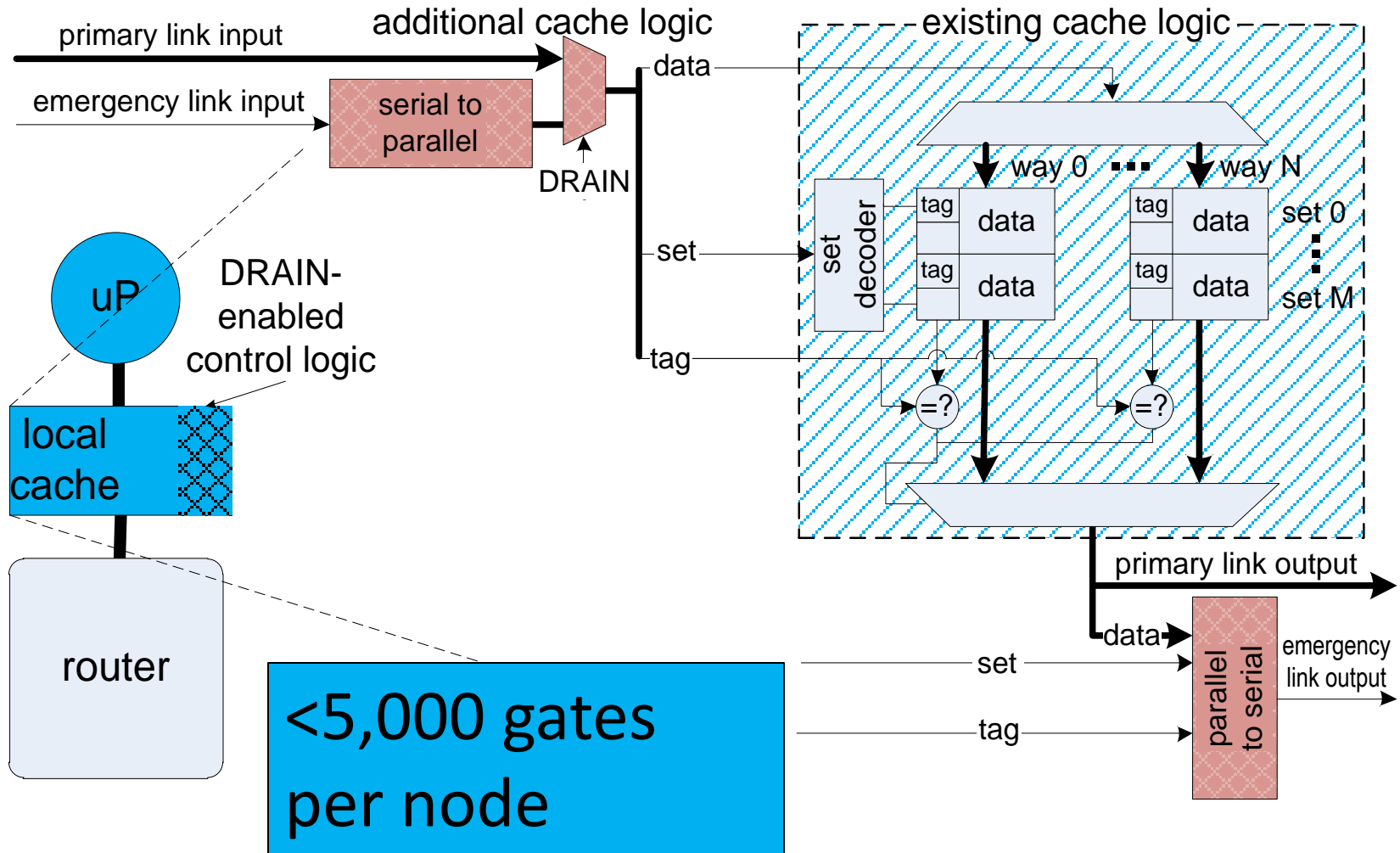


Drain Example

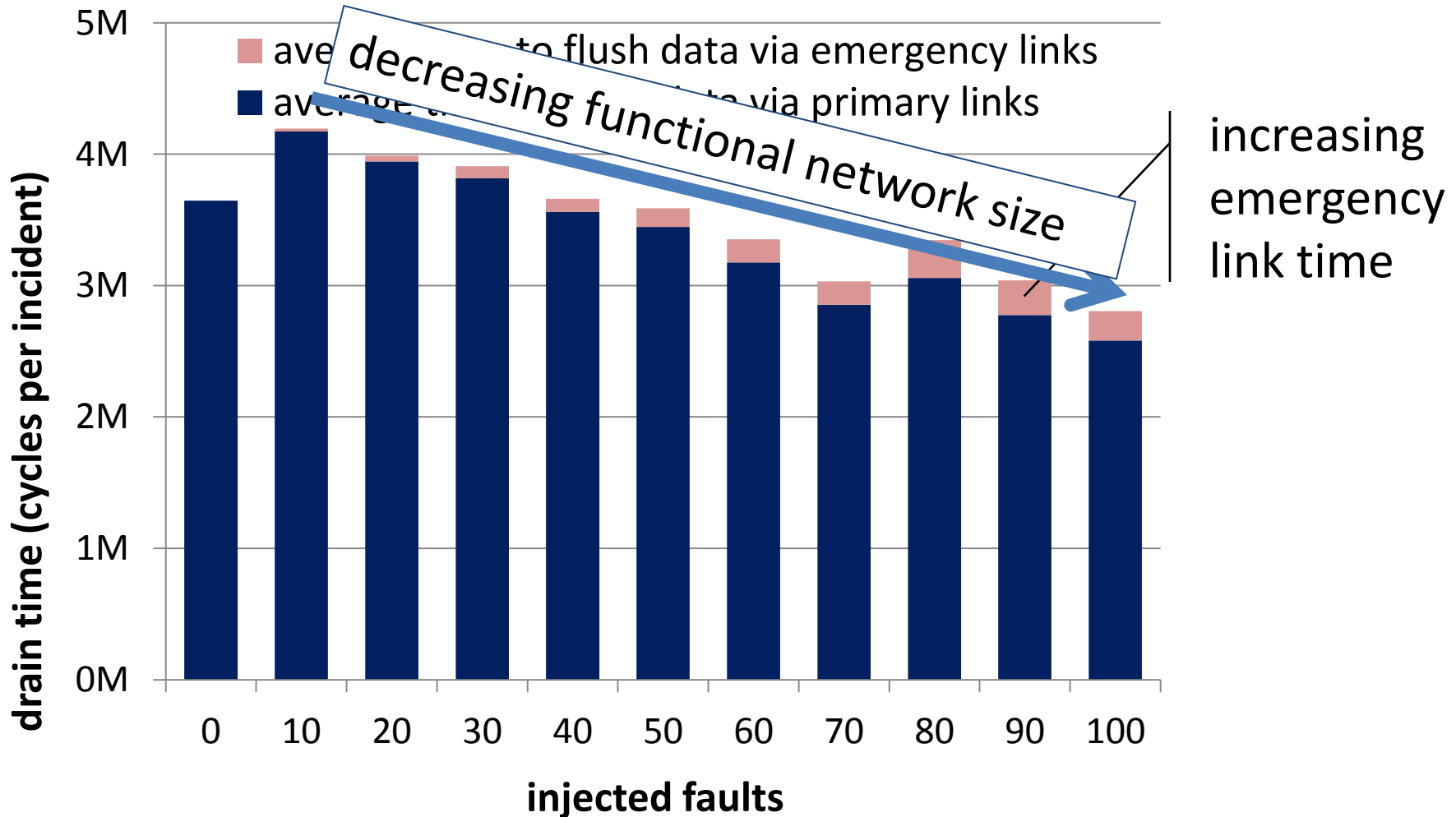
resume
normal
operation



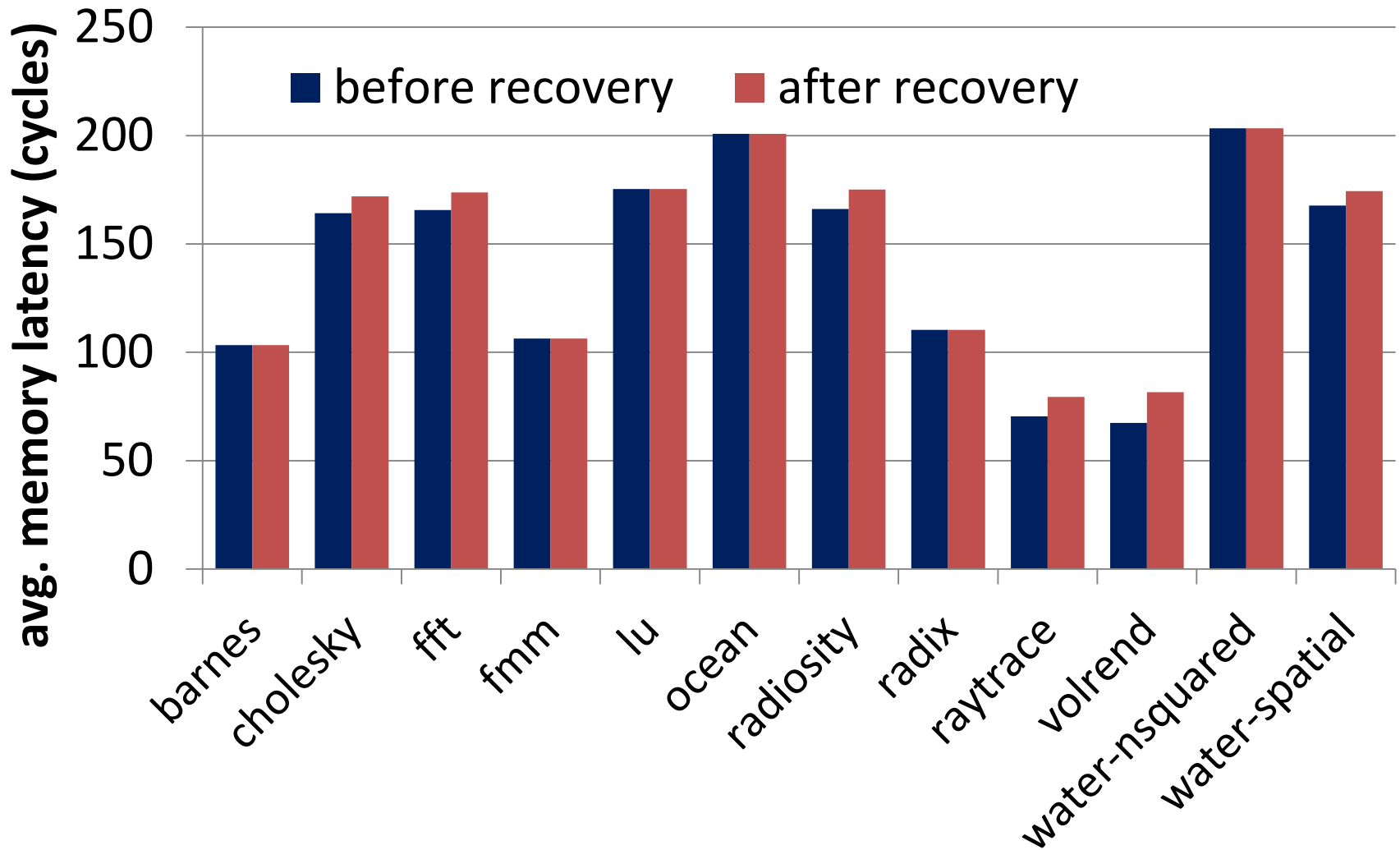
Drain Hardware



Drain Performance as Links Fail



Memory Latency Before and After



Conclusions

- DRAIN is a lightweight recovery mechanism for CMPs
 - 5,000 gates per node
- Recoup cache data and architectural state from disconnected nodes
- Performance overhead only during a recovery incident
 - ~3ms at 1GHz