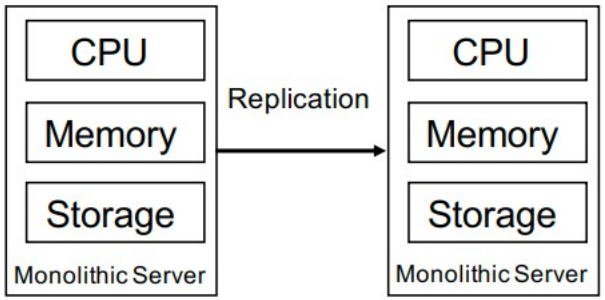
PolarDB Serverless

A Cloud Native Database for Disaggregated Data Centers

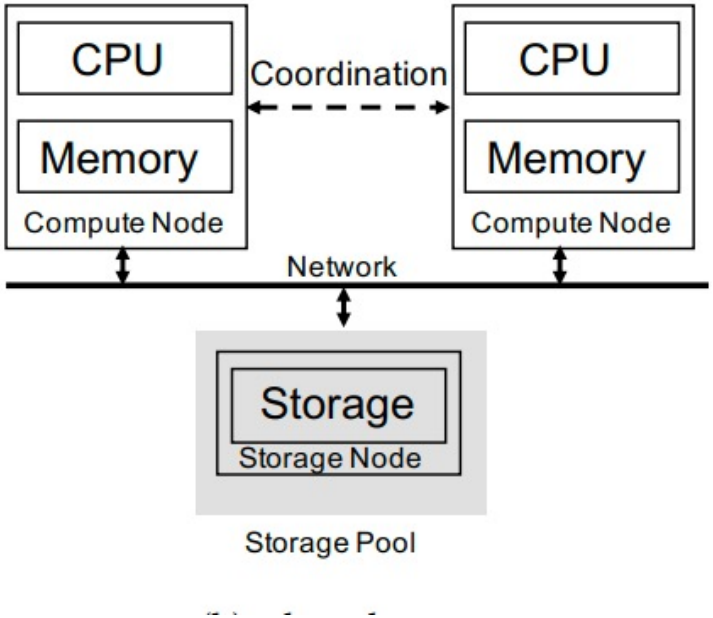
Nian Lu

Previous DB Architecture

- Monolithic Server



- Separation of compute and storage



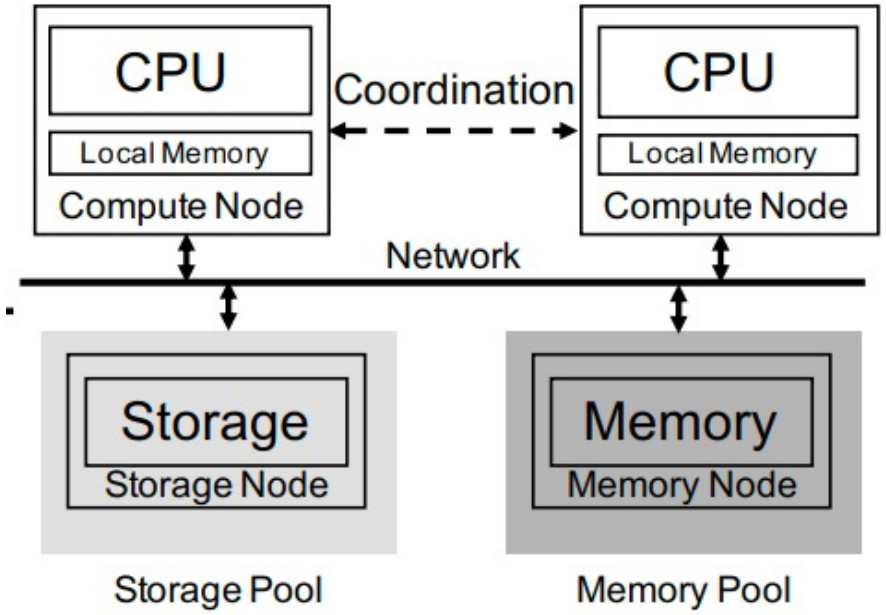
Disaggregation Architecture

Benefits

- More cost-effective auto-scaling
- Better auto-pause capacity
- Scaling transparency

Challenges

- Execute transactions correctly
- Execute transactions efficiently
- Build a reliable system

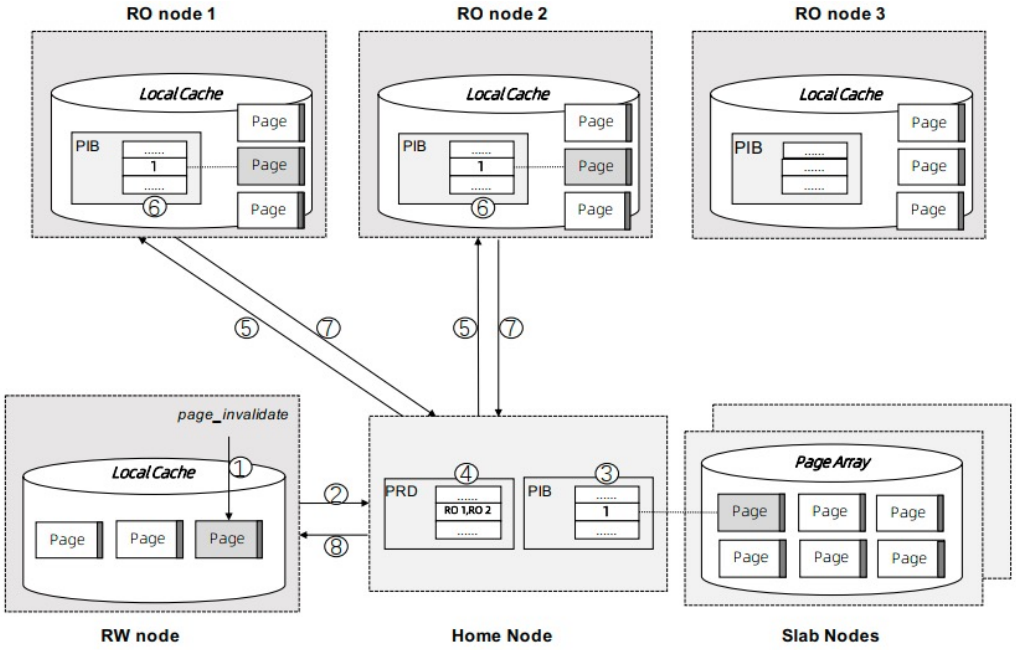


Local Cache of Remote Memory

Problem: Cache Coherency

Solution: Cache Invalidation Mechanism

- set the corresponding bit in PIB
- look up PRD to get a list of RO nodes which hold copies in their local cache
- and then set the corresponding bit in PIB on those RO nodes

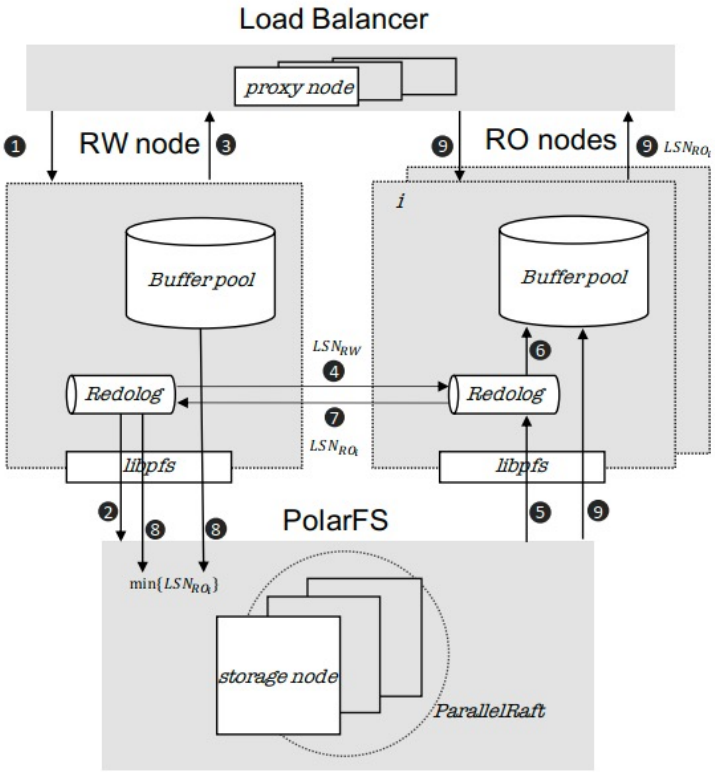


PIB(Page Invalidation Bitmap) :
0 -> latest, 1-> out of data
PAT(Page Address Table): hash table that records the location and reference count of each page
PRD(Page Reference Directory): record reference of page obtained by node

Slab: unit of memory allocation
Slab Node: The memory node that serves slabs
Home Node: first slab is located

B+Tree's Structural Consistency

- Node layer: RW node, RO nodes
- Problem: Structure Modification Operations (SMO)
- Solution: PL(Page Latch)
  - add on to the local page latch, make sure the integrit of index structure in a multi-node environment
- Two steps approaches
  - an optimistic tree traversal for insert/delete
  - a "pessimistic" traversal: If the optimistic traversal finds the leaf page is relatively full or empty and a SMO is possible, then it will restart a "pessimistic" traversal from the root again,



Snapshot Isolation

CTS(centralized timestamp Service)

- **Read-Write Transaction**: acquire the timestamp from CTS twice(cts_read, cts_commit), it writes down cts_commit together with the records it modified
- **read-only transaction**: get cts_read timestamp once
- Reads within a transaction: just read records whose cts_commit < it's cts_read.

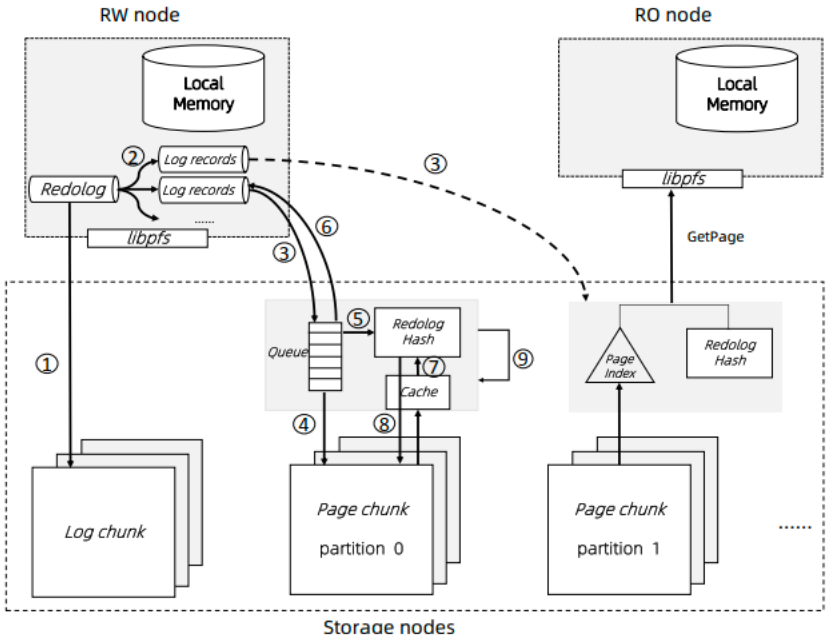
- Problem: cts_commit asychronously update
  - concurrent transactions cannot determine the version of rows without cts_commit.

Solution:

- CTS Log data structure
  - circular array, which records the cts_commit timestamp of the most recent read-write transactions
- One-sided RDMA verbs
  - The CTS timestamp counter is fetched and incremented atomically using RDMA CAS
  - CTS Log is placed in a contiguous memory region registered to RDMA NIC

Other technologies

- Page materization offloading
  - seperate log chunk and page chunk
  - replay redo log to pages on storage node
- Auto scaling
  - proxy node take responsibility for seamless switch



Reliability and Failure Recovery

- Database Node Recovery
  - A failed Read-Only node can be easily replaced with a new one using pages in the shared memory
  - New Read-Write Node
    - plays redo logs to recovery committed transactions
    - synchronize remote memory, evict invalidate and newer pages
    - plays undo logs to rollback uncommitted transactions in the background
- Memory Node Recovery
  - Recovered from storage
- Cluster Recovery
  - Recovered from storage

Thanks!