

Cache-based Directory Protocols

The Sequent NUMA-Q



ANSHUL MITTAL



NISHANK SIDDHANT



RATHOD SAINATH



ASWATHY N S



KUSHAL SANGWAN



GUPTA B



SWATI UPADHYAY



VEDIKA JITENDRAN



IMLIJUNGLA LON...

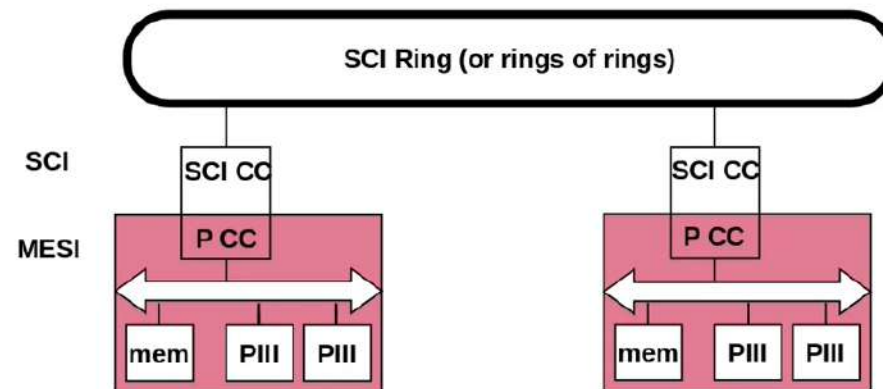


ADITYA KUMAR S...



Hemangee Kalpe...

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

AM

ANSHUL MITTAL

RATHOD SAINATH

AS

ASWATHY N S

KS

KUSHAL SANGW...

G

gupta18e

SU

SWATI UPADHYAY

VK

VEDIKA JITENDR...

IMIJUNGLA LON...

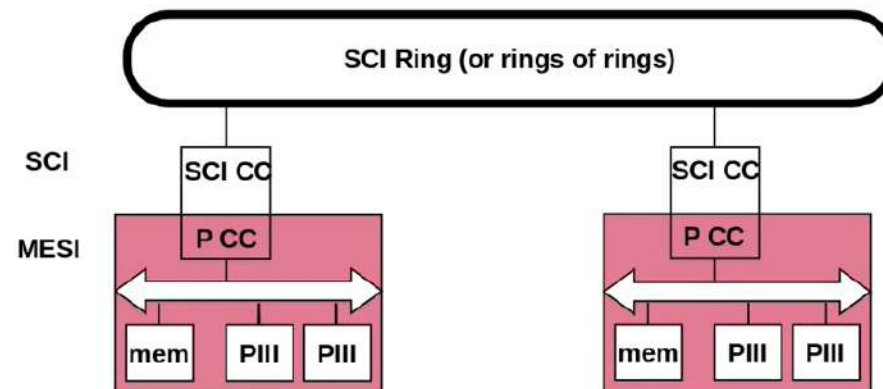
AS

ADITYA KUMAR S...

DODDAVULA LIK...

Hemangee Kalpe...

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



ANSHUL MITTAL



RATHOD SAINATH



ASWATHY N S



KUSHAL SANGW...



GUPTA ISE



SWATI UPADHYAY



VEDIKA JITENDR...



IMIJUNGLA LON...



ADITYA KUMAR S...



DODDAVULA LIK...

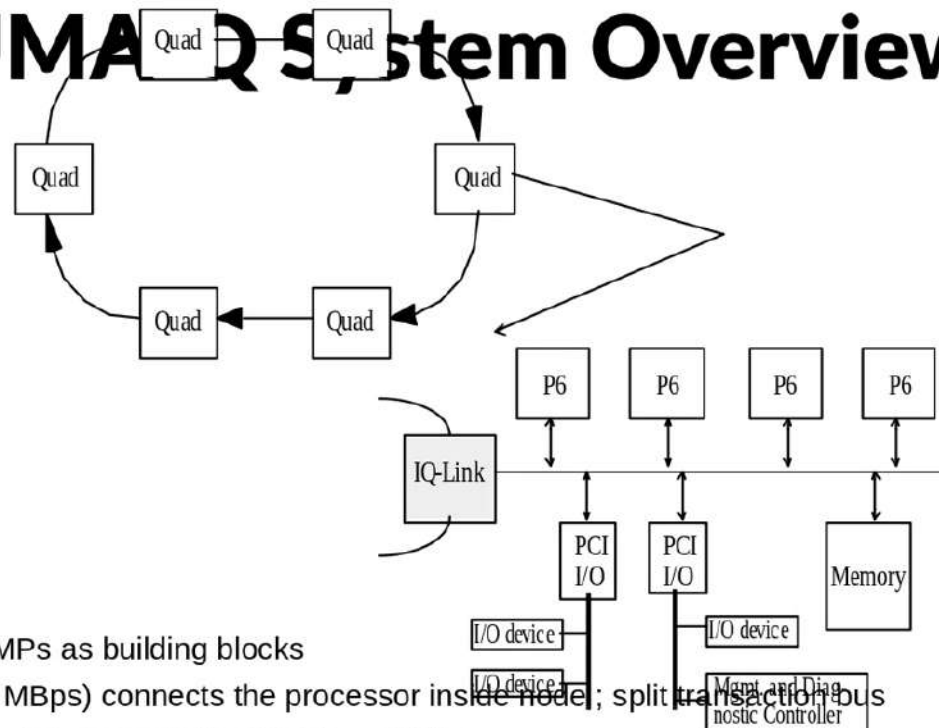


YOGESH KUMAR



Hemangee Kalpe...

NUMA Q S, 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



ANSHUL MITTAL



RATHOD SAINATH



ASWATHY N S



KUSHAL SANGW...



SWATI UPADHYAY



VEDIKA JITENDR...



IMLIJUNGLA LON...



ADITYA KUMAR S...



DODDAVULA LIK...



YOGESH KUMAR



ANNAPURNE KRI...



SARASWATULA P...



DEVANSHI GUPTA



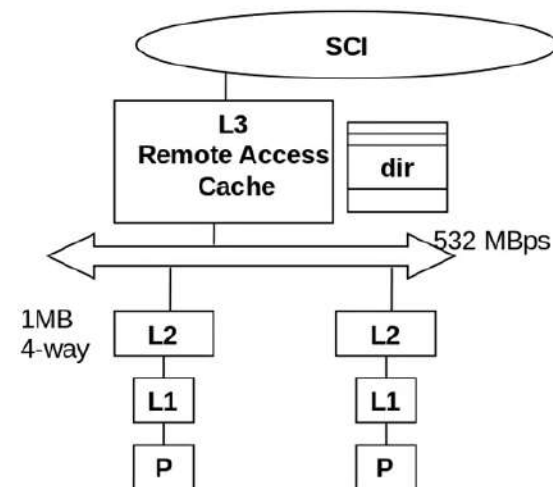
Syam Sankar



Hemangee Kalpe...

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



RATHOD SAINATH



ASWATHY N S



KUSHAL SANGW...



SWATI UPADHYAY



VEDIKA JITENDR...



IMUJUNGLA LON...



ADITYA KUMAR S...



DODDAYULA LIK...



YOGESH KUMAR



ANNAPURNE KRL...



SARASWATULA P...



DEVANSHI GUPTA



Syam Sarker



DARSHIT NAGAR



KOUSIK RAJESH



VATSHAL NILESH...



ANSHUL MITTAL



+2

14-Oct-2021 - Google

jamboard.google.com/d/1UiwQG_BvZIIQyFFOeSRqaRIsWEb6KuLdXtVHfRuS0/viewer

14-Oct-2021

1/1

Set background Clear frame

Share

Q1

Q2

IQ RAC



RATHOD SAINATH



ASWATHY N S



KS



SWATI UPADHYAY



VEDIKA JITENDR...



IMJUNGLA LON...



ADITYA KUMAR S...



DODDAYULA LIK...



ANNAPURNE KRI...



SARASWATULA P...



DEVANSHI GUPTA



Syam Sankar



DARSHIT NAGAR



KOUSIK RAJESH



VATSHAL NILESH...



ANSHUL MITTAL



ADHAY PRATAP G...



+2

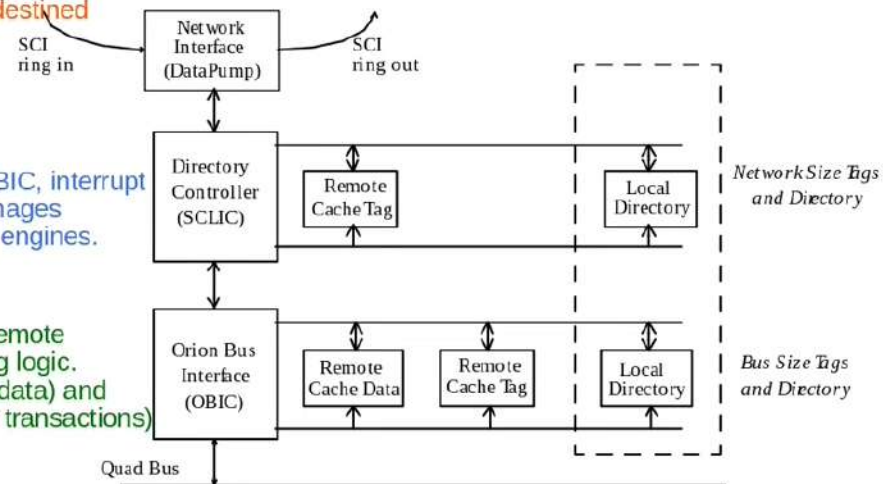


NUMA-Q IQ-link board

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



RATHOD SAINATH



ASWATHY N S



KUSHAL SANGW...



SWATI UPADHYAY



VEDIKA JITENDR...



IMUJUNGLA LON...



ADITYA KUMAR S...



DODDAYULA LIK...



ANNAFURNE KRI...



SARASWATULA P...



DEVANSHI GUPTA



Syam Sankar



DARSHIT NAGAR



KOUSIK RAJESH



VATSHAL NILESH...



ANSHUL MITTAL



ADHAY PRATAP G...



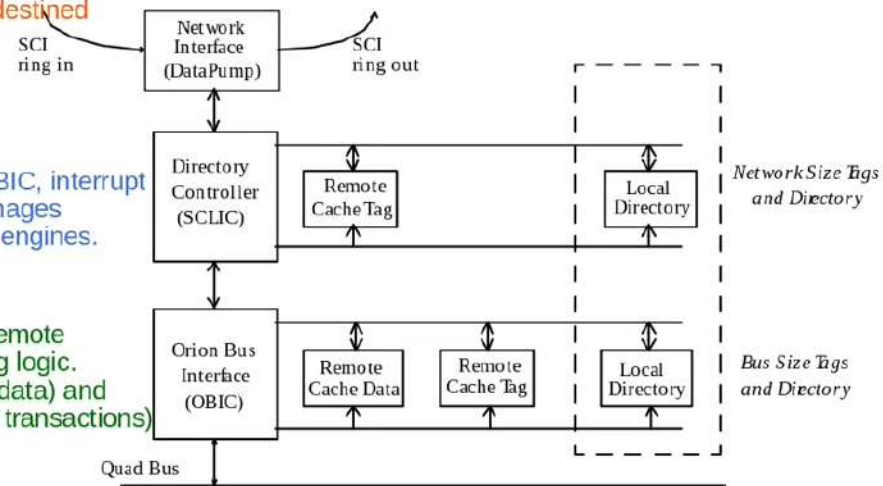
+2

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



RATHOD SAINATH



ASWATHY N S



KUSHAL SANGW...



SWATI UPADHYAY



VEDIKA JITENDR...



IMUJUNGLA LON...



ADITYA KUMAR S...



DODDAYULA LIK...



ANNAFURNE KRI...



SARASWATULA P...



DEVANSHI GUPTA



Syam Sankar



DARSHIT NAGAR



KOUSIK RAJESH



VATSHAL NILESH...



ANSHUL MITTAL



ADHAY PRATAP G...



14-Oct-2021 - Google Jamboard

jamboard.google.com/d/1UiwQG_BvZIIQyFFOeSRqaRIsWEb6KuLbXtVHfRuS0/viewer

14-Oct-2021

Set background Clear frame

Q1 Q2

IQ RAC

Quad

ABIC

Mem Ctrl

SCLIC

Participants:

- RATHOD SAINATH
- ASWATHY N S
- KS
- SU
- KUSHAL SANGW...
- SWATI UPADHYAY
- VK
- VEDIKA JITENDR...
- IMJUNGLA LON...
- AS
- ADITYA KUMAR S...
- DODDAYULA LIK...
- SP
- ANNAPURNE KRI...
- SARASWATULA P...
- DG
- DEVANSHI GUPTA
- Syam Sankar
- DN
- DARSHIT NAGAR
- KOUSIK RAJESH
- VP
- VATSHAL NILESH...
- ANSHUL MITTAL
- ADHAY PRATAP G...
- +2

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.google.com/d/1UiwQG_BvZIIQyFFOeSRqaRIsWEb6KuLdXtVH?RuS0/viewer?f=1

14-Oct-2021

2/2

Share

Set background

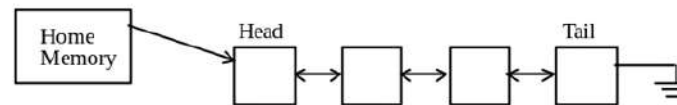
Clear frame

44% (1:09)

Thu, October 14, 14:24

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

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



RATHOD SAINATH



ASWATHY N S



KUSHAL SANGW...



SWATI UPADHYAY



VEDIKA JITENDR...



IMJUNGLA LON...



ADITYA KUMAR S...



DODDAYULA LIK...



ANNAFURNE KRI...



SARASWATULA P...



DEVANSHI GUPTA



Syam Sankar



DARSHIT NAGAR



KOUSIK RAJESH



VATSHAL NILESH...



ANSHUL MITTAL



ABHAY PRATAP G...



+2

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_BvZIIQqyFFoSRqaRIsWEb6KultXtVHfRuS0/viewer?f=2

14-Oct-2021

3/3

Set background Clear frame

The diagram illustrates a linked list structure with the following components and annotations:

- Nodes:** Four circular nodes labeled **Reg**, **Home**, **Fresh**, and **Old Head**.
- Connections:**
 - A blue arrow labeled **Head** points from the **Reg** node to the **Old Head** node.
 - A green arrow labeled **Home** points from the **Home** node to the **Fresh** node.
 - A green arrow labeled **data + Head ptr** points from the **Home** node to the **Old Head** node.
 - A green arrow labeled **SC 1** points from the **Reg** node to the **Old Head** node.
 - A green arrow labeled **pending** points from the **Reg** node to the **Old Head** node.
 - A green arrow labeled **Old Head** points from the **Old Head** node to the **Reg** node.
- Annotations:**
 - Below the **Reg** node: **Reg Invariant** and **Head - Fresh**.
 - Below the **Old Head** node: **Mid valid**, **Head - Fresh**, **Only - Fresh**, and **Tail - valid**.

36% (1:13) Thu, October 14, 14:44

RATHOD SAINATH

AS

ASWATHY N S

KS

SU

SWATI UPADHYAY

VK

IMJUNGLA LON...

VEDIKA JITENDR...

AS

DODDAYULA LIK...

ADITYA KUMAR S...

SP

SARASWATULA P...

ANNAFURNE KRI...

DG

DEVANSHI GUPTA

Syam Sankar

DN

DARSHIT NAGAR

KOUSIK RAJESH

VP

VATSHAL NILESH...

AM

ANSHUL MITTAL

ADHAY PRATAP G...

+2

14-Oct-2021 - Google

jamboard.google.com/d/1UiwQG_BvZIIQyFFOoSRqaRIsWEb6KultXtVHfRuS0/viewer?f=2

14-Oct-2021

3 / 3

Set background Clear frame

Handwritten diagram illustrating a blockchain fork resolution process. The diagram shows a sequence of blocks: 'Reg' (circled), 'Home' (circled), 'Fresh' (circled), and 'Old Head' (circled). A 'pending' block is also shown. Arrows indicate the flow of data and the resolution of a fork. Annotations include 'SC 1', 'data + Head pr.', 'Reg Invariant', 'Head - Fresh', 'Mid Valid', 'Head - Fresh', 'Only - Fresh', 'Tail - Valid', 'Home', 'only Fresh Reg', and 'Head'.

RATHOD SAINATH

AS

ASWATHY N S

KS

SU

SWATI UPADHYAY

VK

IMJUNGLA LON...

VEDIKA JITENDR...

AS

DODDAYULA LIK...

ADITYA KUMAR S...

SP

SARASWATULA P...

ANNAFURNE KRI...

DG

DEVANSHI GUPTA

Syam Sankar

DN

DARSHIT NAGAR

KOUSIK RAJESH

VP

VATSHAL NILESH...

AM

ANSHUL MITTAL

ADHAY PRATAP G...

+2

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 ..



RATHOD SAINATH



ASWATHY N S



KUSHAL SANGW...



SWATI UPADHYAY



VEDIKA JITENDR...



IMJUNGLA LON...



ADITYA KUMAR S...



DODDAYULA LIK...



ANNAFURNE KRI...



SARASWATULA P...



DEVANSHI GUPTA



Syam Sankar



DARSHIT NAGAR



KOUSIK RAJESH



VATSHAL NILESH...



ANSHUL MITTAL



ABHAY PRATAP G...



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 communicate with Home .. as the state is already DIRTY



RATHOD SAINATH



ASWATHY N S



KUSHAL SANGW...



SWATI UPADHYAY



VEDIKA JITENDR...



IMJUNGLA LON...



ADITYA KUMAR S...



DODDAYULA LIK...



ANNAFURNE KRI...



SARASWATULA P...



DEVANSHI GUPTA



Syam Sankar



DARSHIT NAGAR



KOUSIK RAJESH



VATSHAL NILESH...



ANSHUL MITTAL

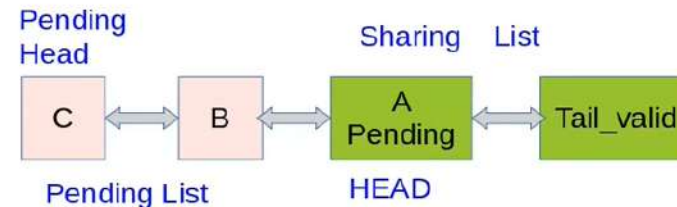


ABHAY PRATAP G...



+2

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



RATHOD SAINATH



ASWATHY N S



KUSHAL SANGW...



SWATI UPADHYAY



VEDIKA JITENDR...



IMJUNGLA LON...



ADITYA KUMAR S...



DODDAYULA LIK...



ANNAFURNE KRI...



SARASWATULA P...



DEVANSHI GUPTA



Syam Sankar



DARSHIT NAGAR



KOUSIK RAJESH



VATSHAL NILESH...



ANSHUL MITTAL



ABHAY PRATAP G...



+2

14-Oct-2021 - Google x +

jamboard.google.com/d/1UiwQG_BvZIIQyFFoSRqaRiSWEb6KultXtVHfRuS0/viewer?f=3

14-Oct-2021

4/4

Set background Clear frame

Diagram illustrating a linked list structure with nodes A and B, and a Home node. The diagram shows various pointers and states:

- Nodes:** A, B, and Home.
- Pointers:** true Head (red), pending Head (yellow), and a pointer from B to A (labeled reg).
- States:** pending (blue), { pending } (black), and { who Head ? } (blue).
- Arrows and Labels:** (1) Read, (2) ptr Old Head, (3) reg, (4) { pending }, (5) ptr B, (6) ptr B, (7) true Head, (8) true Head, (9) pending Head.

Handwritten notes and calculations:

- $\{ \text{who Head ?} \}$
- $= B$
- $= C$

Participants:

- RATHOD SAINATH
- ASWATHY N S
- KS
- SU
- KUSHAL SANGW...
- SWATI UPADHYAY
- VK
- VEDIKA JITENDR...
- IMJUNGLA LON...
- AS
- ADITYA KUMAR S...
- DODDAYULA LIK...
- SP
- ANNAFURNE KRI...
- SARASWATULA P...
- DG
- DEVANSHI GUPTA
- Syam Sankar
- DN
- DARSHIT NAGAR
- KOUSIK RAJESH
- VP
- VATSHAL NILESH...
- ANSHUL MITTAL
- +2
- ADHAY PRATAP G...

14-Oct-2021 - Google Jamboard

14-Oct-2021

Set background Clear frame

4/4

Share

The diagram illustrates a linked list structure with two nodes, A and B. Node A is labeled 'reg' and 'A'. Node B is labeled 'B'. A 'true Head' points to node B. A 'pending Head' points to node B. A 'Home' node is shown with a 'Read' arrow pointing to node B. A 'ptr' (pointer) is shown pointing to node B. A 'who Head?' question is asked, with the answer being 'B' and 'C'. A 'pending list' is shown with a 'sharing' arrow pointing to node A. The diagram includes numbered steps 1 through 9, indicating a sequence of operations or states. Annotations include 'reg', 'Read', 'ptr', and 'who Head?'.

RATHOD SAINATH ASWATHY N S

KS SU

VK

AS

SP

DG

DN

VP AM

+2