Scalable CC-NUMA Design case study
SGI Origin 2000

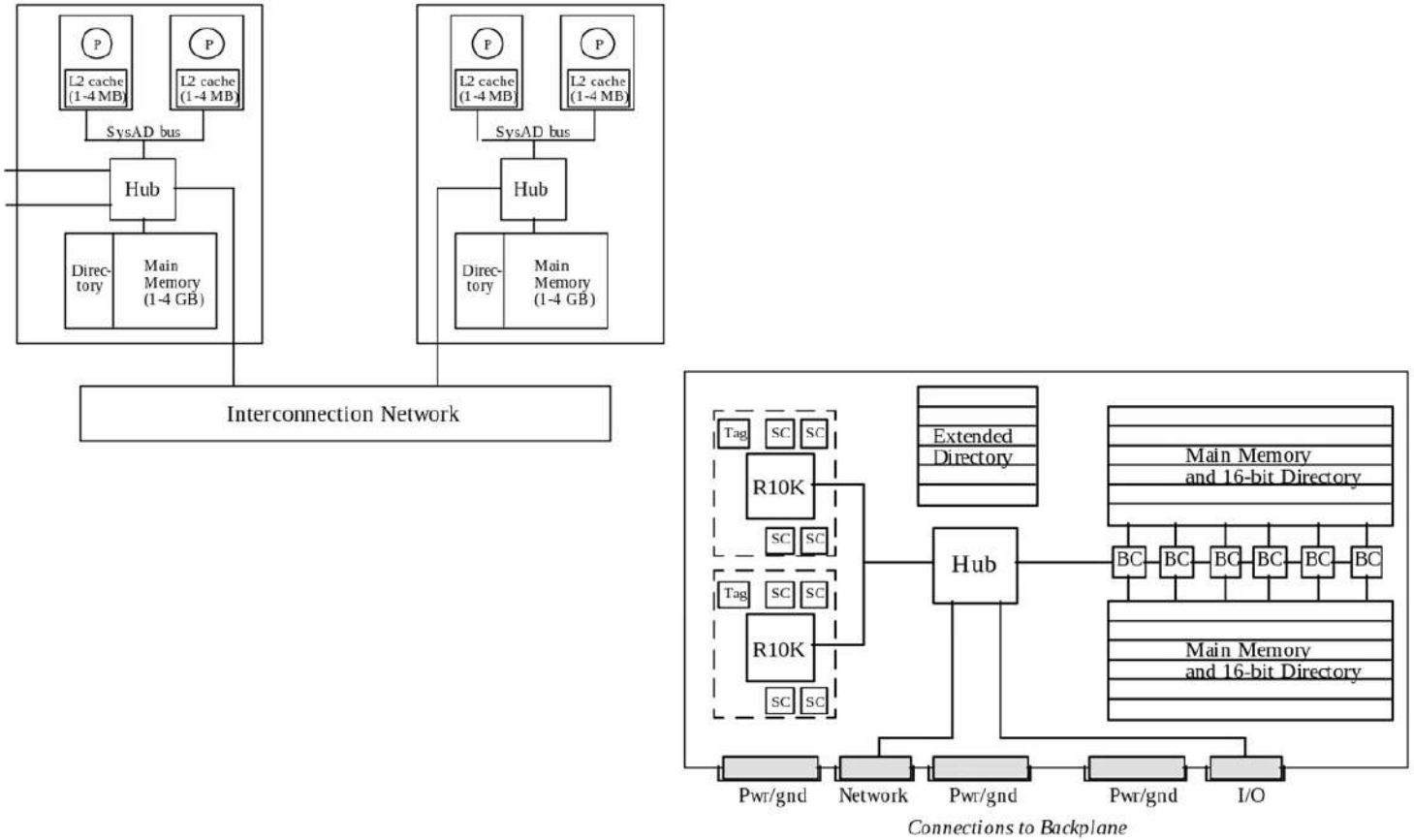
Origin 2000 system overview

- Single 16”-by-11” PCB
- Directory state in same or separate DRAMs, accessed in parallel
- Upto 512 nodes (1024 processors)
- Each node has 2 processors: MIPS R10000
- With 195MHz R10K processor, peak 390MFLOPS or 780 MIPS per proc
- Peak SysAD bus b/w is 780MB/s, so also Hub-Mem
- Hub to router chip and to Xbow is 1.56 GB/s (both are off-board)
- Hub has 4 outstanding transaction buffers
  - Connects processor, memory, network and I/O
  - Provides synch primitives
- 2 processors treated independently

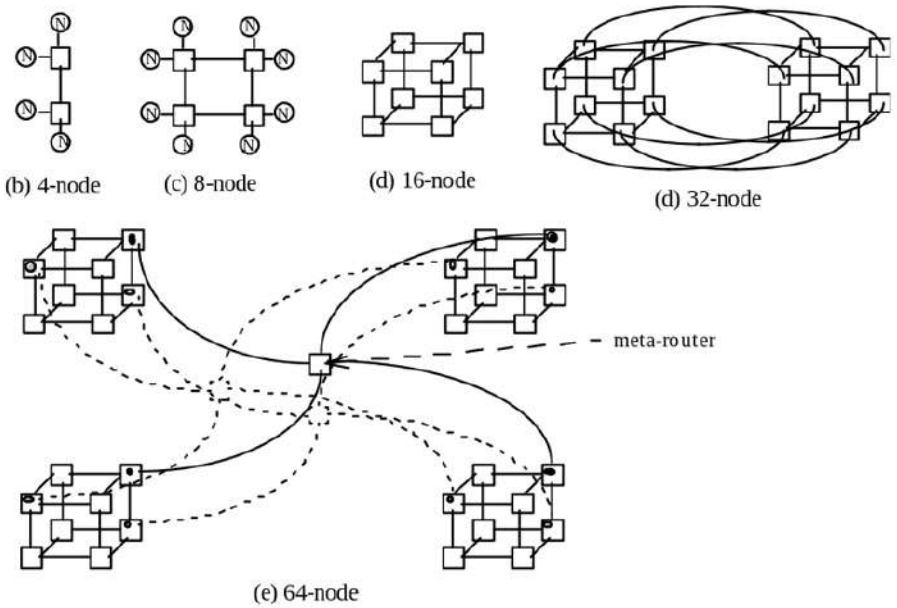
Origin 2000 system overview

- Single 16”-by-11” PCB
- Directory state in same or separate DRAMs, accessed in parallel
- Upto 512 nodes (1024 processors)
- Each node has 2 processors: MIPS R10000
- With 195MHz R10K processor, peak 390MFLOPS or 780 MIPS per proc
- Peak SysAD bus b/w is 780MB/s, so also Hub-Mem
- Hub to router chip and to Xbow is 1.56 GB/s (both are off-board)
- Hub has 4 outstanding transaction buffers
  - Connects processor, memory, network and I/O
  - Provides synch primitives
- 2 processors treated independently

Origin 2000 system overview



Origin network



- Each router has six pairs of 1.56GB/s unidirectional links
- Two to nodes, four to other routers
- latency: 41ns pin to pin across a router
- Flexible cables up to 3 ft long
- Four “virtual channels”: request, reply, other two for priority or I/O

Origin Directory Structure

- Flat, Memory-based: All directory information is at the home
- Complex as needs to scale to more than 64-nodes with 64-bit entry
- Three possible formats or interpretations
  - Exclusive
    - If a block is in exclusive state in a processor-cache then the rest of the directory-entry is not a bit-vector but an explicit pointer to the processor (not to the node)
  - Shared: bit-vector
    - If the block state is shared the directory entry is a bit-vector
    - Bits correspond to nodes (not processors)
    - The 2-proc in node are not snoop-coherent but unit of visibility to the directory is the node
    - Invalidation to a node are broadcast by Hub to both processors
    - Bit vector sizes: 16-bit format (32 procs): keep in main memory, DRAM ; 64-bit format (128 procs): extra bits in extension memory
  - Larger system: coarse vector ..

Origin directory structure

- For larger systems format used is coarse vector with each bit corresponds to P/64 nodes in a P-node system
- Group size = P/64
- Inv sent to all Hubs in the group/set and then Hub broadcasts inv to its 2 processors in node
- Ex: for max supported size = 1024 procs = 512 nodes, we need 512/64 = 8 nodes in a set
  - System dynamically chooses between coarse vector and bit-vector
- Ex: if applicat~~ion~~ (i.e. nodes sharing blocks) is confined to 64-nodes or less
  - Part of machine uses bit-vector else coarse vector

Origin cache and Directory states

- Cache states: MESI
- 7 directory states
  - Unowned: no cache has a copy, memory copy is valid
  - Shared: one/more caches have a shared copy; mem=valid
  - Exclusive: one cache (pointed to) has block in modified/exclusive state
  - Three pending/busy states
    - Busy Indicates that directory has received a previous request for the block
    - Could not satisfy it and therefore has sent to another node and directory is waiting.
    - Directory cannot take another request for the block yet as transaction is yet to complete
  - Poisoned: used for efficient page migration (not done in this course)
- Lets see how read and write are handled
  - No point-to-point order assumed in network
- 3 busy states = (1) Read ongoing (2) RdX or Upgr (3) Read on a block that will not be sent to any processor cache: uncached Read

Handling a Read Miss

- On a cache read miss the Hub examines the address: Remote? Or Local?
- Remote: send request to “home” and at home same actions as in local case
- Local: looks up directory entry and memory
  - Start (dir-read) || (memory access)
  - Speculative block read
  - Directory read completes 1 cycle before speculative read
- Directory can be in one of many states ...
  - List the states and possible actions taken

Handling Read miss

- (1) Shared or unowned
  - If shared: set presence bit
  - If unowned: set exclusive state (use ptr-format)
  - Both cases speculative memory read successful
  - Reply: send block to requestor. Strict request-response
  - If directory state is something else, speculative read is wasted
- (2) Busy
  - Home is not ready to handle the request
  - Send NACK to requestor ; requestor retries later
  - Avoids holding up buffer space for long time
- (3) Exclusive (interesting case) ....
  - If home node is not the owner then
    - Get data from owner and send to requestor + update home node
    - Uses reply forwarding for lowest latency and traffic. Not strict request-response

Read miss: state: exclusive

- Home forwards request to owner
  - Owner replies directly to requestor
  - Owner sends revision message to home
  - Set bit for new requestor
  - Change state to “shared”
- Action taken at home and owner for exclusive state

Read miss: state: exclusive

- Home forwards request to owner
  - Owner replies directly to requestor
  - Owner sends revision message to home
  - Set bit for new requestor
  - Change state to “shared”
- Action taken at home and owner for exclusive state
- At home node:
  - Memory read speculatively, but block state = ‘E’
  - Set state to busy-exclusive and NACK future requests
  - State is not yet change to
    - Shared: as mem does not have up-to-date copy
    - Exclusive: as subsequent request will chase again to owner and serialisation responsibilities will be of owner
  - Change presence vector: set requestor and unset owner (reason will be done later)
  - Home assumes block is clean-exclusive and sends a speculative reply to the requestor
  - Home forwards request to owner

4-Oct-2021 - Google

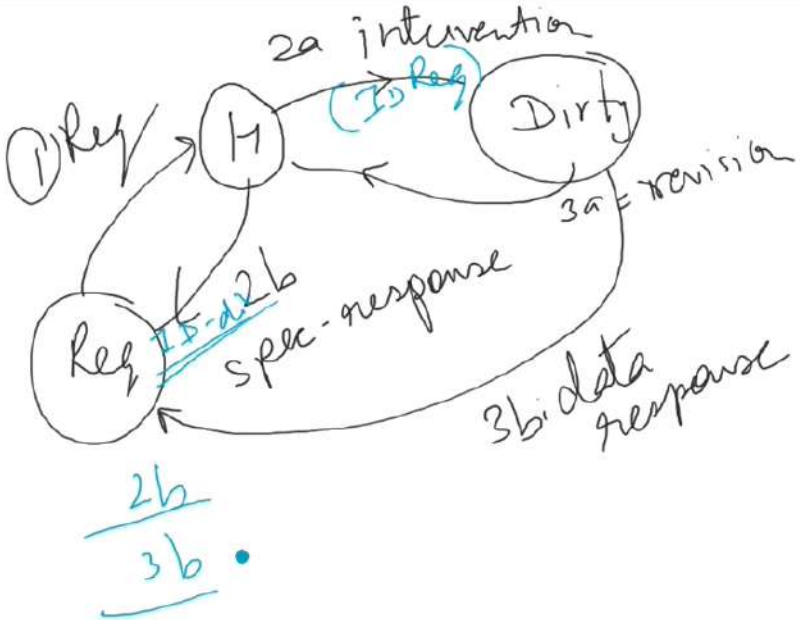
jamboard.google.com/d/1xgN9R4EQq0OZrDzyhb5wMm6ZrOqQYsfUxoR5tdgHi1U/viewer?f=7

4-Oct-2021

8/8

Share

Set background | Clear frame



VARHADE AMEY...
KOUSIK RAJESH
YOGESH KUMAR
AMIT
NALABOLU SAN...
DODDAYULA LIK...
ASWATHY N S
KUSHAL SANGW...
SWATI UPADHYAY
VEDIKA JITENDR...
VATSHAL NILESH...
MOHIT JAIN
KARTIKEYA SAXE...
Hemangee Kalpe...
IMLIJUNGLA LON...
DARSHIT NAGAR
NISHANK SIDDH...
+2

67% (1:56) Fri, October 8, 14:33

Read miss: state: exclusive

- At owner node block can be: Dirty? Or clean-exclusive?
- If dirty (='M')
  - Send data reply to requestor and revision message to home node
  - At requestor this reply over-writes the stale speculative reply from the home
  - Revision message = data sent to home = called "sharing writeback" : as (owner does) write back + keep block
- If exclusive-clean ('E')
  - Same but do not send data to requestor and home
  - ACK sent to requestor
  - Downgrade : revision to home : E->S
- Finally home changes state to shared. Busy -> 'S'

4-Oct-2021 - Google

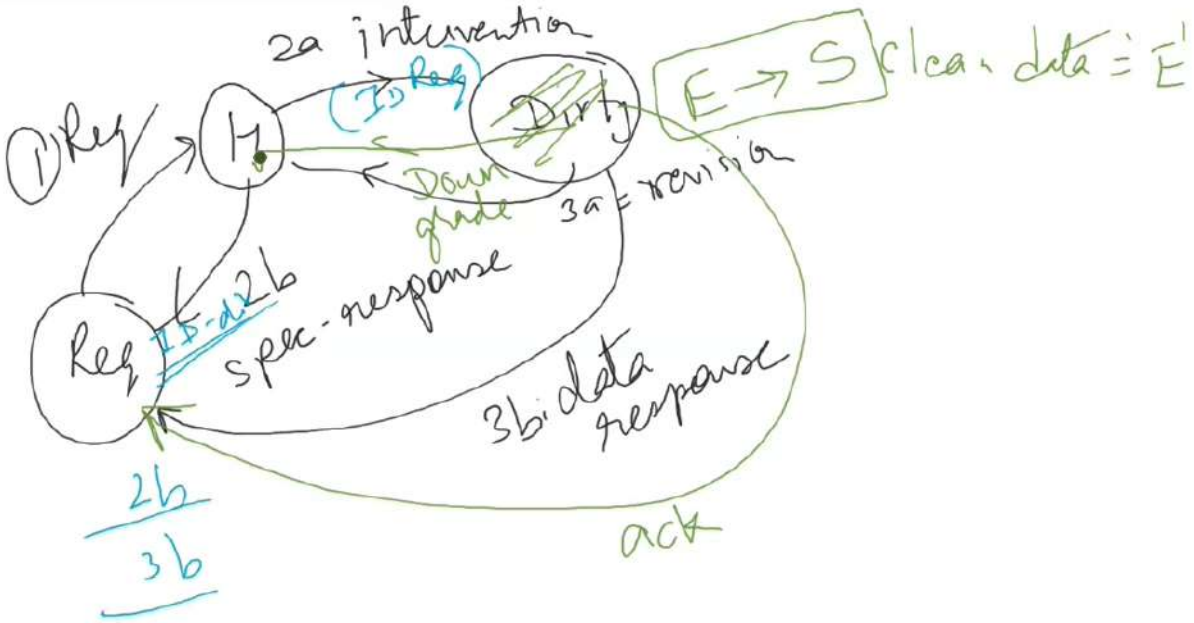
jamboard.google.com/d/1xgN9R4EQq0OZrDzyhb5wMm6ZrOqQYsfUxoR5tdgHi1U/viewer?f=7

4-Oct-2021

8 / 8

Share

Set background | Clear frame



65% (2:10) Fri, October 8, 14:36

VARHADE AMEY...
KOUSIK RAJESH
YOGESH KUMAR
AMIT
NALABOLU SAN...
DODDAYULA LIK...
ASWATHY N S
KUSHAL SANGW...
SWATI UPADHYAY
VEDIKA JITENDR...
VATSHAL NILESH...
MOHIT JAIN
KARTIKEYA SAXE...
Hemangee Kalpe...
IMLIJUNGLA LON...
DARSHIT NAGAR
NISHANK SIDDH...
+2

Speculative replies

- Requestor has to anyway wait for owner to know if copy with owner is dirty
- There are no latency savings
- We could simple always get data from owner (dirty or clean)
- Why send speculative reply?

Speculative replies

- Requestor has to anyway wait for owner to know if copy with owner is dirty
- There are no latency savings
- We could simple always get data from owner (dirty or clean)
- Why send speculative reply?
  - 2 reasons and these are based on the processor design and how protocol optimisations affect each other
  - Reason-1: R10000 L2 cache controller designed to not reply with data if copy is clean-exclusive
    - Home has to send data (just in case owner has clean-exclusive)
    - If cache has clean ex => memory has up-to-date copy so let memory send the block
    - We do not need speculative reply with intervention forwarding
  - Reason-2: ..

Speculative reply

- Reason-2: speculative replies enable write-back optimisation
  - When a cache (Pi) replaces clean-exclusive block, it simple deletes the copy and does not need to inform dir/mem
  - Memory may assume block exists somewhere (with Pi) but memory will always send block to requestor (Pj)
  - If Pi no longer has block it is OK and memory not informed
  - (Pi, after block replacement may inform Dir later in time)

Handling a Write Miss

- Write miss will result in request to home
  - Read exclusive: if block is not present
  - Upgrade: if block is valid (read-only)
- Directory state: Busy, Unowned, 'S', 'E'
- (1) Busy: home sends NACK to requestor
- (2) Un-owned:
  - If ReadEx: set bit, change state to exclusive, send data
  - If Upgrade:

4-Oct-2021 - Google

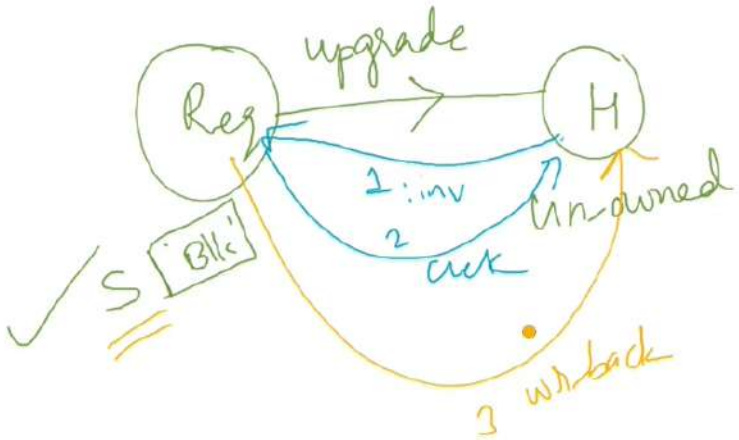
jamboard.google.com/d/1xgN9R4EQq0OZrDzyhb5wMm6ZrOqQYsfUxoR5tdgHi1U/viewer?f=8

4-Oct-2021

9/9

Share

Set background | Clear frame



62% (2:19) Fri, October 8, 14:45

VARHADE AMEY...
KOUSIK RAJESH
YOGESH KUMAR
AMIT
NALABOLU SAN...
DODDAYULA LIK...
ASWATHY N S
KUSHAL SANGW...
SWATI UPADHYAY
VEDIKA JITENDR...
VATSHAL NILESH...
MOHIT JAIN
KARTIKEYA SAXE...
Hemangee Kalpe...
IMLIJUNGLA LON...
DARSHIT NAGAR
NISHANK SIDDH...
+2

Handling a Write Miss

- Write miss will result in request to home
  - Read exclusive: if block is not present
  - Upgrade: if block is valid (read-only)
- Directory state: Busy, Unowned, 'S', 'E'
- (1) Busy: home sends NACK to requestor
- (2) Un-owned:
  - If ReadEx: set bit, change state to exclusive, send data
  - If Upgrade:
    - This is a mis-match. May be some older request reaching late ! As Origin does not assume point-to-point network order
    - As block in memory is unowned, means it was replaced from the cache and the directory was already notified
    - Upgrade is an in-appropriate request. So send NACK.
    - Requestor will later retry with ReadEx

Handling write miss

- (3) Shared or Exclusive
  - Invalidations must be sent
  - Use reply-forwarding to reduce latency
  - Home send inv
  - Home sends list of sharers to requestor
  - Sharers send inv-ack to requestor

4-Oct-2021 - Google

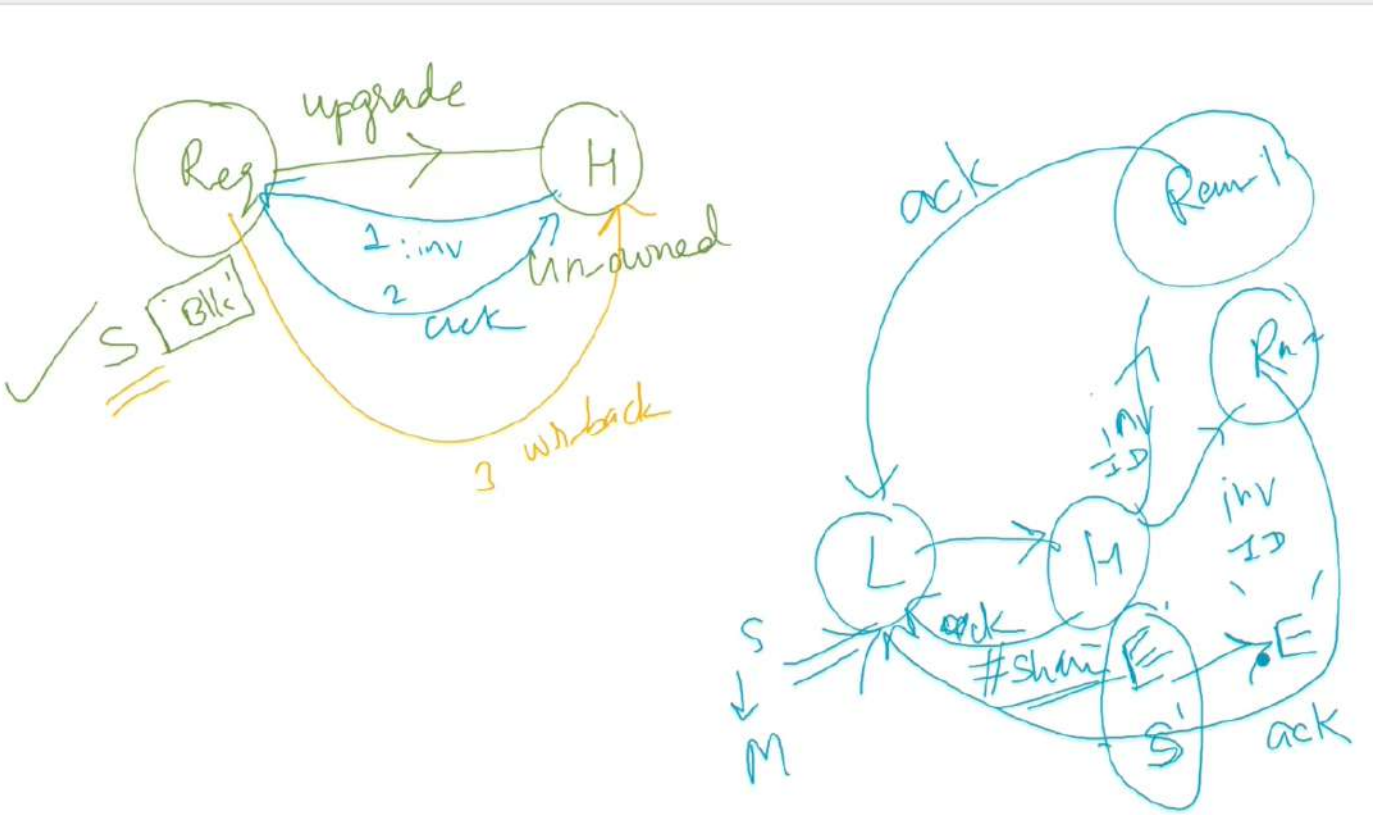
jamboard.google.com/d/1xgN9R4EQq0OZrDzyhb5wMm6ZrOqQYsfUxoR5tdgHi1U/viewer?f=8

4-Oct-2021

9/9

Share

Set background | Clear frame



60% (1:48) Fri, October 8, 14:49

VARHADE AMEY...
KOUSIK RAJESH
YOGESH KUMAR
AMIT
NALABOLU SAN...
DODDAYULA LIK...
ASWATHY N S
KUSHAL SANGW...
SWATI UPADHYAY
VEDIKA JITENDR...
VATSHAL NILESH...
MOHIT JAIN
KARTIKEYA SAXE...
Hemangee Kalpe...
IMLIJUNGLA LON...
DARSHIT NAGAR
NISHANK SIDDH...
+2

Write to block in shared state

- Requestor = Pi
- At the home:
  - (1) Set dir-state to 'E' and set presence bit of requestor (Pi). This ensures that future requests will go to Pi
  - (2) Request = RdEx
    - Send reply = "exclusive reply with invalidations pending". Contains data
    - Pi gets data and number of sharers from which to expect inv-ack. ID of sharer not required
  - (3) Request = Upgrade
    - Reply = "upgrade ack with invals pending". No data sent
  - (4) send inv to sharers, which will ack requestor

Write to block in shared state

- At the Requestor
  - Wait for all ACKs to come before closing the operation
- Another Pj sends subsequent request to Home, where:
  - Home (state= ‘E’) forwards it to Pi as intervention (and Home = Busy-Ex state)
  - For serialisation, requestor Pi does not handle the new request (Pj) until all ACKs are received for its outstanding request
- One more Req (say from Pk) will find Home = “busy-ex” state

4-Oct-2021 - Google

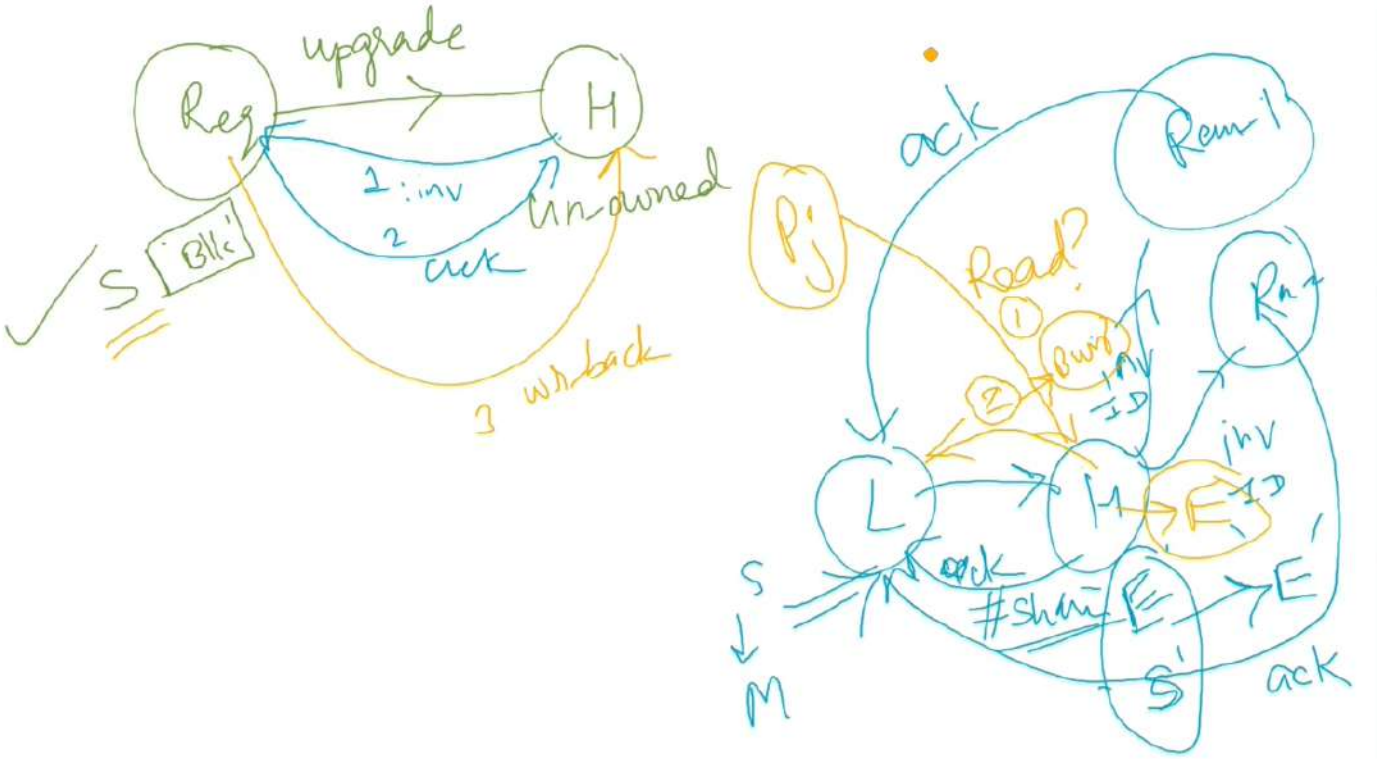
jamboard.google.com/d/1xgN9R4EQq0OZrDzyhb5wMm6ZrOqQYsfUxoR5tdgHi1U/viewer?f=8

4-Oct-2021

9 / 10

Share

Set background | Clear frame



57% (1:38) Fri, October 8, 14:55

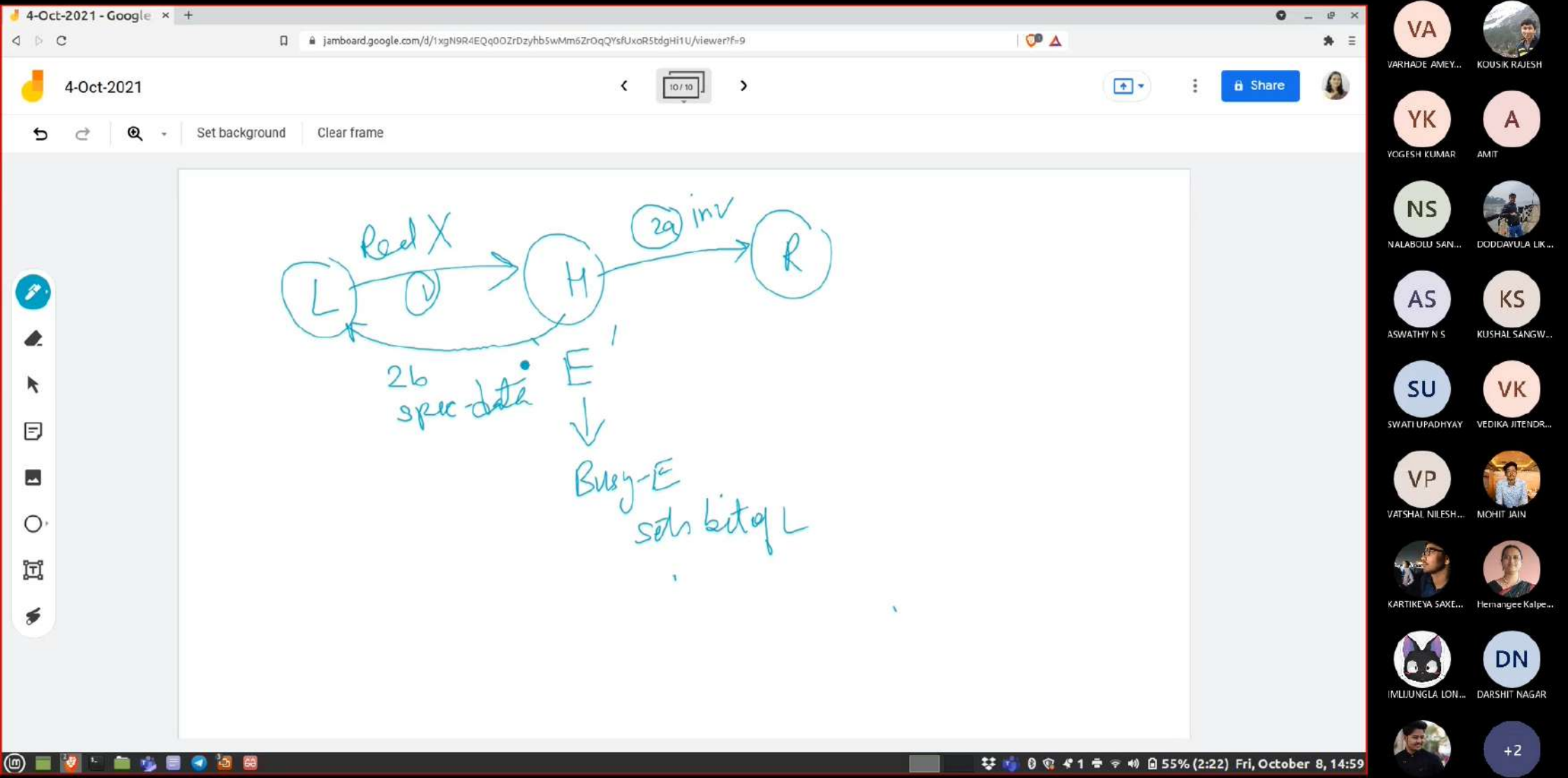
VARHADE AMEY...
KOUSIK RAJESH
YOGESH KUMAR
AMIT
NALABOLU SAN...
DODDAYULA LIK...
ASWATHY N S
KUSHAL SANGW...
SWATI UPADHYAY
VEDIKA JITENDR...
VATSHAL NILESH...
MOHIT JAIN
KARTIKEYA SAXE...
Hemangee Kalpe...
IMLIJUNGLA LON...
DARSHIT NAGAR
NISHANK SIDDH...
+2

Write to block in shared state

- At the Requestor
  - Wait for all ACKs to come before closing the operation
- Another Pj sends subsequent request to Home, where:
  - Home (state= 'E') forwards it to Pi as intervention (and Home = Busy-Ex state)
  - For serialisation, requestor Pi does not handle the new request (Pj) until all ACKs are received for its outstanding request
- One more Req (say from Pk) will find Home = "busy-ex" state

Write to block in Exclusive state

- If request = Upgrade
  - Upgrade will be sent if processor has block in ‘S’ state
  - This request has reached late as another write has beaten this one at the home. Requestors current data is not valid
  - NACK the request (... later retried as ReadEx)
- If request = ReadEx
  - Set state – Busy
  - Set presence bit
  - Send speculative reply
  - Send inv to owner with identity of requestor


4-Oct-2021 - Google
jamboard.google.com/d/1xgN9R4EQq0OZrDzyhb5wMm6ZrOqQYsfUxoR5tdgHi1U/viewer?f=9
4-Oct-2021
10 / 10
Share
Set background
Clear frame
Read X
L
M
R
2a inv
2b
spec data
E'
Busy-E
sets bit of L
VARHADE AMEY...
KOUSIK RAJESH
YOGESH KUMAR
AMIT
NALABOLU SAN...
DODDAYULA LIK...
ASWATHY N S
KUSHAL SANGW...
SWATI UPADHYAY
VEDIKA JITENDR...
VATSHAL NILESH...
MOHIT JAIN
KARTIKEYA SAXE...
Hemangee Kalpe...
IMLIJUNGLA LON...
DARSHIT NAGAR
NISHANK SIDDH...
+2
55% (2:22) Fri, October 8, 14:59

Write to block in Exclusive state

- At the Owner:
  - (1) Block = Dirty
    - Send “ownership transfer” revision message to Home (no data)
    - Send response with data to requestor (override speculative reply)
  - (2) Block = clean exclusive
    - Send “ownership transfer” revision message to Home (no data)
    - Send ACK to requestor (no data; got data from spec-reply)

4-Oct-2021 - Google

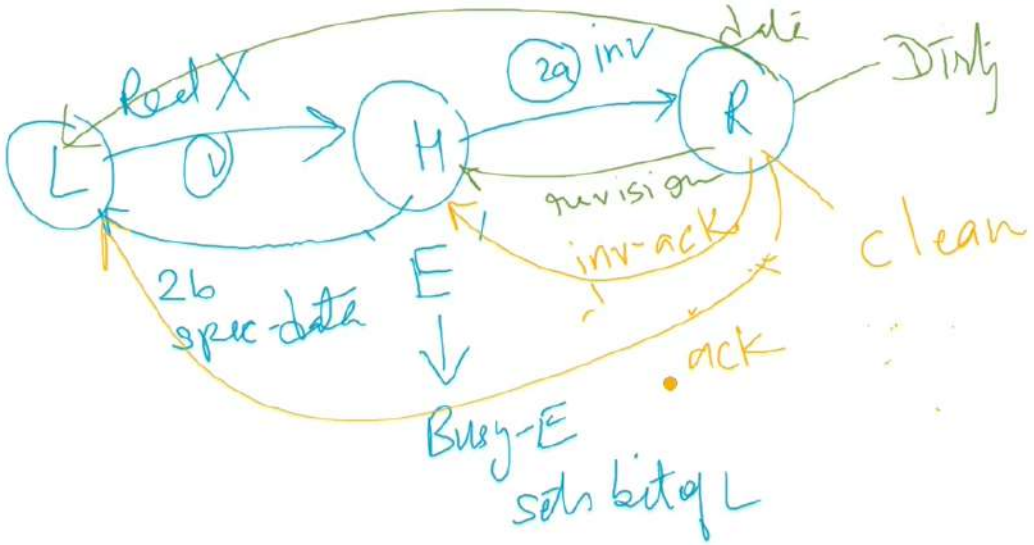
jamboard.google.com/d/1xgN9R4EQq0OZrDzyhb5wMm6ZrOqQYsfUxoR5tdgHi1U/viewer?f=9

4-Oct-2021

10 / 10

Share

Set background | Clear frame



VARHADE AMEY...
KOUSIK RAJESH
YOGESH KUMAR
AMIT
NALABOLU SAN...
DODDAYULA LIK...
ASWATHY N S
KUSHAL SANGW...
SWATI UPADHYAY
VEDIKA JITENDR...
VATSHAL NILESH...
MOHIT JAIN
KARTIKEYA SAXE...
Hemangee Kalpe...
IMLIJUNGLA LON...
DARSHIT NAGAR
NISHANK SIDDH...
+2

54% (1:41) Fri, October 8, 15:00

Handling write-back requests

- Here the requestor (node-X) is holding a dirty copy to be written back
  - Directory state cannot be shared or un-owned
    - If another request (=Y) has come which will set the state to shared, this new request will have been forwarded to node-X and the state of Home would be Busy
- State = Exclusive
  - Dir-state is set to un-owned and requestor is ACKed
- State = Busy (interesting race condition)