



GridGain and Apache Ignite In-Memory Performance with Durability of Disk

Dmitriy Setrakyan
Apache Ignite PMC
GridGain Founder & CPO

<http://ignite.apache.org>



#apacheignite

Agenda

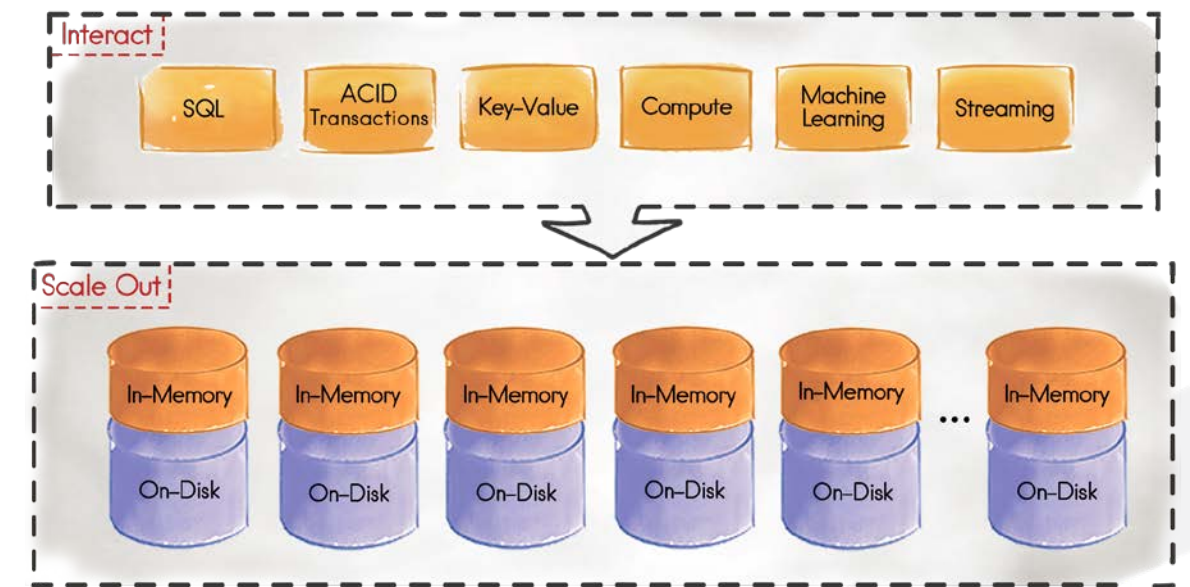
- What is GridGain and Ignite
 - Key Components
 - Feature Comparison
 - Ignite and CAP Theorem
- Ignite Durable Memory
 - In-Memory and On-Disk
 - Architecture Dive-In
- Q&A

GridGain 8.1 and Ignite 2.1

From In-Memory to Memory-Centric

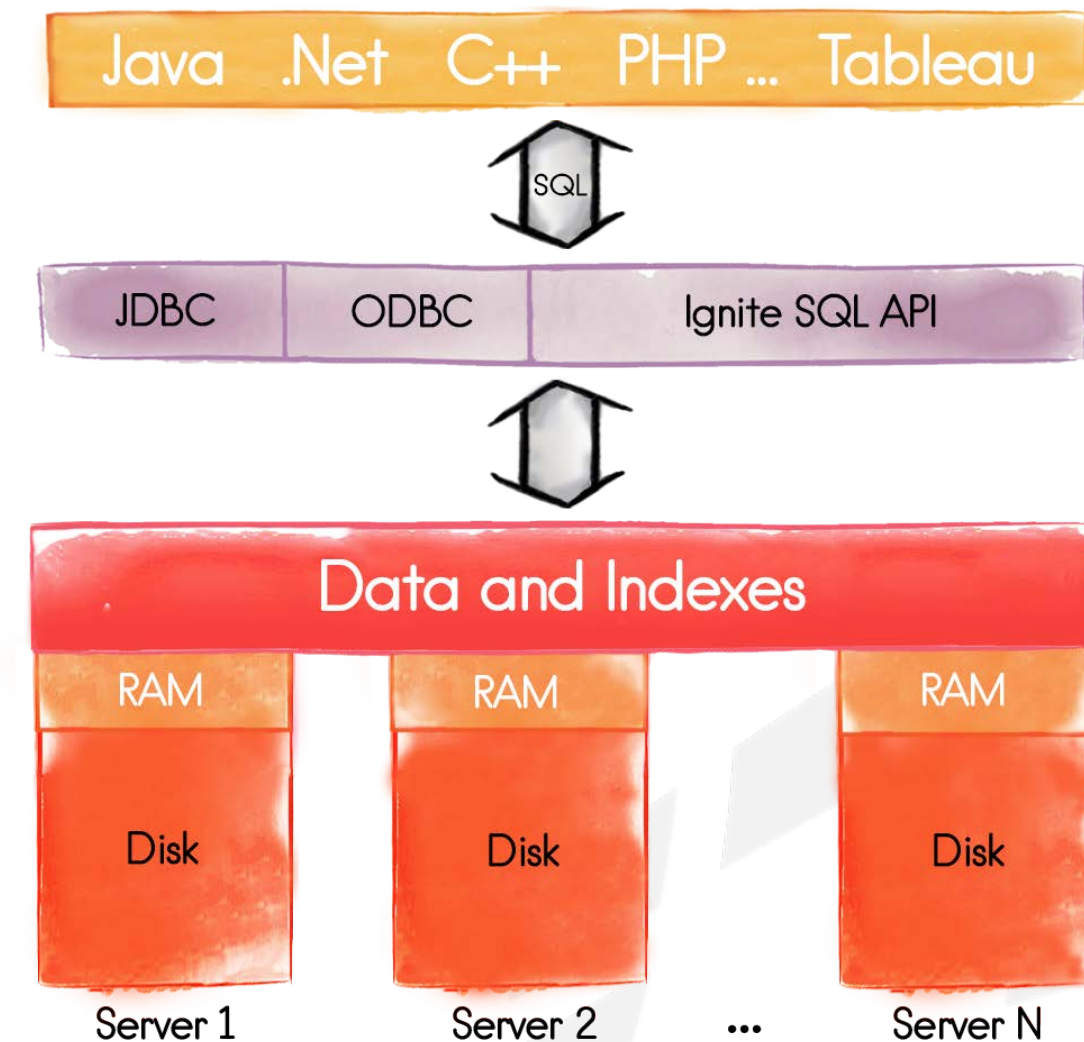
Ignite is a *memory-centric* platform

- combining a distributed **SQL** database
- and a **key-value** data grid,
- that is **ACID**-compliant,
- always **available**
- and horizontally **scalable**

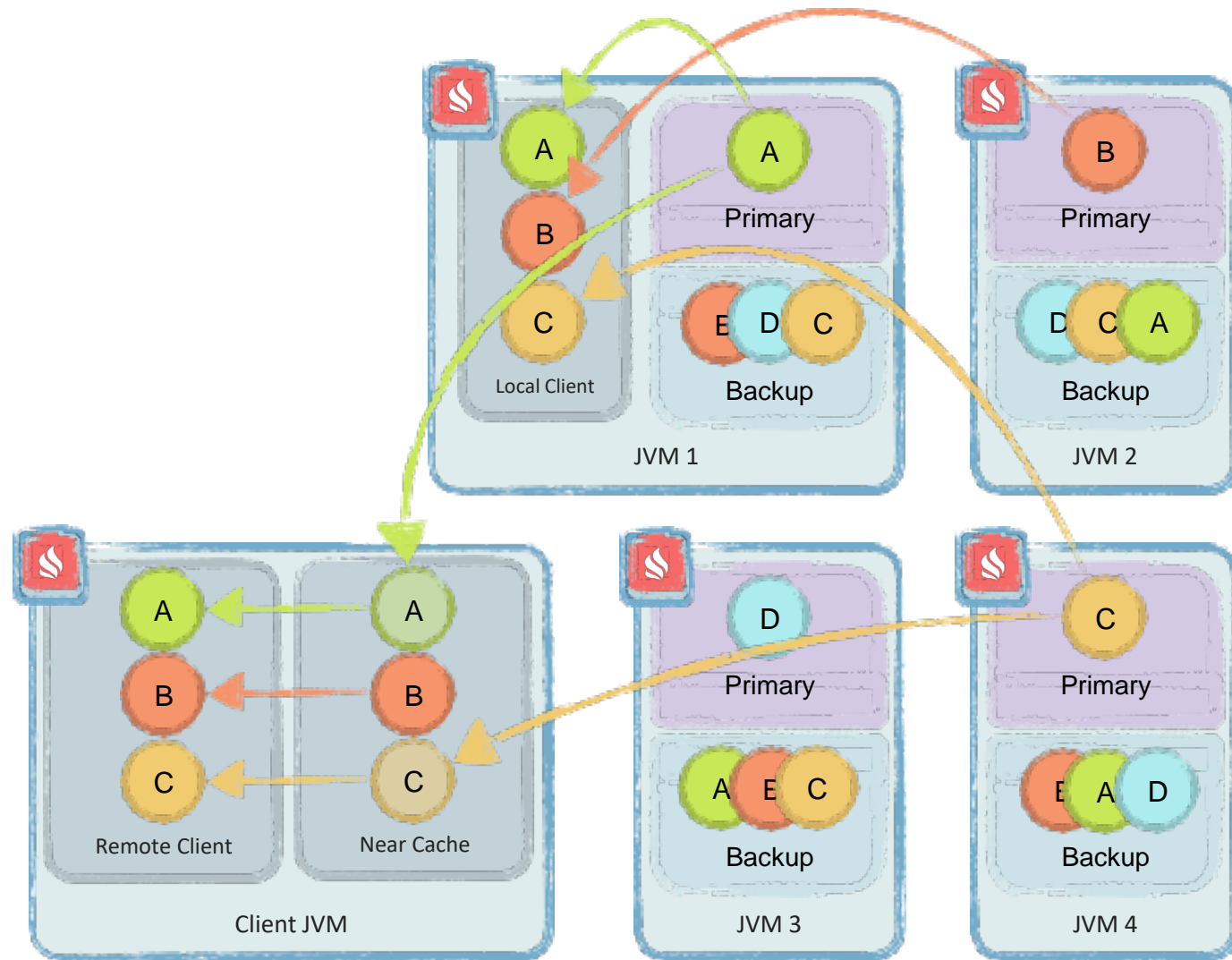


Distributed SQL Database

- In-Memory and On-Disk
- Replicated or Partitioned
- ACID-compliant
- Distributed Joins
- B+Tree Indexes

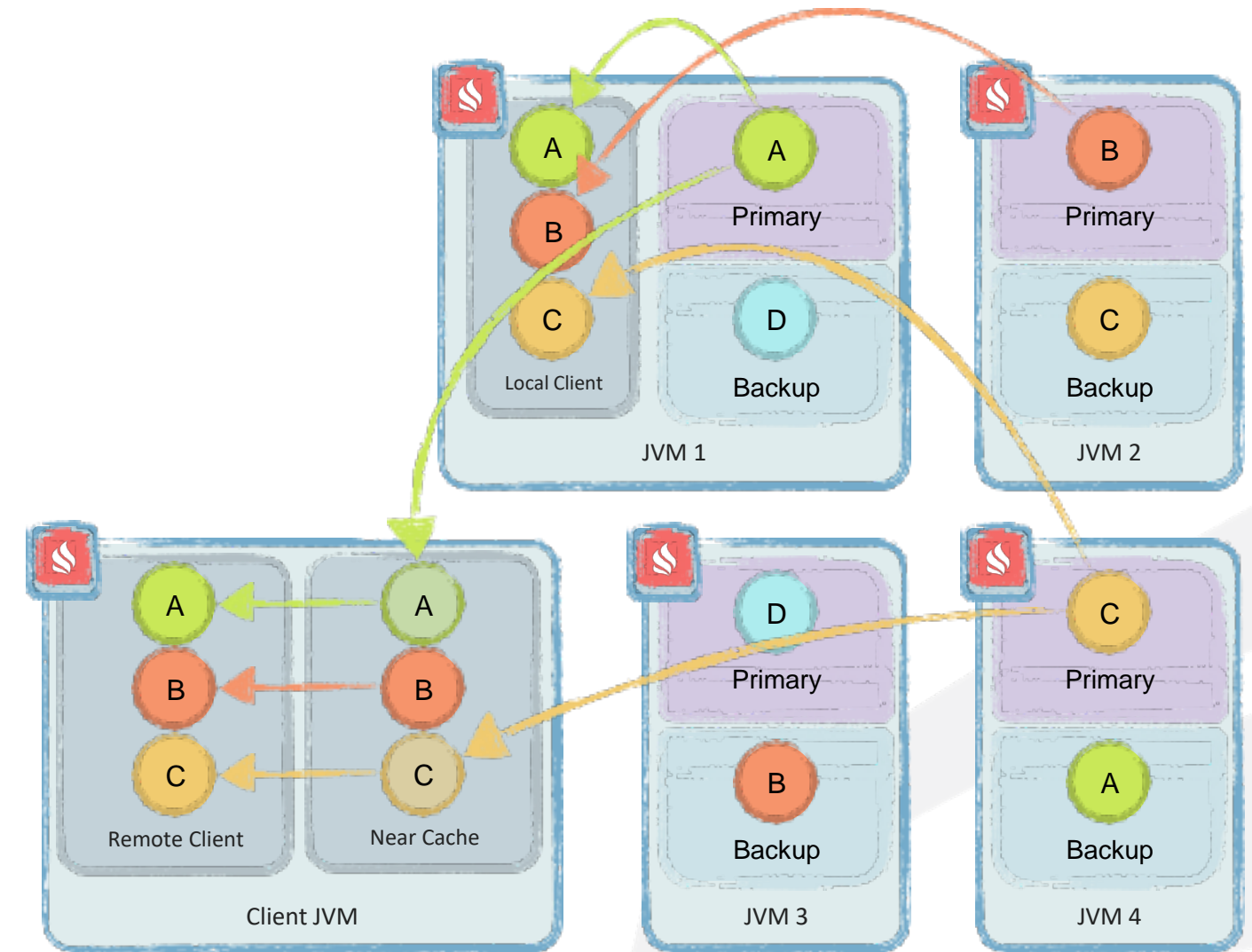


Replication Schemes



Full Replication

backups = Number of Nodes

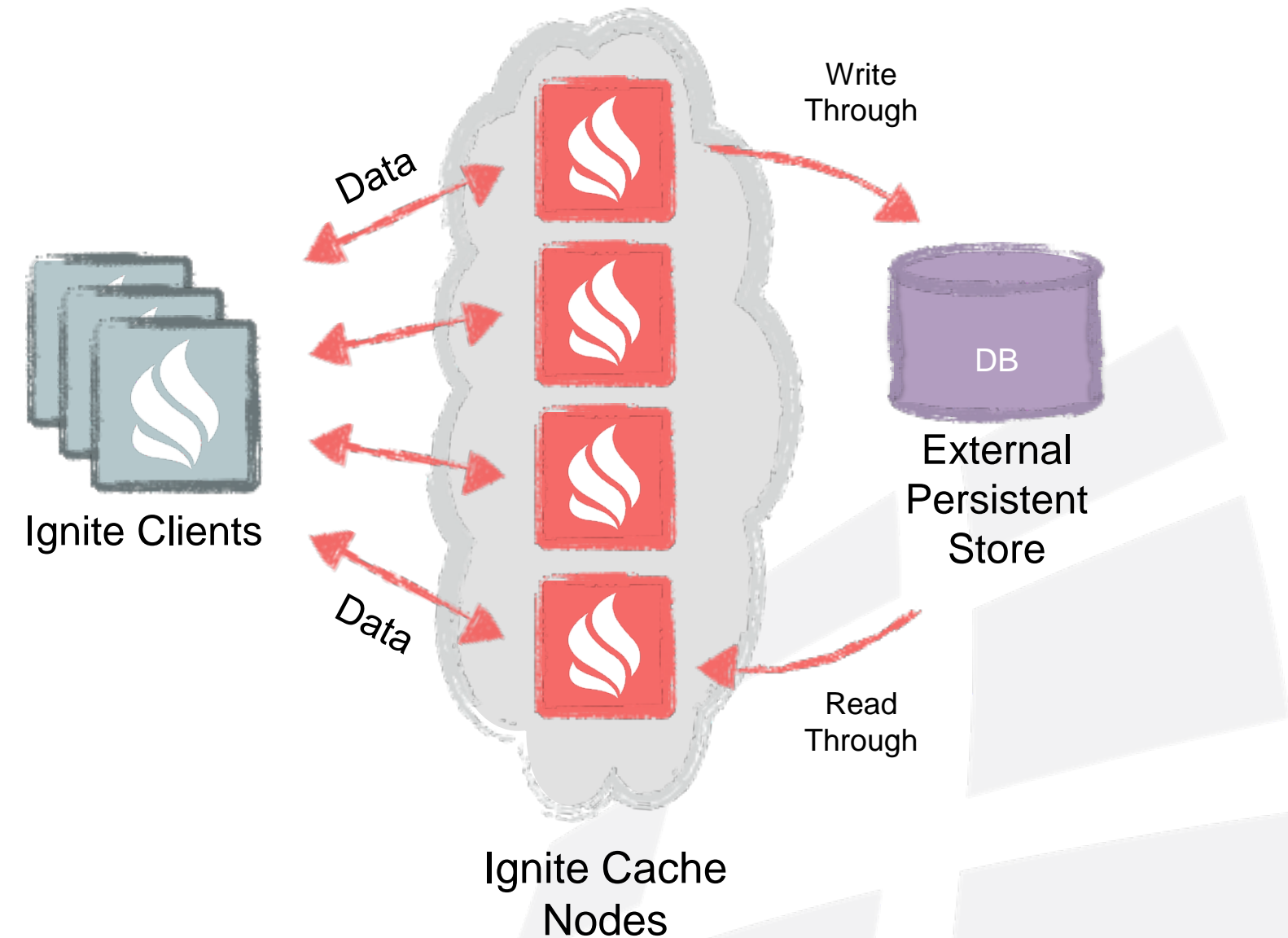


Partitioned Replication

backups = 1

Key-Value Data Grid

- In-Memory Key-Value Store
- ACID Compliant
- Collocated Processing
- Persistence
 - Native Ignite persistence
 - Pluggable 3rd party persistence



Memory-Centric vs Disk-Centric

Memory Centric =

durable +

in-memory **store** +

collocated processing

Disk Centric =

durable +

in-memory **caching** +

client-server processing

Durable Distributed ACID Systems

- Memory Centric
 - Apache Ignite (collocated compute)
 - NuoDB (kind of... stored procedures)
- Disk Centric
 - Google Spanner
 - Cockroach DB



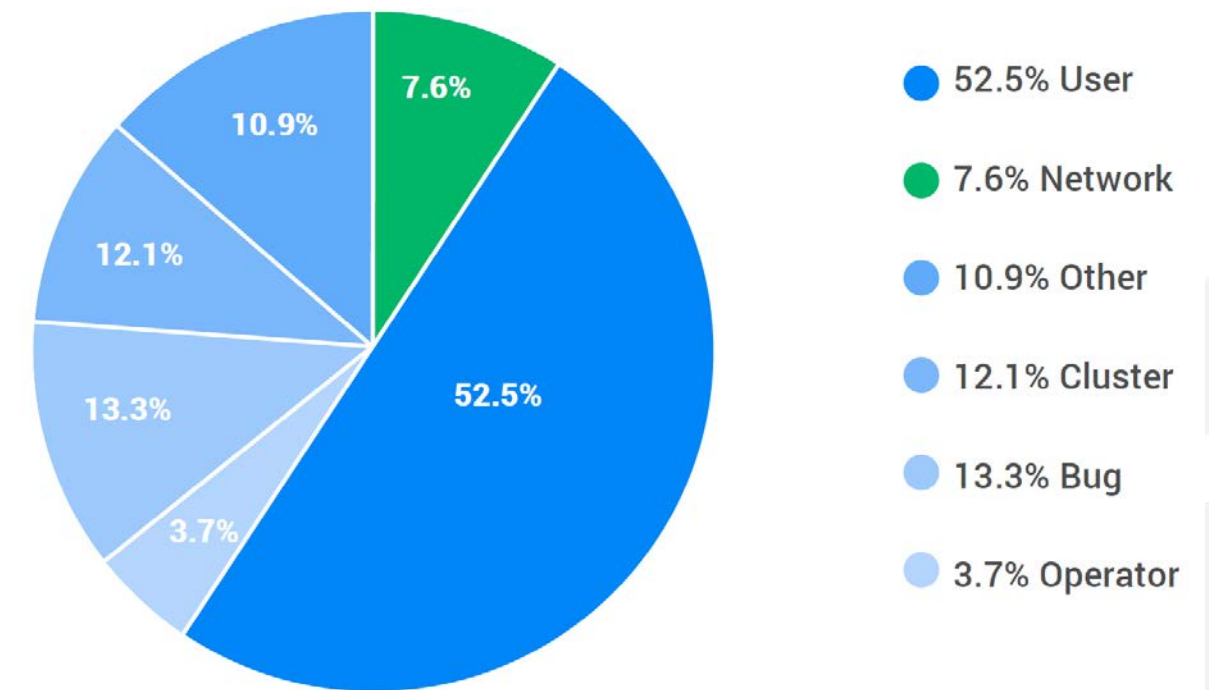
Key Feature Comparison

	RDBMS	NoSQL	IMDG	Apache Ignite
Scalability	x vertical	✓ horizontal	✓ horizontal	✓ horizontal
Availability	x failover only	✓ high	✓ high	✓ high
Consistency	✓ strong	x eventual	✓ strong	✓ strong
In Memory	✓	x	✓	✓
Persistence	✓	✓	x	✓
SQL	✓	x	x	✓ with JOINS
Key-Value	x	✓	✓	✓
Collocated Processing	x	x	✓	✓

Ignite and CAP Theorem

- CAP
 - Consistency (C)
 - Availability (A)
 - Network Partition Tolerance (P)
 - Possible: CA, CP, or AP
 - Impossible: CAP
- Ignite
 - Strongly CP
 - Effectively CA
 - Definitely not AP

Google Spanner Network Outages



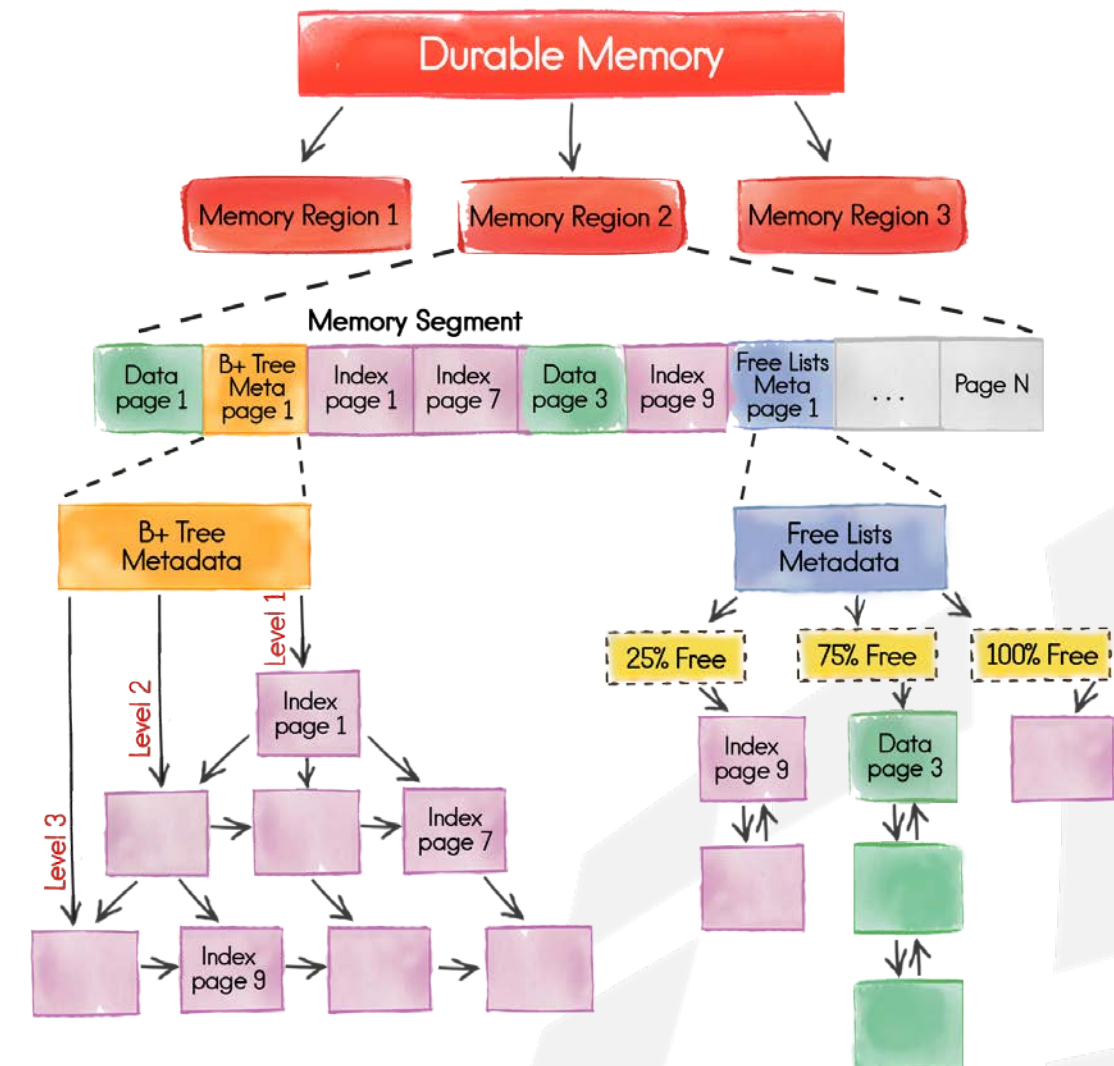
Ignite Durable Memory

Ignite Durable Memory

- In-Memory
 - Off-Heap memory
 - Removes noticeable GC pauses
 - Automatic Defragmentation
 - Predictable memory consumption
 - Boosts SQL performance
- On-Disk (New)
 - Optional Persistence
 - Flash, SSD, Intel 3D Xpoint
 - Stores superset of data
 - Fully Transactional
 - Write-Ahead-Log (WAL)
 - Instantaneous Restarts

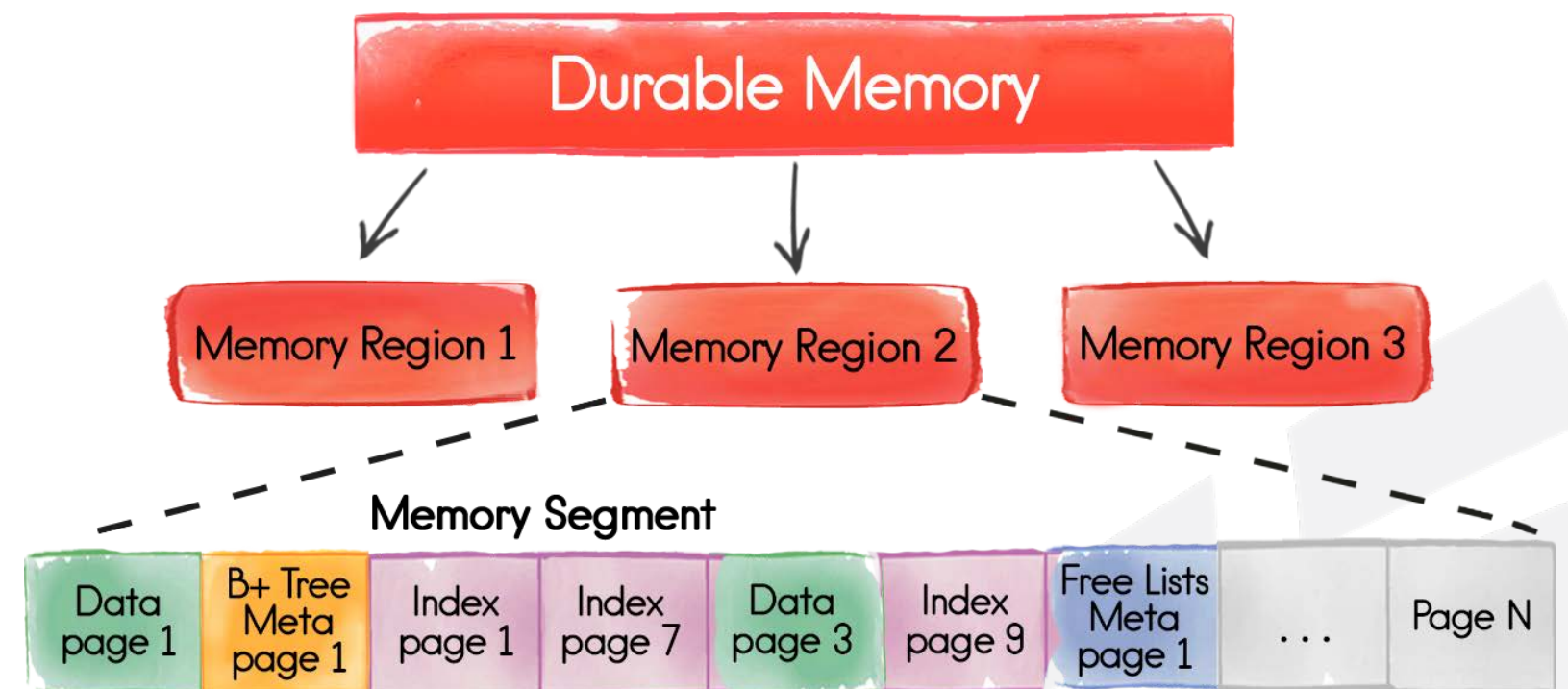
Durable Memory Architecture

- Memory split into *regions*
- Regions split into *segments*
- Segments split into *pages*
- Pages can be swapped to disk
- Free-Lists track free space
- Pages can store:
 - Data
 - Indexes
 - Metadata



Memory Regions and Segments

- Consumes Total Space Available
 - RAM and disk
 - Default Behavior
- Memory Region
 - Initial & max size
 - Data eviction mode
- Memory Segment
 - Continuous physical space
 - Regions grow by Segments
- Page
 - Fixed-length block

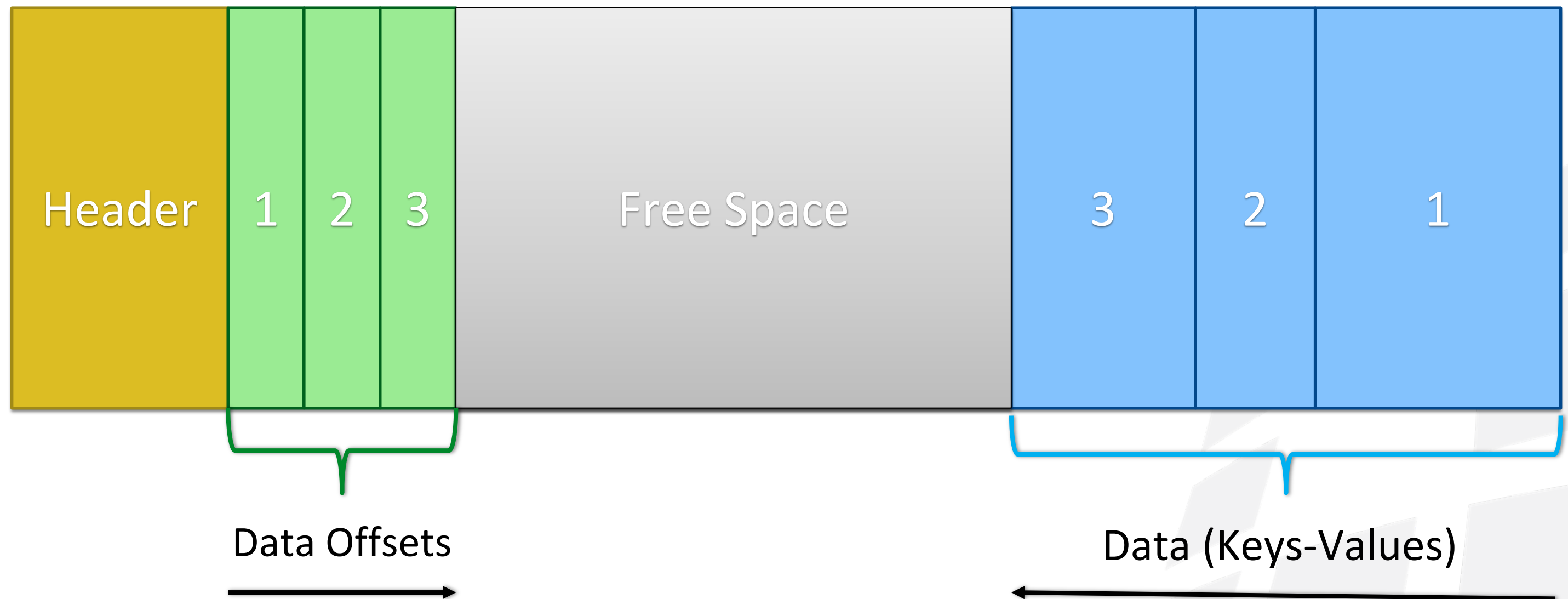


Pages

- Fixed-length block
- Frugal memory usage
- Automatic defragmentation
- Data Page
 - Stores key-value pairs
- Index Page
 - Linked by B+Tree
 - Refers to data pages
- Meta Page

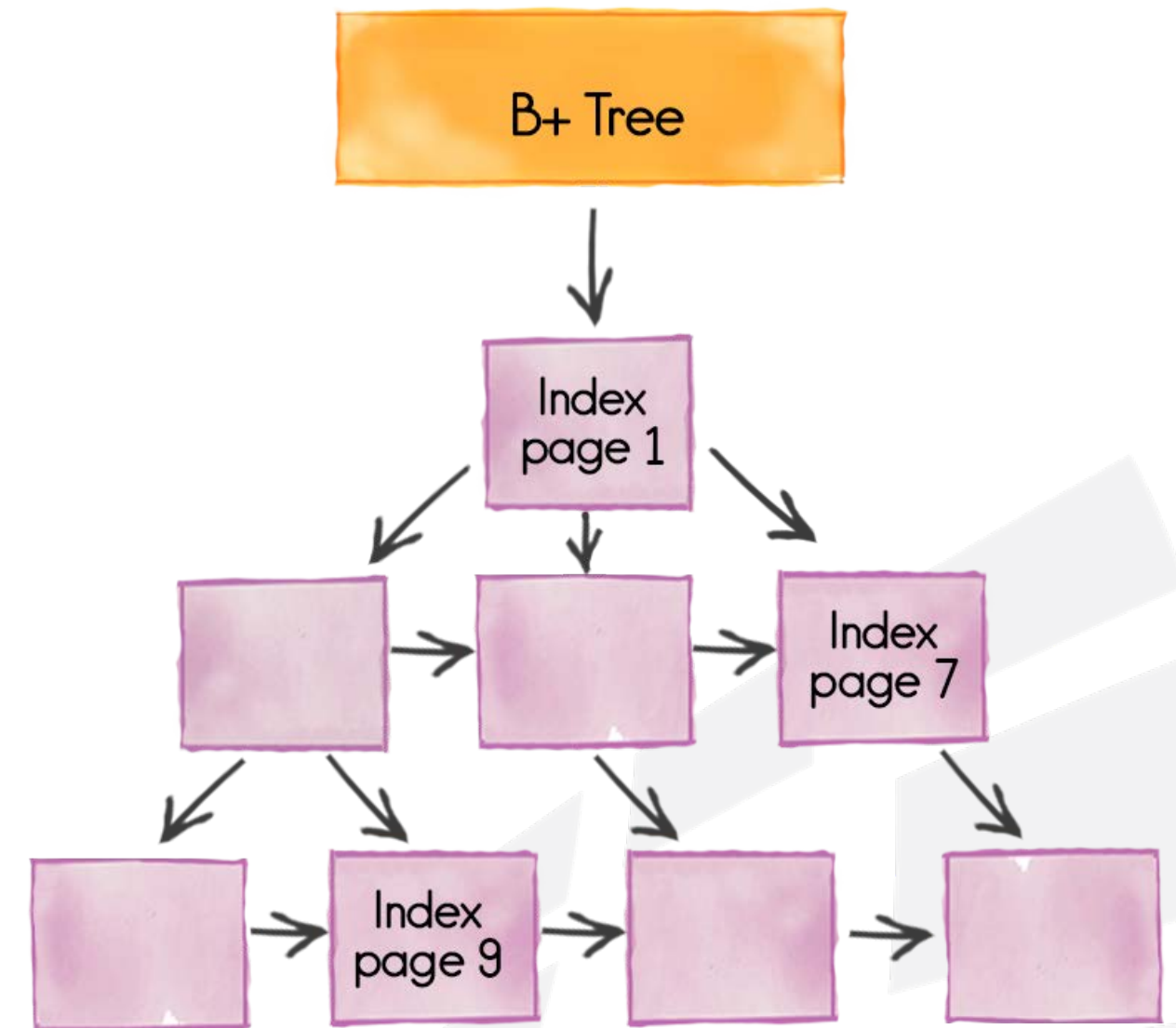


Data Page

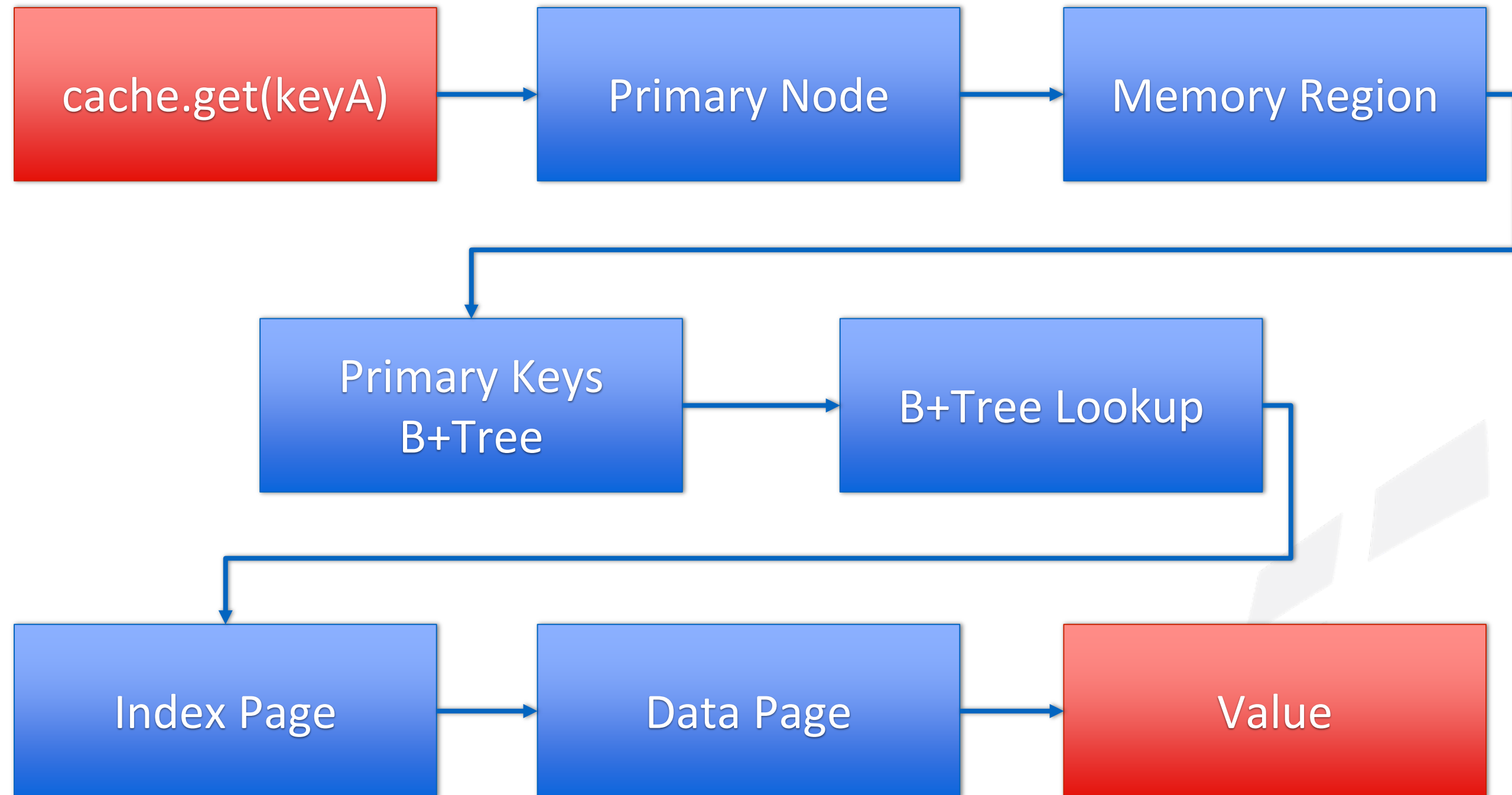


B+Tree Indexes

- Self-balancing Tree
 - Memory & Disk
- Links and Sorts Index Pages
- Sorted Index
 - Secondary indexes
- Hash Index
 - Primary indexes
 - Hash code based sorting

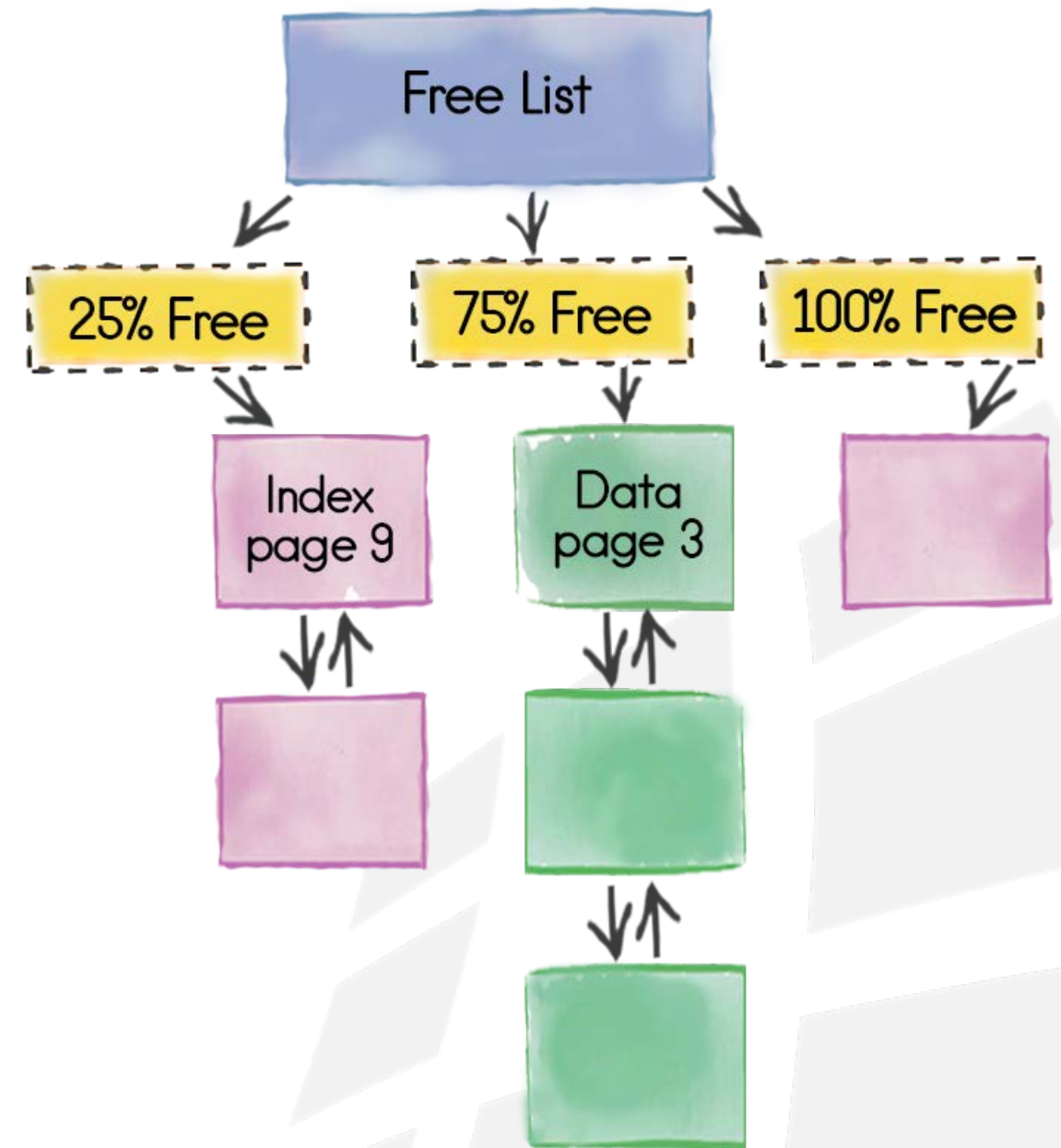


B+Tree: Key-Value Read



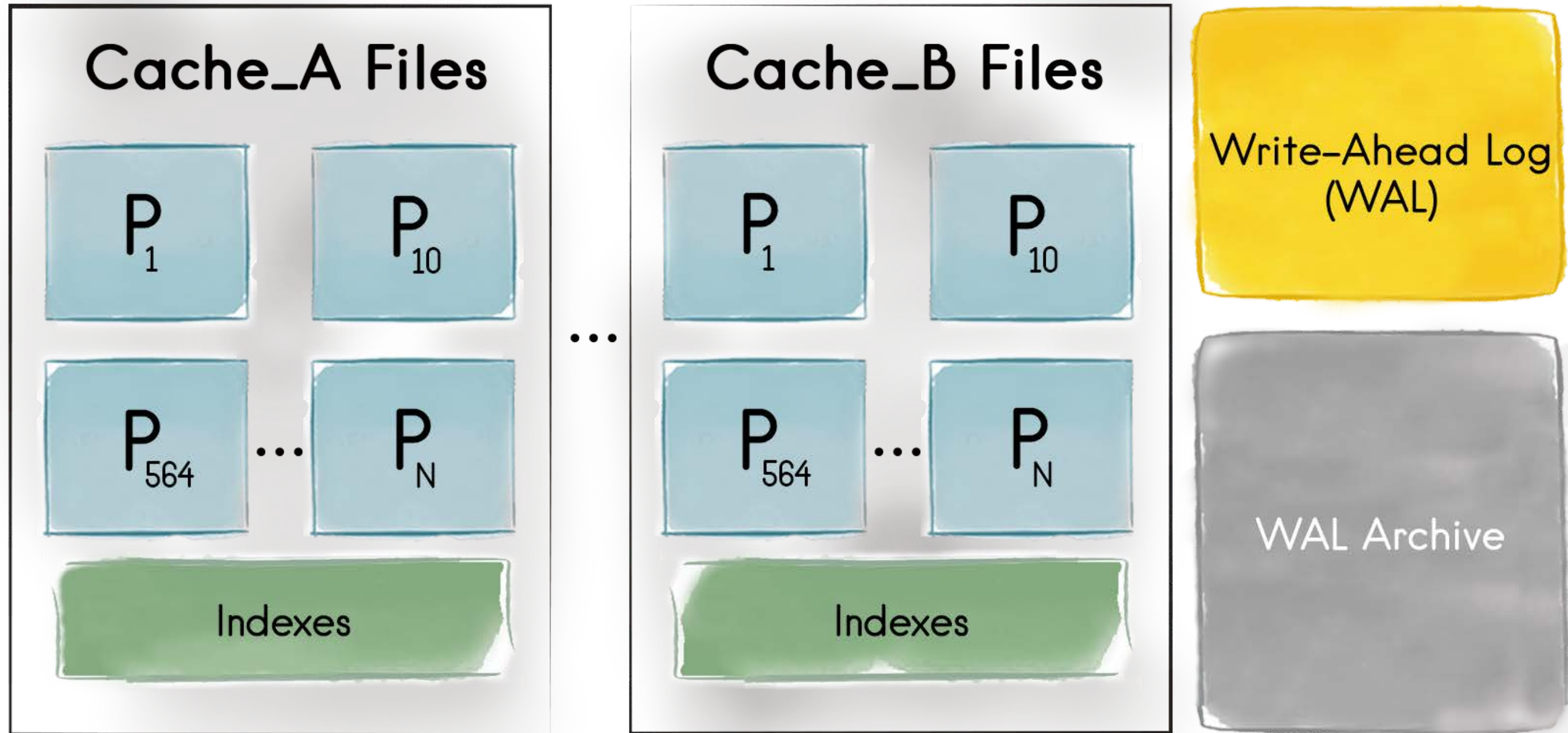
Free Lists

- Tracks Pages of about Equal Free Space
 - 25% free
 - 75% free
 - etc.
- Essential for Update Operations
 - Returns a page with min space needed
 - Reduces fragmentation
 - Lowers compaction activity



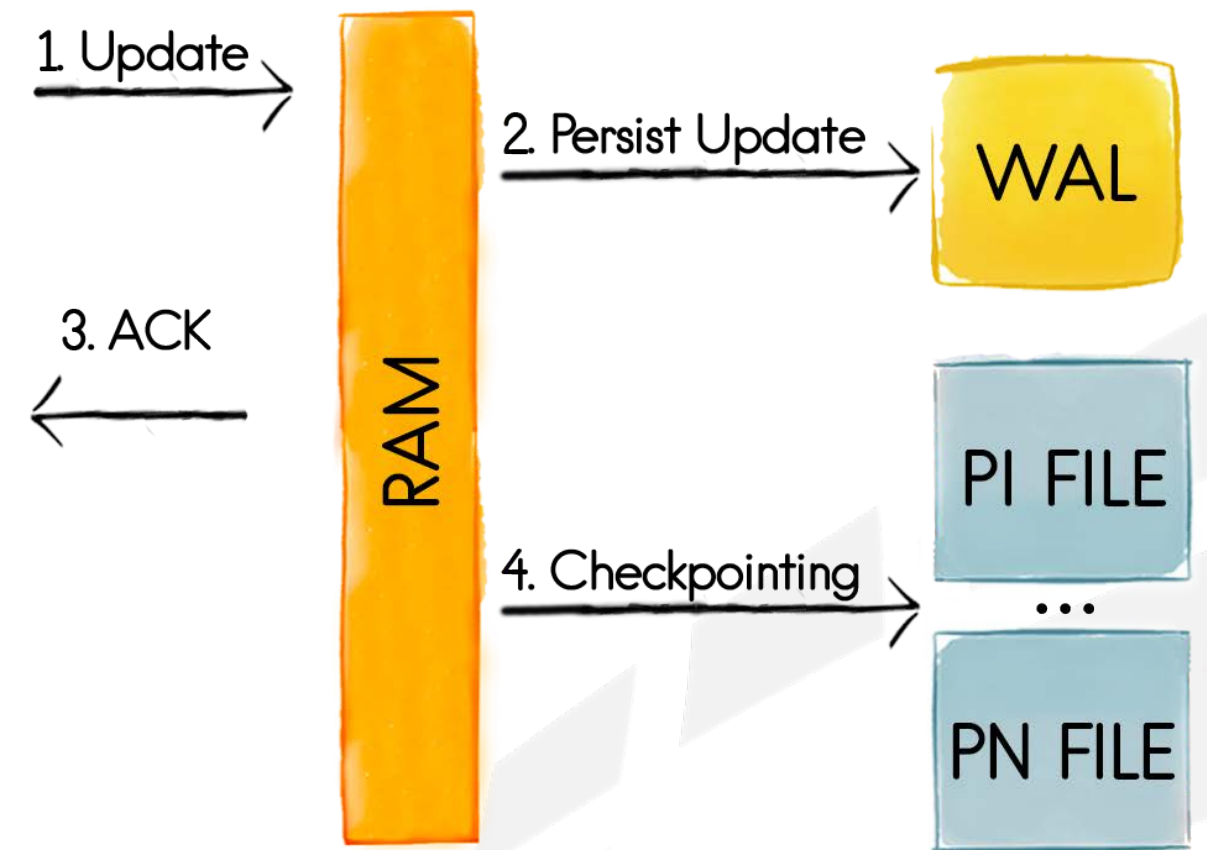
Ignite Persistence

Per-Node Architecture



Consistency and Durability

- Write-Ahead Log (WAL)
 - Updates appended to node's WAL
 - WAL propagates updates to disk
 - Provides recovery mechanism
- Checkpointing
 - Triggered periodically based upon config
 - Copy dirty pages from RAM to disk
 - Reduces WAL size



Snapshots and Backups

- Full snapshots
 - Full state
 - May take long time
- Incremental snapshots
 - Partial state
 - Only delta since last snapshot
- Scheduled snapshots
- Restore on different clusters
- Fully managed
- Available in **GridGain Ultimate Edition**

The screenshot displays the GridGain Ultimate Edition web interface. The top navigation bar includes the GridGain logo, links for 'Configure', 'Queries', and 'Monitoring', a 'Start Demo' button, and user information for 'Cluster 6CD0FF49' and 'Alex Smith'. The main section is titled 'Snapshots' and features a 'Download Web Agent' button and a 'Set Schedule' button. Below this is a table for 'My schedule' with columns for Name, Command, Schedule, Type, Last Run, Next Run, and Enable. The table lists three items: 'Snapshot' (create), 'Cleanup' (delete), and 'Move to archive' (move), all with 'Enable' toggles turned on. Below the schedule table is a section for 'All' snapshots, showing '1 item selected' and a period from '19/06/2017 00:00' to '20/06/2017 00:00'. This section contains a table with columns for ID, Start Time, Duration, Mode, Type, and Caches. The table lists four items, with the first one selected (ID: 1 497 863 437 767 INC 2, Start Time: 19/06/2017 16:10, Mode: console, Type: FULL, Caches: 5).

Name	Command	Schedule	Type	Last Run	Next Run	Enable
Snapshot	create					<input checked="" type="checkbox"/>
Cleanup	delete					<input checked="" type="checkbox"/>
Move to archive	move					<input checked="" type="checkbox"/>

ID	Start Time	Duration	Mode	Type	Caches
1 497 863 437 767 INC 2	19/06/2017 16:10		console	FULL	5
1 497 863 443 613	19/06/2017 16:10		wizard	INC	5
1 497 863 448 438	19/06/2017 16:10		console	INC	5
1 497 863 768 026 INC 1	19/06/2017 16:16		console	FULL	1



ANY QUESTIONS?

Thank you for joining us. Follow the conversation.

<http://ignite.apache.org>



#apacheignite