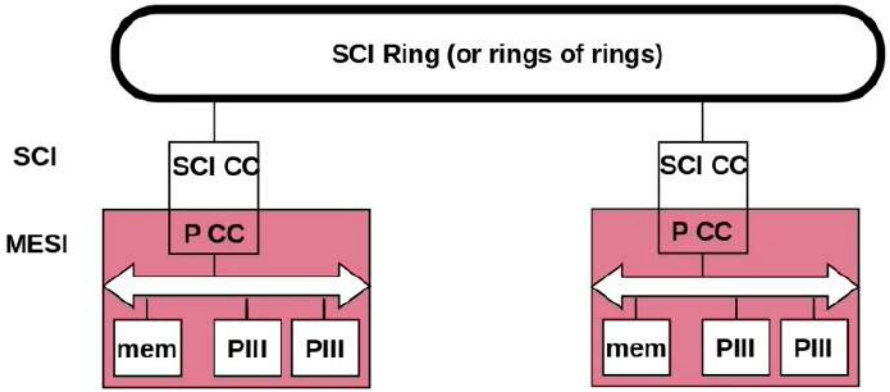
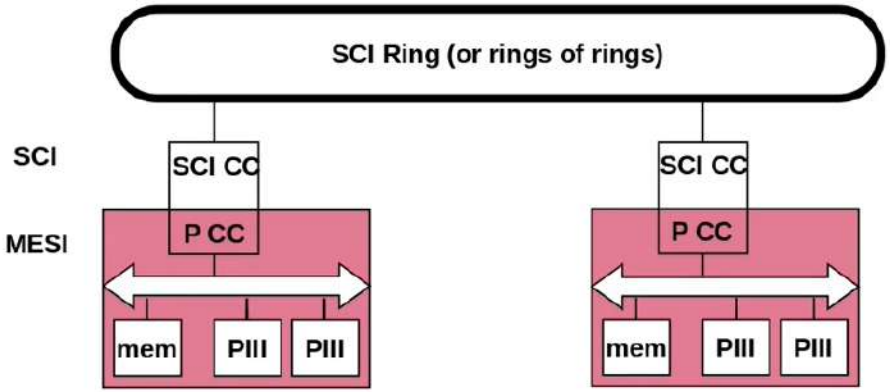
Cache-based Directory Protocols
The Sequent NUMA-Q

Composing commodity SMPs



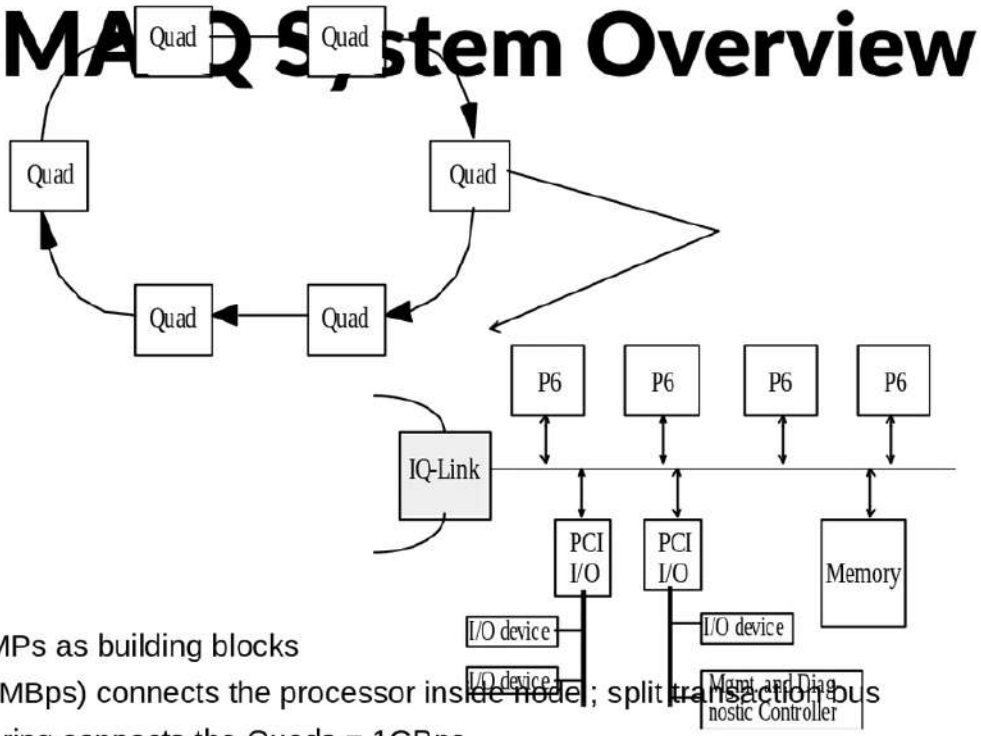
- Key concepts
  - Compose logically disparate protocols
  - Cache will provide protocol abstraction
    - That connect/expose nodes to outside components
    - i.e. a node can have its own/existing CC protocol and still be usable to compose with other heterogeneous nodes
  - This is ... .towards a “scalable ready” node
- SCI: Scalable Coherent Interface

Composing commodity SMPs



- Key concepts
  - Compose logically disparate protocols
  - Cache will provide protocol abstraction
    - That connect/expose nodes to outside components
    - i.e. a node can have its own/existing CC protocol and still be usable to compose with other heterogeneous nodes
  - This is ... .towards a “scalable ready” node
- SCI: Scalable Coherent Interface

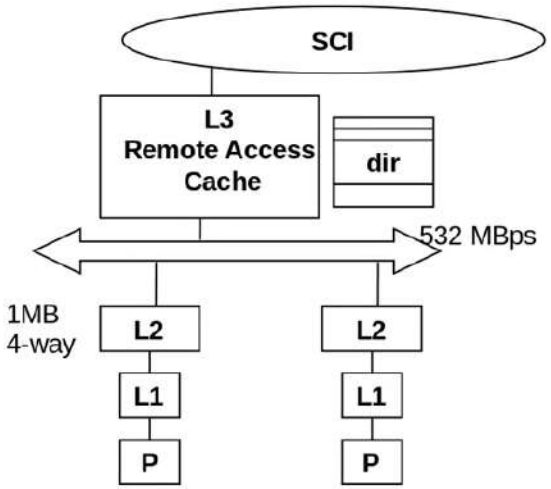
NUMA-Q System Overview



- High volume SMPs as building blocks
- Quad-bus (532 MBps) connects the processor inside node; split transaction bus
- A unidirectional ring connects the Quads = 1GBps
- Larger SCI systems built by bridging multiple rings
- Quad = 4 processors, each Intel Pentium Pro, I/O links from third party, network interface DataPump from Vitesse semiconductors, 4GB globally addressable main memory
- Only customisation is IQ-link board implementing SCI protocol

Conceptual hierarchy

- Remote access cache (RAC) represents one-node to the SCI protocol
- Associated directory points/refers to other Remote-access caches – locally and remotely allocated blocks
- RAC caches blocks fetched from remote homes
- Processor caches kept coherent with RAC using snooping protocol
- Inclusion preserved between RAC and processor caches
- The SCI directory-protocol is oblivious of how many processors are in the node



14-Oct-2021 - Google Jamboard

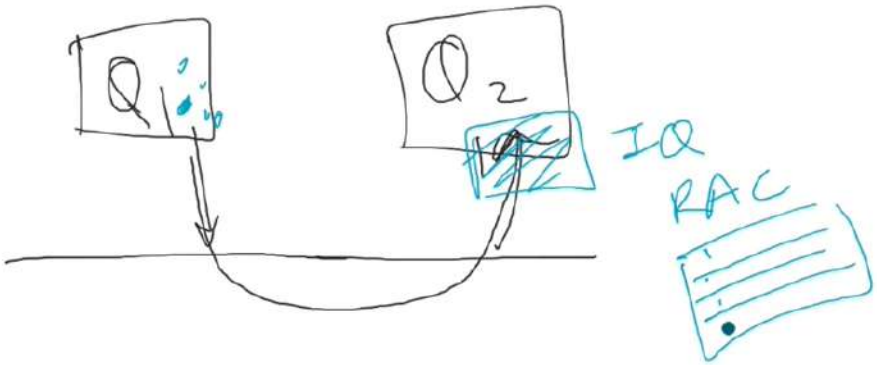
jamboard.google.com/d/1UiwQG_BvZIIGQyFfOoSRqaRIsWEb6KuLtlXtVHfRuS0/viewer

14-Oct-2021

1/1

Share

Set background | Clear frame



RATHOD SAINATH
ASWATHY N S
KUSHAL SANGW...
SWATI UPADHYAY
VEDIKA JITENDR...
IMUJUNGLA LON...
ADITYA KUMAR S...
DODDAVULA LIK...
ANNAPURNE KRI...
SARASWATULA P...
DEVANSHI GUPTA
Syam Sankar
DARSHIT NAGAR
KOUSIK RAJESH
VATSHAL NILESH...
ANSHUL MITTAL
ABHAY PRATAP G...
+2

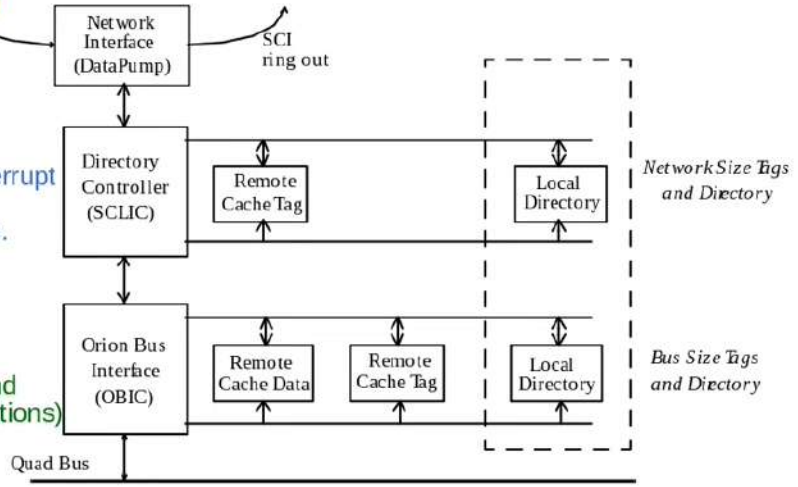
49% (1:57) Thu, October 14, 14:15

NUMA-Q IQ-link board

DataPump: provides link and packet-level transport protocol of the SCI standard. Pulls off packets destined to this quad and letting other packets go by

SCLIC: Interface to data pump, OBIC, interrupt Controller and directory tags. Manages SCI protocol using programmable engines.

OBIC: Interface to quad bus. Manages remote cache data, bus snooping and requesting logic. Pseudo-memory controller(for non-local data) and pseudo-processor (for incoming network transactions)



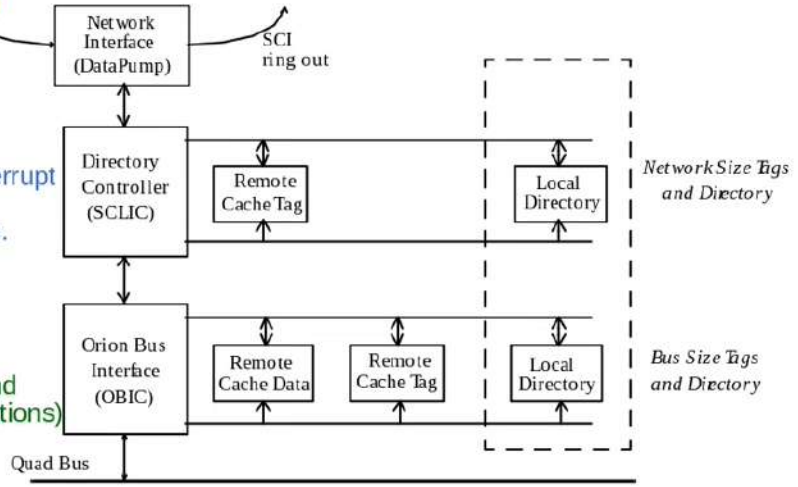
- Plays the role of Hub chip in SGI Origin
- Can generate interrupts between quads
- Remote cache (visible to SCI) has 64B block and 32 MB, 4-way capacity
- Data Pump (GaAs) implements SCI transport, pulls off relevant packets

NUMA-Q IQ-link board

DataPump: provides link and packet-level transport protocol of the SCI standard. Pulls off packets destined to this quad and letting other packets go by

SCLIC: Interface to data pump, OBIC, interrupt Controller and directory tags. Manages SCI protocol using programmable engines.

OBIC: Interface to quad bus. Manages remote cache data, bus snooping and requesting logic. Pseudo-memory controller(for non-local data) and pseudo-processor (for incoming network transactions)



- Plays the role of Hub chip in SGI Origin
- Can generate interrupts between quads
- Remote cache (visible to SCI) has 64B block and 32 MB, 4-way capacity
- Data Pump (GaAs) implements SCI transport, pulls off relevant packets

14-Oct-2021 - Google Jamboard

jamboard.google.com/d/1UiwQG_BvZIIGQyFfOoSRqaRIsWEb6KuLtlXtVHfRuS0/viewer

14-Oct-2021

1/1

Share

Set background | Clear frame

RATHOD SAINATH
ASWATHY N S
KUSHAL SANGW...
SWATI UPADHYAY
VEDIKA JITENDR...
IMUJUNGLA LON...
ADITYA KUMAR S...
DODDAVULA LIK...
ANNAPURNE KRI...
SARASWATULA P...
DEVANSHI GUPTA
Syam Sankar
DARSHIT NAGAR
KOUSIK RAJESH
VATSHAL NILESH...
ANSHUL MITTAL
ABHAY PRATAP G...
+2

46% (1:47) Thu, October 14, 14:21

How are Quads connected?

- Connect up to 8 quads on a ring
- 18-bit wide SCI ring driven by Data Pump at 1 GBps
- Strict request-response transport protocol
- Keep copy of packet in out-going buffer until ACK (echo) is returned
- When it takes a packet off the ring, replace with “positive echo”
- If packet is relevant, but cannot take it in, send “negative echo” or NACK
- When sender data pump gets NACK, it will retry automatically

14-Oct-2021 - Google Jamboard

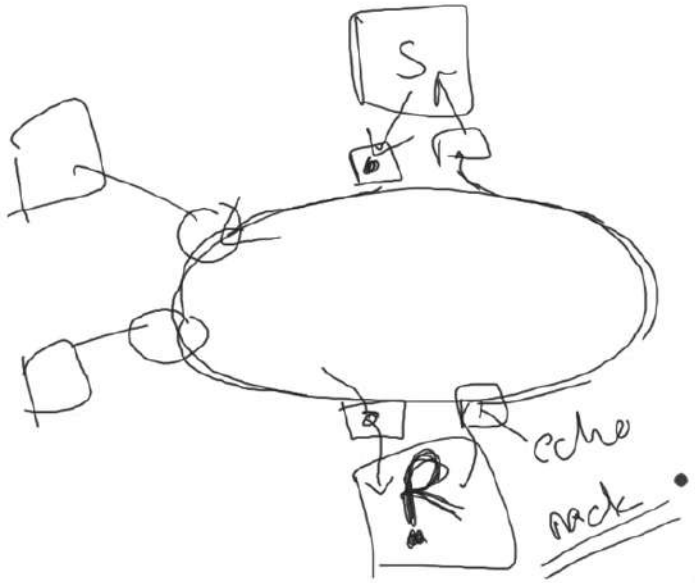
jamboard.google.com/d/1UiwQG_BvZIIGQyFfOoSRqaRIsWEb6KuLtlXtVHfRuS0/viewer?f=1

14-Oct-2021

2 / 2

Share

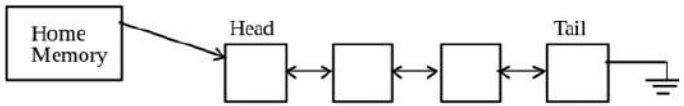
Set background | Clear frame



RATHOD SAINATH
ASWATHY N S
KUSHAL SANGW...
SWATI UPADHYAY
VEDIKA JITENDR...
IMUJUNGLA LON...
ADITYA KUMAR S...
DODDAVULA LIK...
ANNAPURNE KRI...
SARASWATULA P...
DEVANSHI GUPTA
Syam Sankar
DARSHIT NAGAR
KOUSIK RAJESH
VATSHAL NILESH...
ANSHUL MITTAL
ABHAY PRATAP G...
+2

44% (1:09) Thu, October 14, 14:24

SCI Directory structure



- List of sharers is in the form of distributed linked list
- Each entry corresponds to Remote cache in Quad
- Forward pointer = towards tail = downstream pointer
- Pointer towards Head = backward pointer = upstream pointer
- Home has state+pointer to Head-node
  - Head has read + write permission
  - Others have read-only access
- Head, Tail, Middle nodes

Directory states

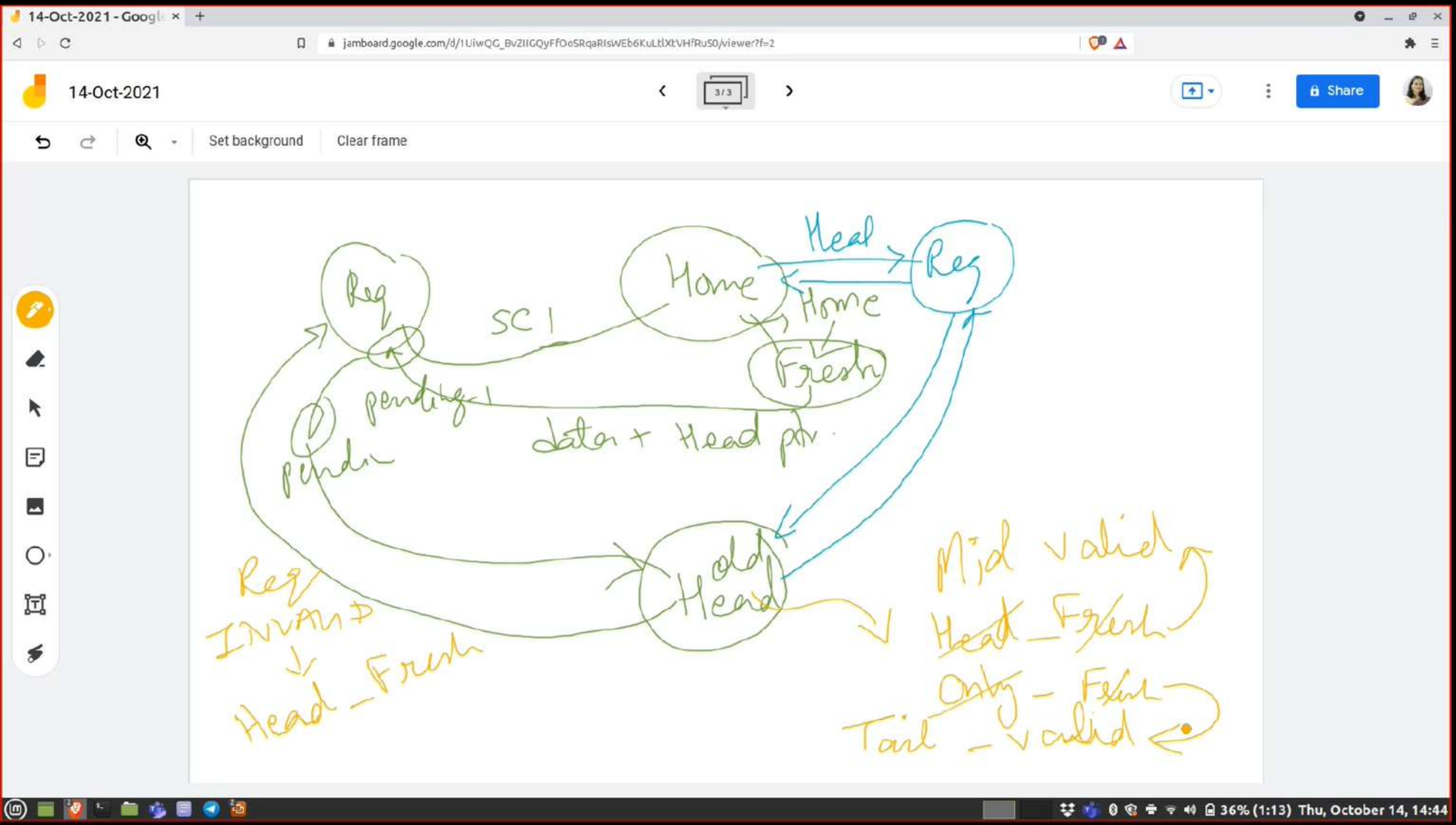
- Directory states
  - (1) **Home**: no remote cache has a copy of this block. NOTE: a local processor inside this Quad may cache it; but SCI is oblivious to it. Blocks in home memory are not kept in RAC. These blocks can be cached in the processor caches and are kept coherent with the memory by the bus protocol
  - (2) **Fresh**: One or more read-only copies in sharing list. Memory is valid
  - (3) **Gone**: Another remote cache has writable copy (exclusive or dirty). Home copy not -valid
- Processor cache states
  - Are MESI, governed by the internal protocol
  - These are not directly related to state in RAC
- RAC block states
  - There are 29 stable-states and several pending states
  - Stable states are identified by 2-parts
    - 1st part = Position of entry in the sharing list = Only, Head, Mid, Tail
    - 2nd part = state of block = Dirty, Clean (but can write, like state='E'), Fresh (Read-only), Copy, Pending, ...
  - Complete list in IEEE standard for SCI documents – in the year 1993
  - We will see few of these in our scenarios

Operations on the list

- (1) List construction: Add new node at Head
- (2) Roll-out: remove a node, needs to communicate with neighbours
- (3) Purge: (invalidation) Node at Head can send inv to all others and it becomes the only element
  - Only Head can issue a purge

Handling a Read Miss

- At the Requestor
  - Allocate block entry in the Requestor RAC
  - Set state of this block entry = "Pending". This protocol makes blocks even at requestors in busy state... but does not NACK requests
  - Start list construction to add self to Head of the sharing list by sending request to Home
- At the Directory
  - Directory states can be: Home, Fresh, Gone
- (1) Dir-state = HOME
  - No shared copies
  - Home updates state => FRESH + send block to requestor
  - Home sets Head pointer to requestor
  - After requestor gets data, requestor changes Pending to => ONLY_FRESH
    - All actions are atomic, i.e. complete all actions then handle another request
    - Follows strict-request-response protocol


14-Oct-2021 - Google
jamboard.google.com/d/1UiwQG_BvZIIGQyFfOoSRqaRIsWEb6KuLtlXtVHfRuS0/viewer?f=2
14-Oct-2021
3 / 3
Share
Set background
Clear frame
Req
SC 1
Home
Heal
Req
Home
Fresh
pending
data + Head ptr
Req
Old Head
Req/ INVAL
Head_Fresh
Mid valid
Head_Fresh
Only - Fresh
Tail_Valid
RATHOD SAINATH
ASWATHY N S
KUSHAL SANGW...
SWATI UPADHYAY
VEDIKA JITENDR...
IMIJUNGLA LON...
ADITYA KUMAR S...
DODDAVULA LIK...
ANNAPURNE KRI...
SARASWATULA P...
DEVANSHI GUPTA
Syam Sankar
DARSHIT NAGAR
KOUSIK RAJESH
VATSHAL NILESH...
ANSHUL MITTAL
ABHAY PRATAP G...
+2
36% (1:13) Thu, October 14, 14:44

14-Oct-2021 - Google Jamboard

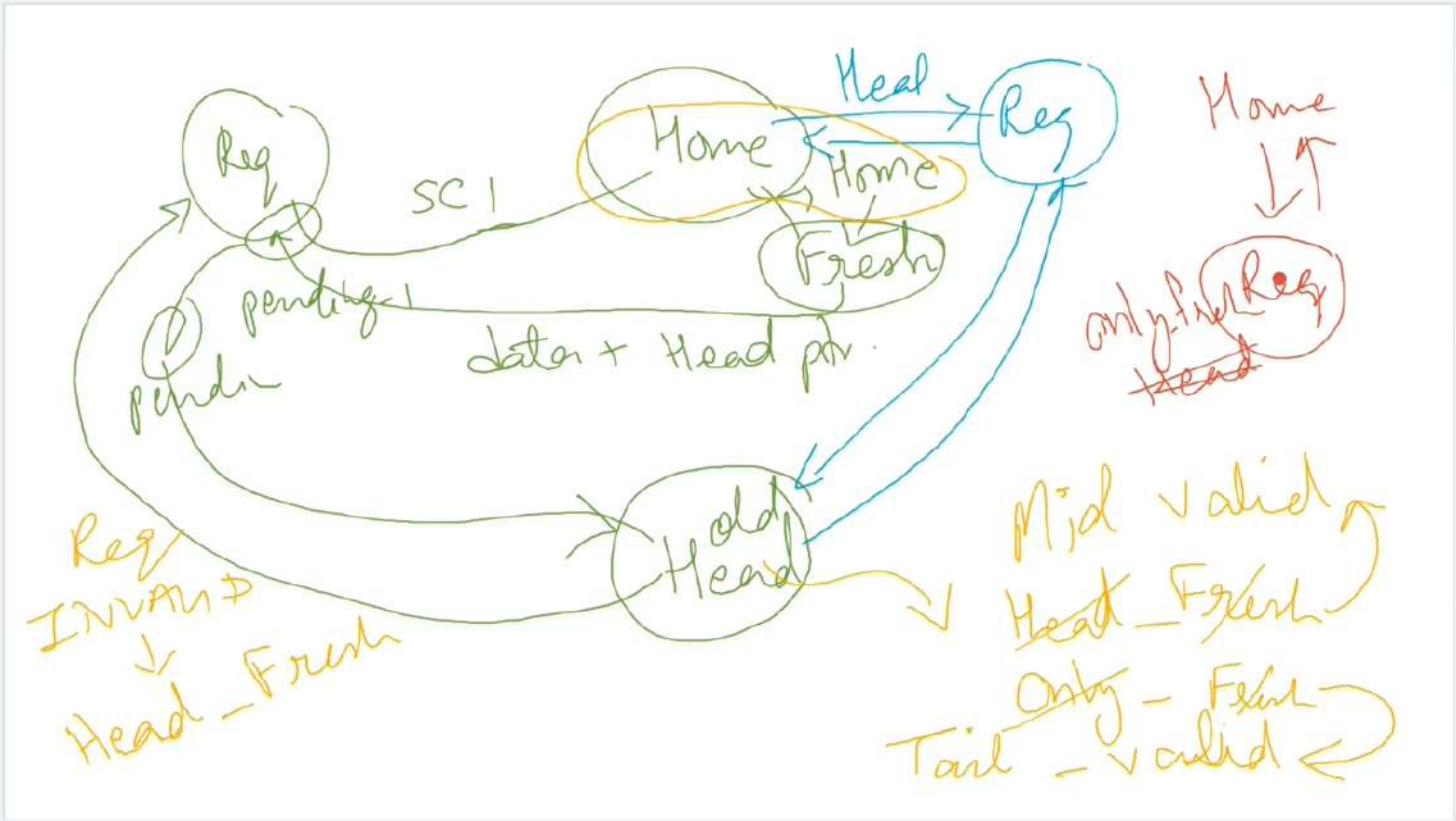
jamboard.google.com/d/1UiwQG_BvZIIGQyFfOoSRqaRIsWEb6KuLtlXtVHfRuS0/viewer?f=2

14-Oct-2021

3 / 3

Share

Set background | Clear frame



RATHOD SAINATH
ASWATHY N S
KUSHAL SANGW...
SWATI UPADHYAY
VEDIKA JITENDR...
IMIJUNGLA LON...
ADITYA KUMAR S...
DODDAVULA LIK...
ANNAPURNE KRI...
SARASWATULA P...
DEVANSHI GUPTA
Syam Sankar
DARSHIT NAGAR
KOUSIK RAJESH
VATSHAL NILESH...
ANSHUL MITTAL
ABHAY PRATAP G...
+2

36% (1:13) Thu, October 14, 14:45

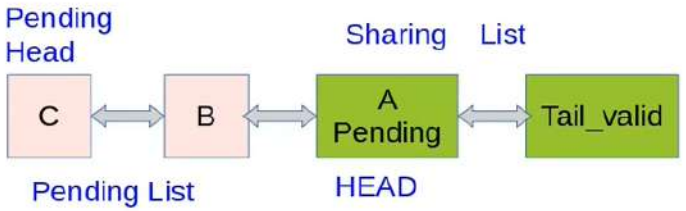
Handling a Read Miss

- (1) Dir-state = HOME
- (2) Dir-State = FRESH (i.e. sharing list exists)
  - Home copy = valid
  - Home sends data to requestor + old-Head pointer and updates its Head pointer to the new requestor
  - Requestor gets data and old Head pointer
    - Changes to new-pending state
    - Sends request to old-Head to add Requestor as new Head
  - Old-Head moves HEAD_FRESH => MID_VALID or ONLY_FRESH => TAIL_VALID
  - Old-Head updates back pointer and replies
  - Requestor moves .... pending => HEAD_FRESH
- (3) Dir-State = GONE ..

Handling a Read Miss

- (3) Dir-State = GONE
  - Home: updates Head pointer to new requestor and replies with pointer to old-head
    - Does not know/care about details of block state
  - Requestor: new-pending state, sends request to old-head for data and attach
  - Old-head: respond with data, update back pointer
  - Change state HEAD_DIRTY => MID_VALID or ONLY_DIRTY => TAIL_VALID
  - Requestor: state pending => HEAD_DIRTY
    - !!! This is a Read-Miss !!!
    - HEAD_DIRTY does not mean it can write without inv sharing list
    - It can update but must inv sharing list first
    - No need to communiate with Home .. as the state is already DIRTY

Read-miss: old-head pending?



- What if old-head was pending?
- Let old-head == A, new-requestor=B
- B goes to Home, home sends ptr-A and B goes to A, A=pending
- A is **busy** in some memory operation
  - A does not buffer request-B
  - A does not NACK B
  - Instead it adds B at the Head by extending the list backward into a pending list
- Node-B is physically attached to the head but is still waiting to be the true-head.
  True-head = A, Pending-head = B
- If another request from C goes to home: Home forwards it to B
  - B attaches 'C' to pending list: Pending-Head= C
- When A completes operation it replies to B, and True-Head = B
- When B completes: it passes True-Head to C
- Home Dir-state is never pending/busy, it always returns the Head pointer

14-Oct-2021 - Google Jamboard

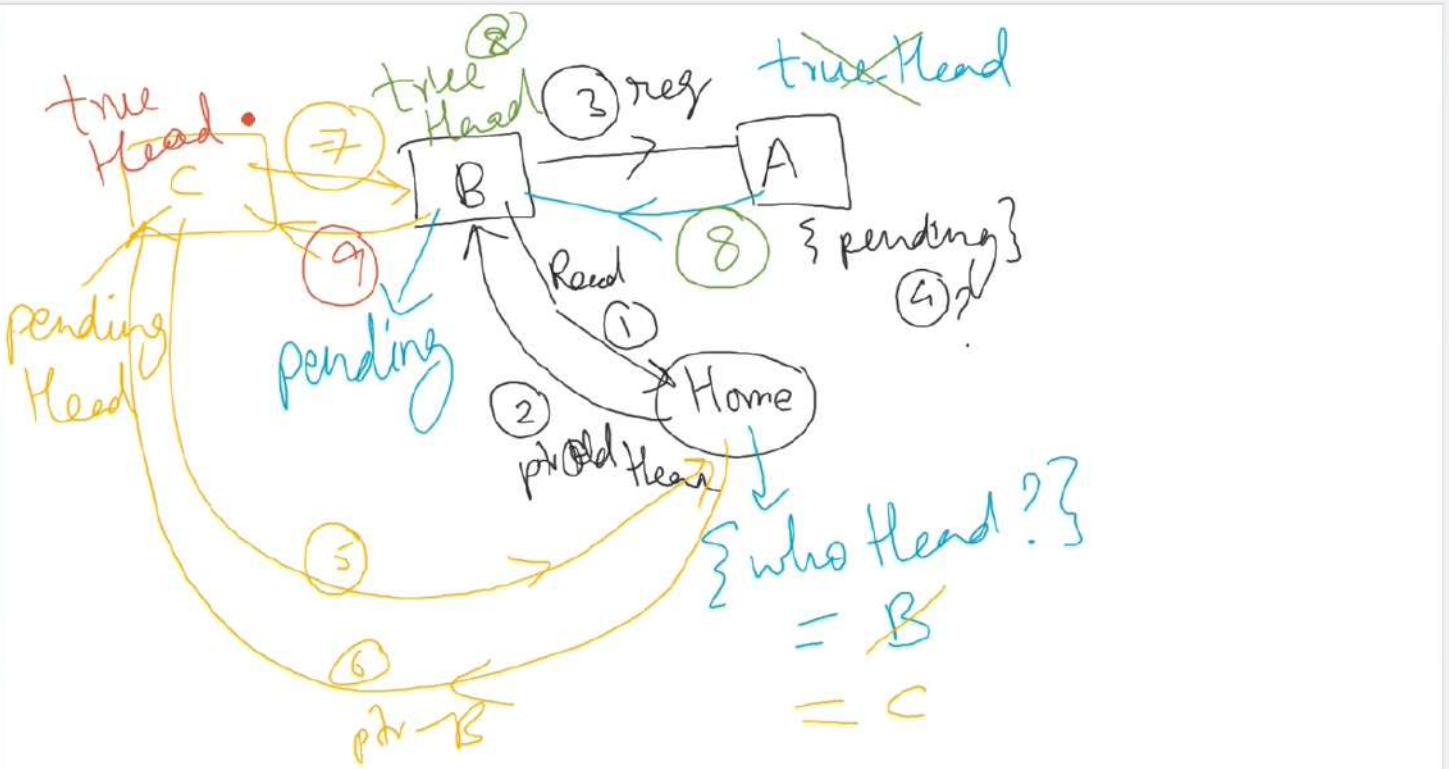
jamboard.google.com/d/1UiwQG_BvZIIGQyFfOoSRqaRIsWEb6KuLtlXtVHfRuS0/viewer?f=3

14-Oct-2021

4 / 4

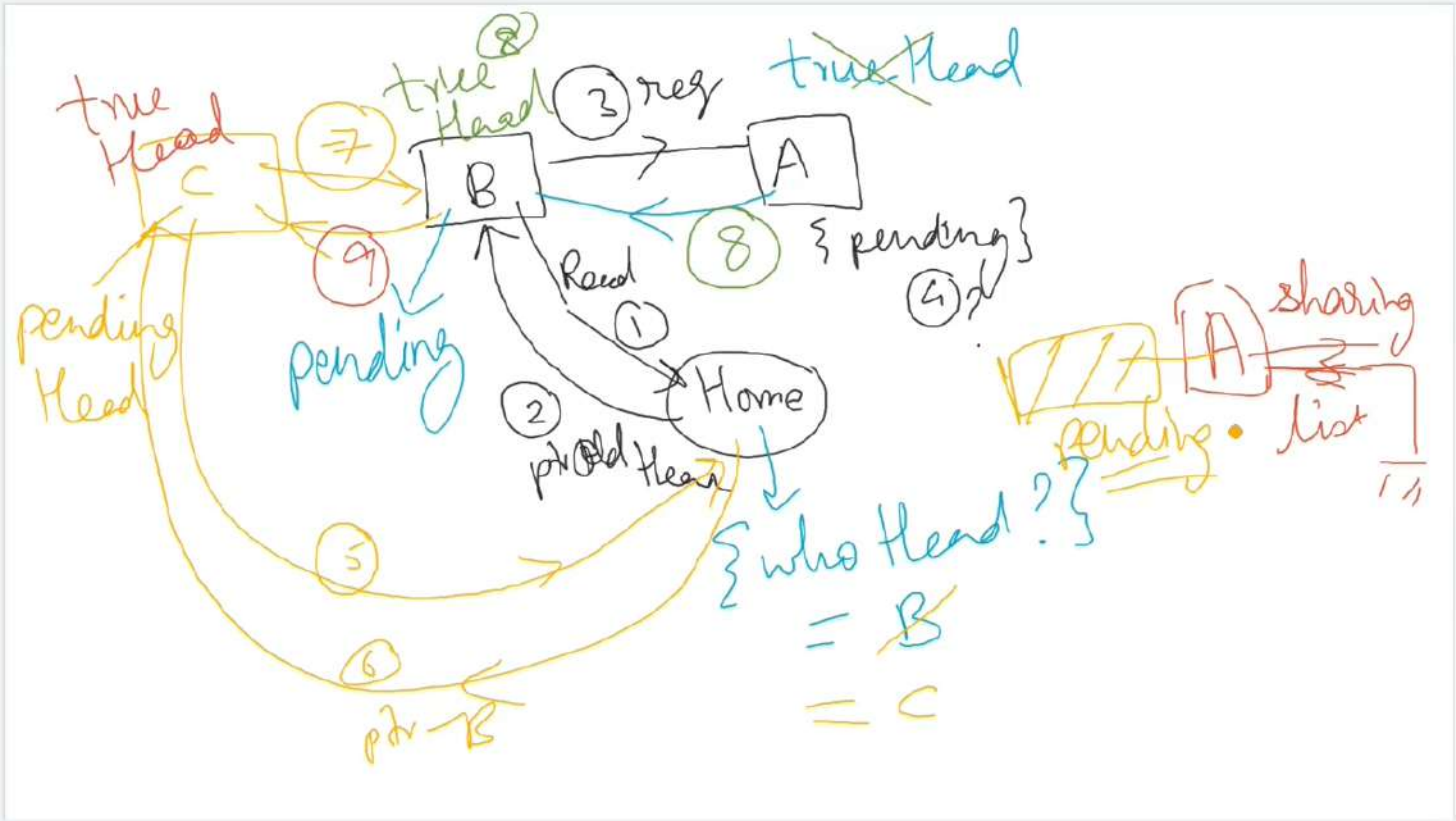
Share

Set background | Clear frame



RATHOD SAINATH
ASWATHY N S
KUSHAL SANGW...
SWATI UPADHYAY
VEDIKA JITENDR...
IMIJUNGLA LON...
ADITYA KUMAR S...
DODDAVULA LIK...
ANNAPURNE KRI...
SARASWATULA P...
DEVANSHI GUPTA
Syam Sankar
DARSHIT NAGAR
KOUSIK RAJESH
VATSHAL NILESH...
ANSHUL MITTAL
ABHAY PRATAP G...
+2

31% (0:53) Thu, October 14, 14:55


true Head
C
pending Head
pending
B
A
Home
Read
pending
sharing
pending list
who Head ?
= B
= C