The Bw-Tree: A B-Tree On Steroids

Justin Levandoski
David Lomet
Sudipta Sengupta

The Bw-Tree: What is it?

"A Latch-free, Log-structured B-tree for Multi-core Machines with Large Main Memories and Flash Storage"

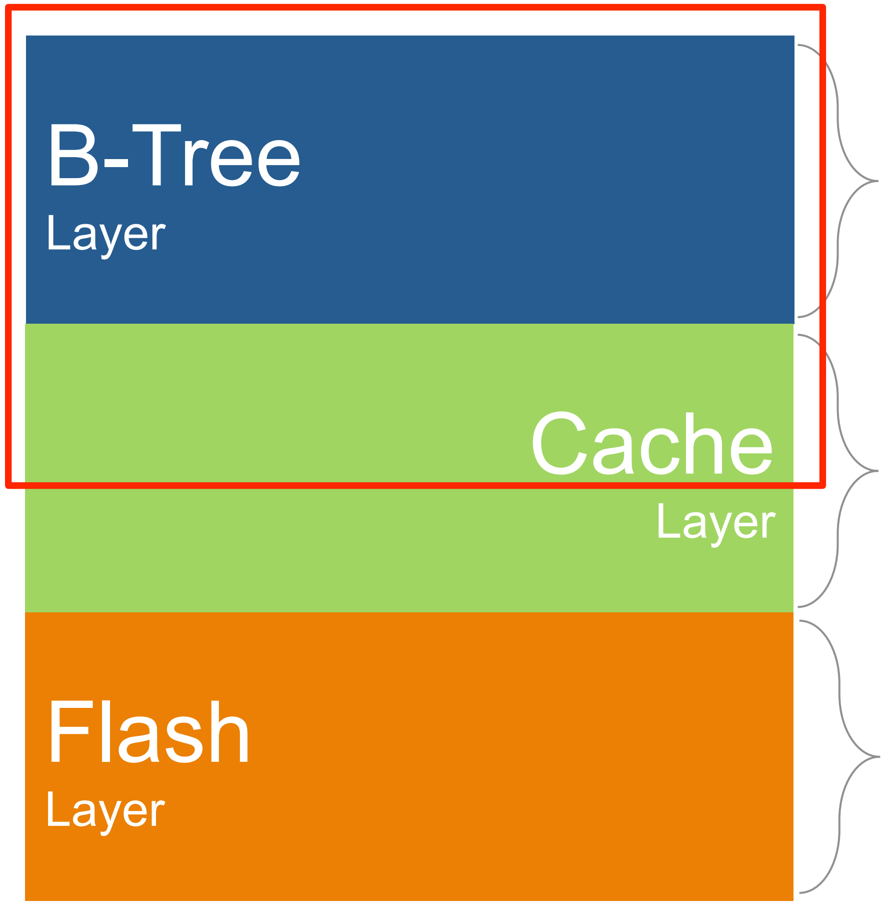
Bw = "Buzz Word"

The Buzz Words: Attacking Two Trends

- Multi-core + large main memories
  - Latch (lock) free
    - Worker threads do not set latches for any reason
    - No latch contention
  - “Delta” updates
    - No updates in place
    - Reduces cache invalidation
- Flash storage
  - Good at random reads and sequential reads/writes
  - Bad at random writes
  - Use flash as append log
  - Implement log-structured storage layer over flash
  - Must run efficiently on both expensive AND cheap devices

Architecture

Let’s talk about memory first…

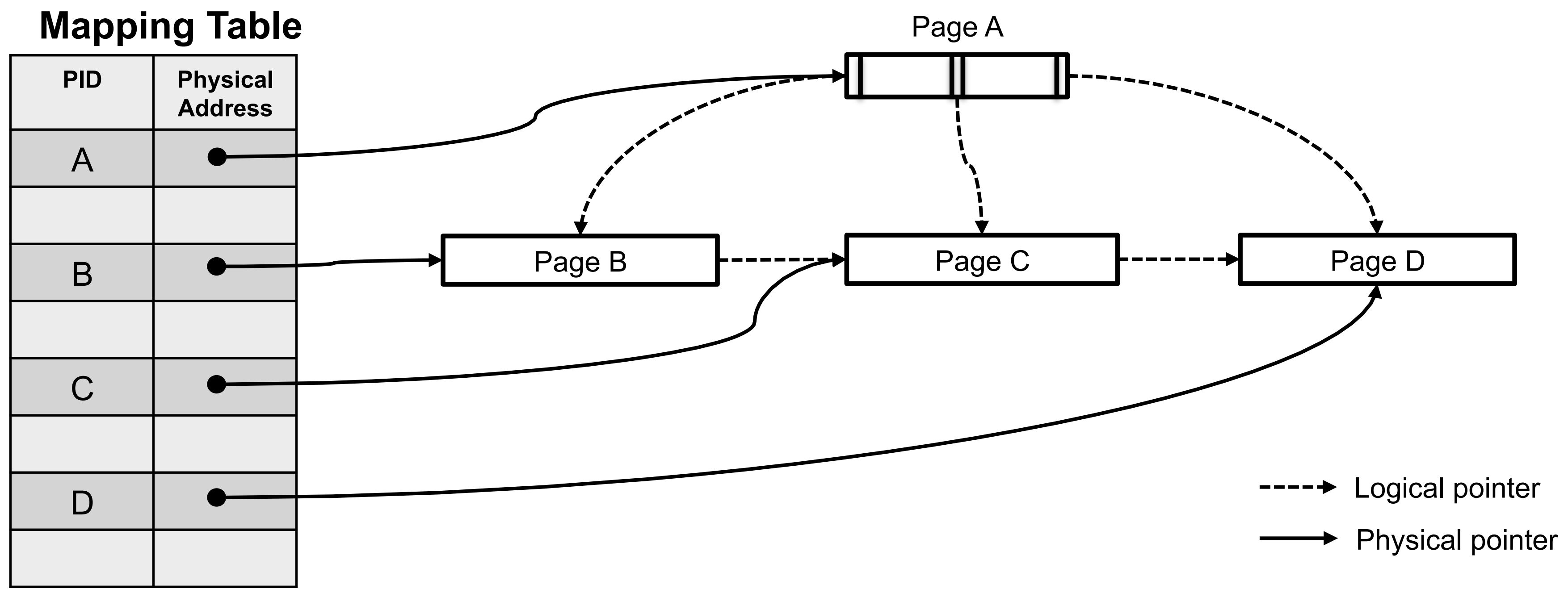


- “CRUD” API
- B-tree search/update logic
- In-memory pages only

- Logical page abstraction for B-tree layer
- Brings pages from flash to RAM as necessary

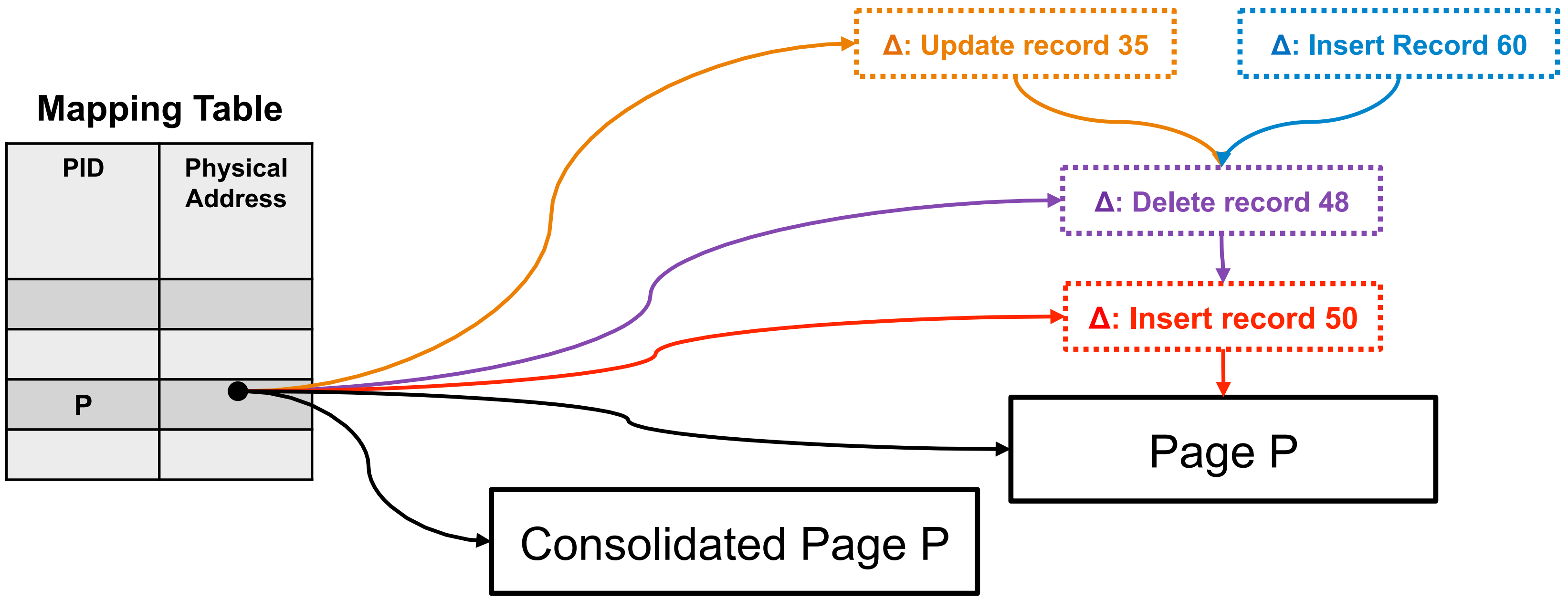
- Sequential writes to log-structured storage
- Flash garbage collection

Logical Pages and Mapping Table



- Logical pages identified by mapping table index
- Isolates update to a single page
- Important for latch-free behavior and log-structuring

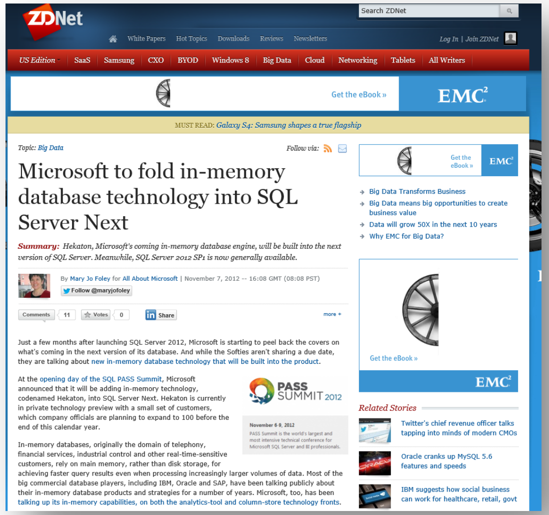
Delta Updates



- Each page update produces a new address (the delta).
- Install new page address in map using compare-and-swap.
- Only one winner on concurrent update to the same address.
- Eventually install new consolidate page with deltas applied.
- Single-page updates are easy, solved node splits and deletes.

Microsoft SQL Server Hekaton

- Main-memory optimized OLTP engine
  - Engine is completely latch-free
  - Multi-versioned, optimistic concurrency control (VLDB 2012)
- Bw-tree is the ordered index in Hekaton



MUST READ: *Galaxy S4: Samsung shapes a true flagship*

Topic: Big Data

Microsoft to fold in-memory database technology into SQL Server Next

***Summary:** Hekaton, Microsoft's coming in-memory database engine, will be built into the next version of SQL Server. Meanwhile, SQL Server 2012 SP1 is now generally available.*

By Mary Jo Foley for All About Microsoft | November 7, 2012 -- 16:08 GMT (08:08 PST)
Follow @maryjofoley

Just a few months after launching SQL Server 2012, Microsoft is starting to peel back the covers on what's coming in the next version of its database. And while the Softies aren't sharing a due date, they are talking about new in-memory database technology that will be built into the product.

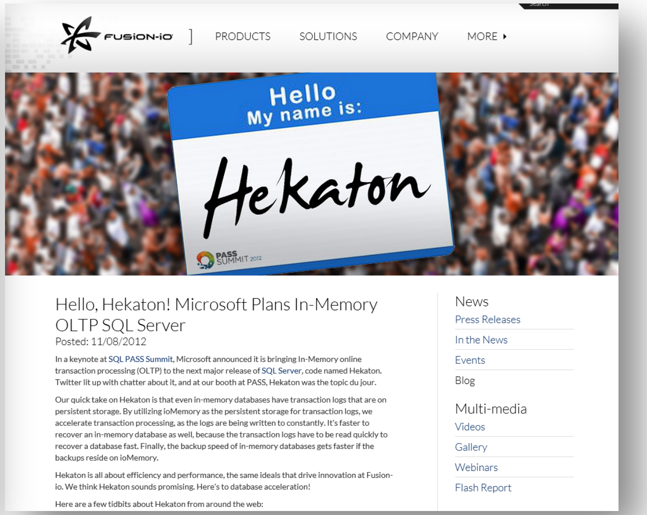
At the opening day of the SQL PASS Summit, Microsoft announced that it will be adding in-memory technology, codenamed Hekaton, into SQL Server Next. Hekaton is currently in private technology preview with a small set of customers, which company officials are planning to expand to 100 before the end of this calendar year.

November 6-9, 2012
PASS Summit is the world's largest and most intensive technical conference for Microsoft SQL Server and BI professionals.

In-memory databases, originally the domain of telephony, financial services, industrial control and other real-time-sensitive customers, rely on main memory, rather than disk storage, for achieving faster query results even when processing increasingly larger volumes of data. Most of the big commercial database players, including IBM, Oracle and SAP, have been talking publicly about their in-memory database products and strategies for a number of years. Microsoft, too, has been talking up its in-memory capabilities, on both the analytics-tool and column-store technology fronts.

- Big Data Transforms Business
- Big Data means big opportunities to create business value
- Data will grow 50X in the next 10 years
- Why EMC for Big Data?

Related Stories
- Twitter's chief revenue officer talks tapping into minds of modern CMOs
- Oracle cranks up MySQL 5.6 features and speeds
- IBM suggests how social business can work for healthcare, retail, govt



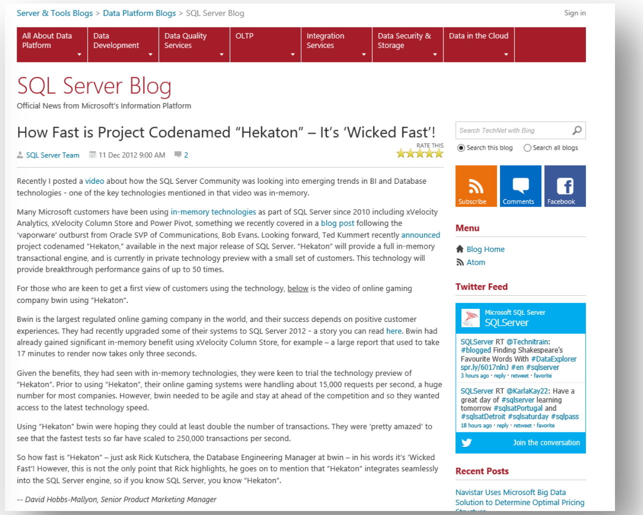
Hello, Hekaton! Microsoft Plans In-Memory OLTP SQL Server
Posted: 11/08/2012

In a keynote at SQL PASS Summit, Microsoft announced it is bringing In-Memory online transaction processing (OLTP) to the next major release of SQL Server, code named Hekaton. Twitter lit up with chatter about it, and at our booth at PASS, Hekaton was the topic du jour.

Our quick take on Hekaton is that even in-memory databases have transaction logs that are on persistent storage. By utilizing ioMemory as the persistent storage for transaction logs, we accelerate transaction processing, as the logs are being written to constantly. It's faster to recover an in-memory database as well, because the transaction logs have to be read quickly to recover a database fast. Finally, the backup speed of in-memory databases gets faster if the backups reside on ioMemory.

Hekaton is all about efficiency and performance, the same ideals that drive innovation at Fusion-io. We think Hekaton sounds promising. Here's to database acceleration!

Here are a few tidbits about Hekaton from around the web:



SQL Server Blog
Official News from Microsoft's Information Platform

How Fast is Project Codenamed "Hekaton" – It's 'Wicked Fast'!

SQL Server Team 11 Dec 2012 9:00 AM 2

Recently I posted a video about how the SQL Server Community was looking into emerging trends in BI and Database technologies - one of the key technologies mentioned in that video was in-memory.

Many Microsoft customers have been using in-memory technologies as part of SQL Server since 2010 including xVelocity Analytics, xVelocity Column Store and Power Pivot, something we recently covered in a blog post following the 'vaporware' outburst from Oracle SVP of Communications, Bob Evans. Looking forward, Ted Kummert recently announced project codenamed "Hekaton," available in the next major release of SQL Server. "Hekaton" will provide a full in-memory transactional engine, and is currently in private technology preview with a small set of customers. This technology will provide breakthrough performance gains of up to 50 times.

For those who are keen to get a first view of customers using the technology, below is the video of online gaming company bwin using "Hekaton".

Bwin is the largest regulated online gaming company in the world, and their success depends on positive customer experiences. They had recently upgraded some of their systems to SQL Server 2012 - a story you can read here. Bwin had already gained significant in-memory benefit using xVelocity Column Store, for example – a large report that used to take 17 minutes to render now takes only three seconds.

Given the benefits, they had seen with in-memory technologies, they were keen to trial the technology preview of "Hekaton". Prior to using "Hekaton", their online gaming systems were handling about 15,000 requests per second, a huge number for most companies. However, bwin needed to be agile and stay at ahead of the competition and so they wanted access to the latest technology speed.

Using "Hekaton" bwin were hoping they could at least double the number of transactions. They were 'pretty amazed' to see that the fastest tests so far have scaled to 250,000 transactions per second.

So how fast is "Hekaton" – just ask Rick Kutschera, the Database Engineering Manager at bwin – in his words it's 'Wicked Fast'! However, this is not the only point that Rick highlights, he goes on to mention that "Hekaton" integrates seamlessly into the SQL Server engine, so if you know SQL Server, you know "Hekaton".

-- David Hobbs-Mallyon, Senior Product Marketing Manager

Microsoft SQL Server
SQLServer

SQLServer RT @Technitrain: #blogged Finding Shakespeare's Favourite Words With #DataExplorer spr.ly/6017nlnJ #en #sqlserver
3 hours ago · reply · retweet · favorite

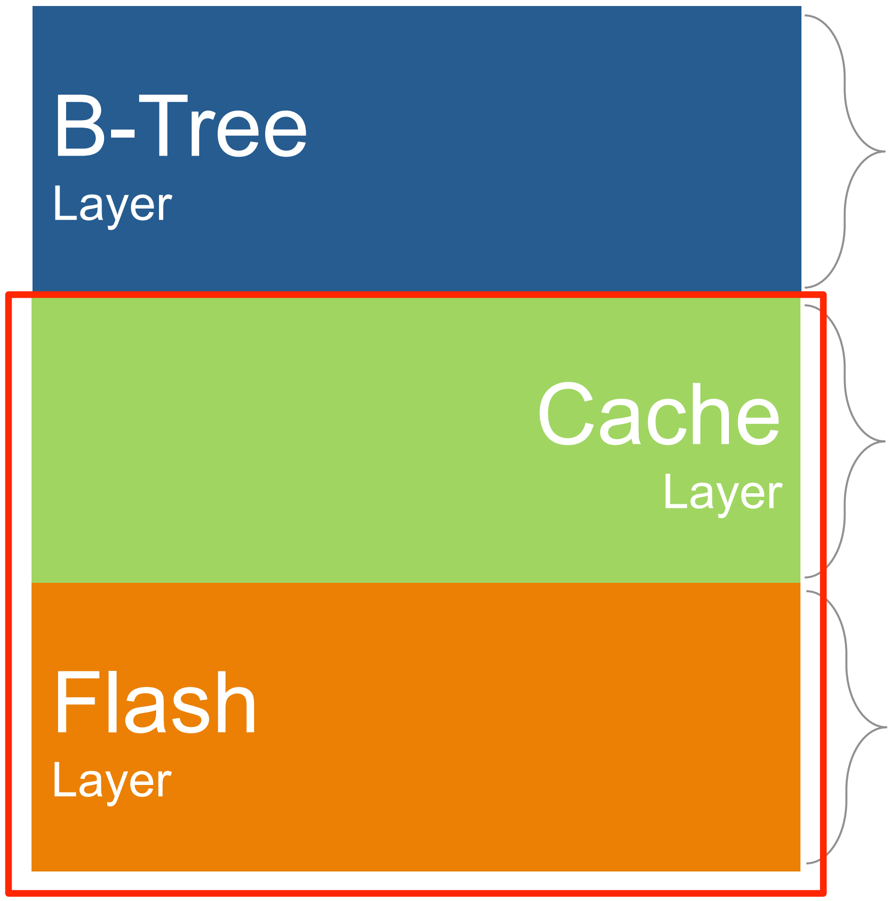
SQLServer RT @KarlaKay22: Have a great day of #sqlserver learning tomorrow #sqlsatPortugal and #sqlsatDetroit #sqlsaturday #sqlpass
18 hours ago · reply · retweet · favorite

Join the conversation

Navistar Uses Microsoft Big Data Solution to Determine Optimal Pricing

http://research.microsoft.com/main-memory_dbs/

Architecture

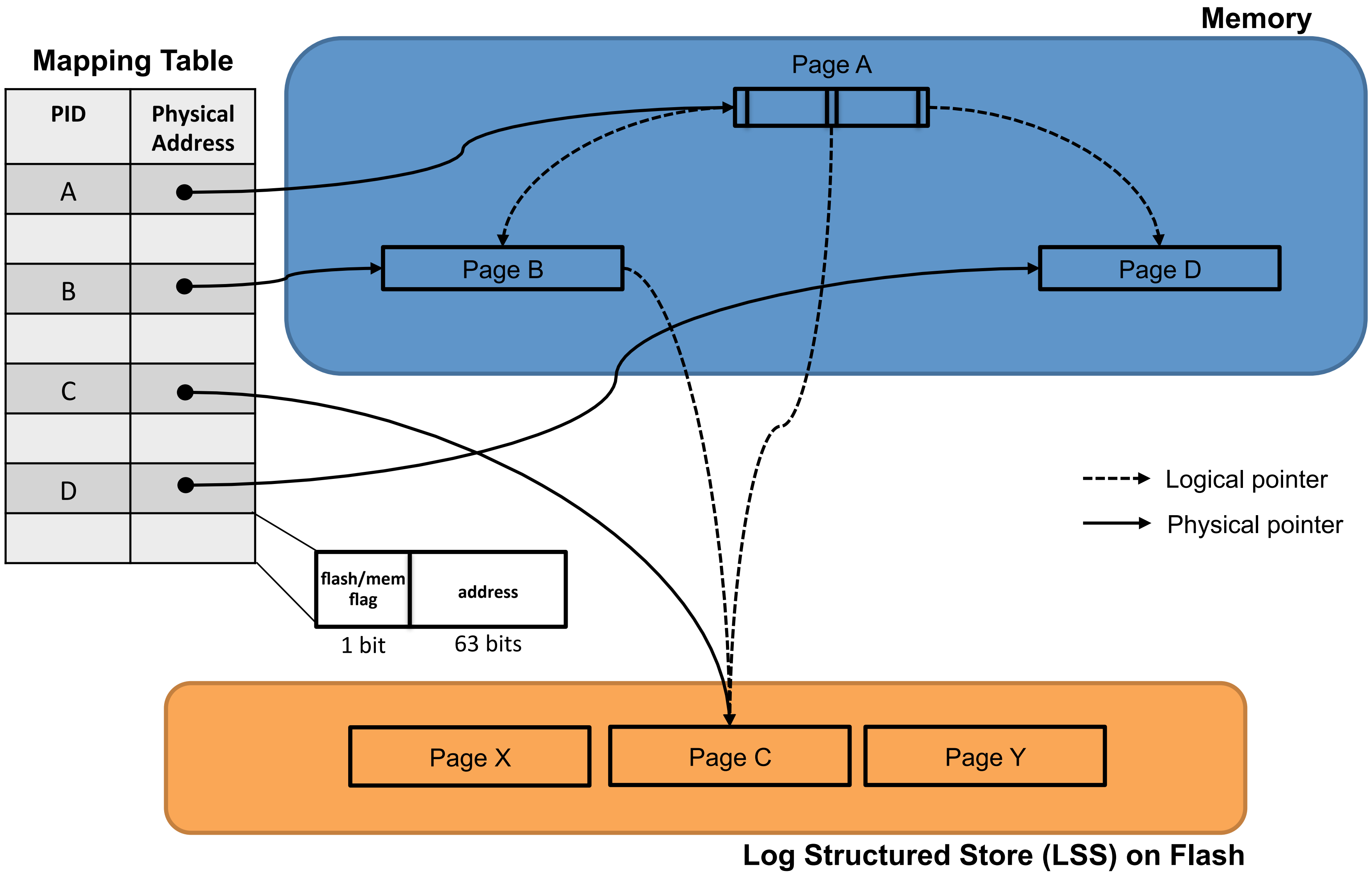


- API
- B-tree search/update logic
- In-memory pages only

- Logical page abstraction for B-tree layer
- Brings pages from flash to RAM as necessary

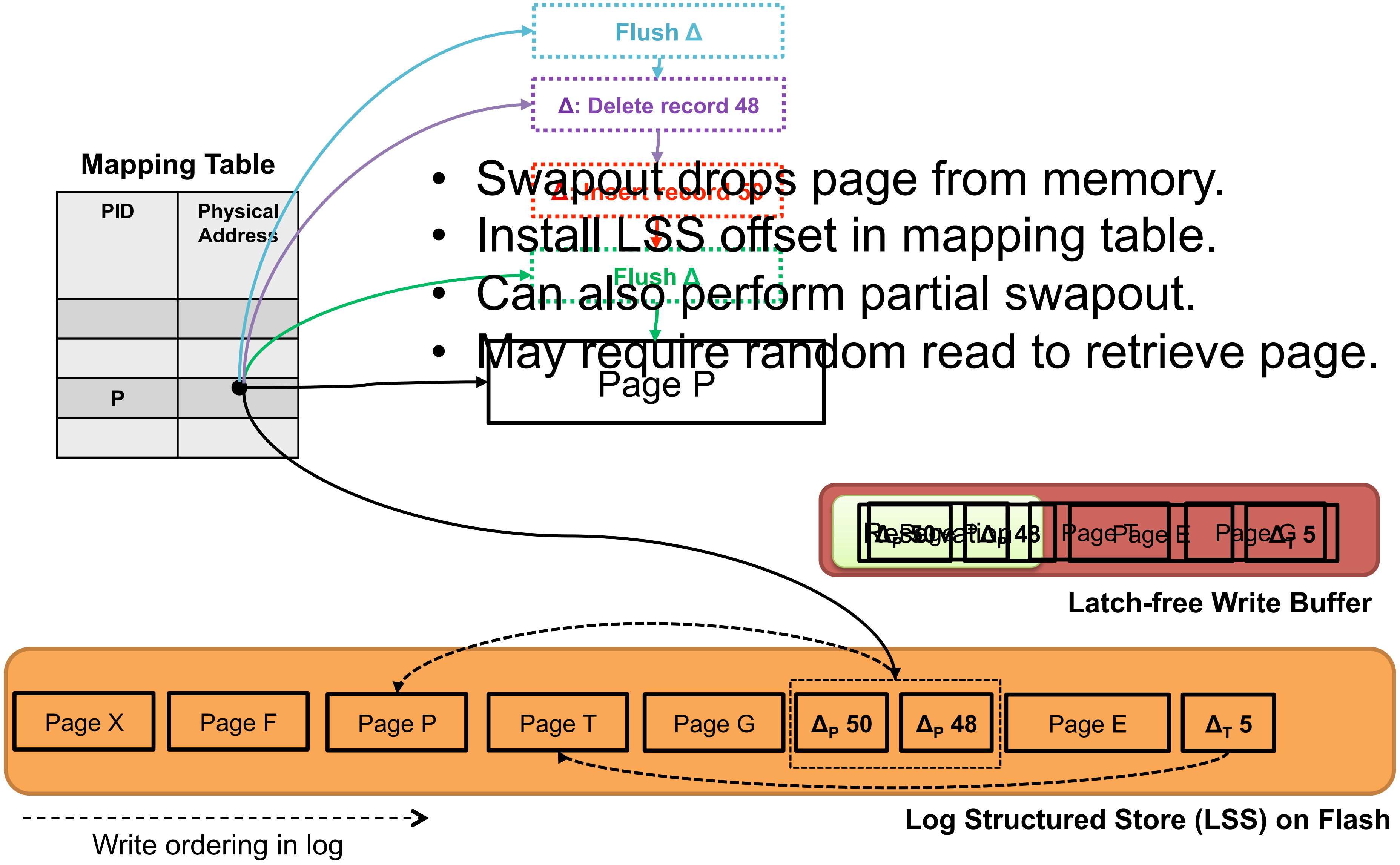
- Sequential writes to log-structured storage
- Flash garbage collection

Handling pages located on flash



Flushing pages

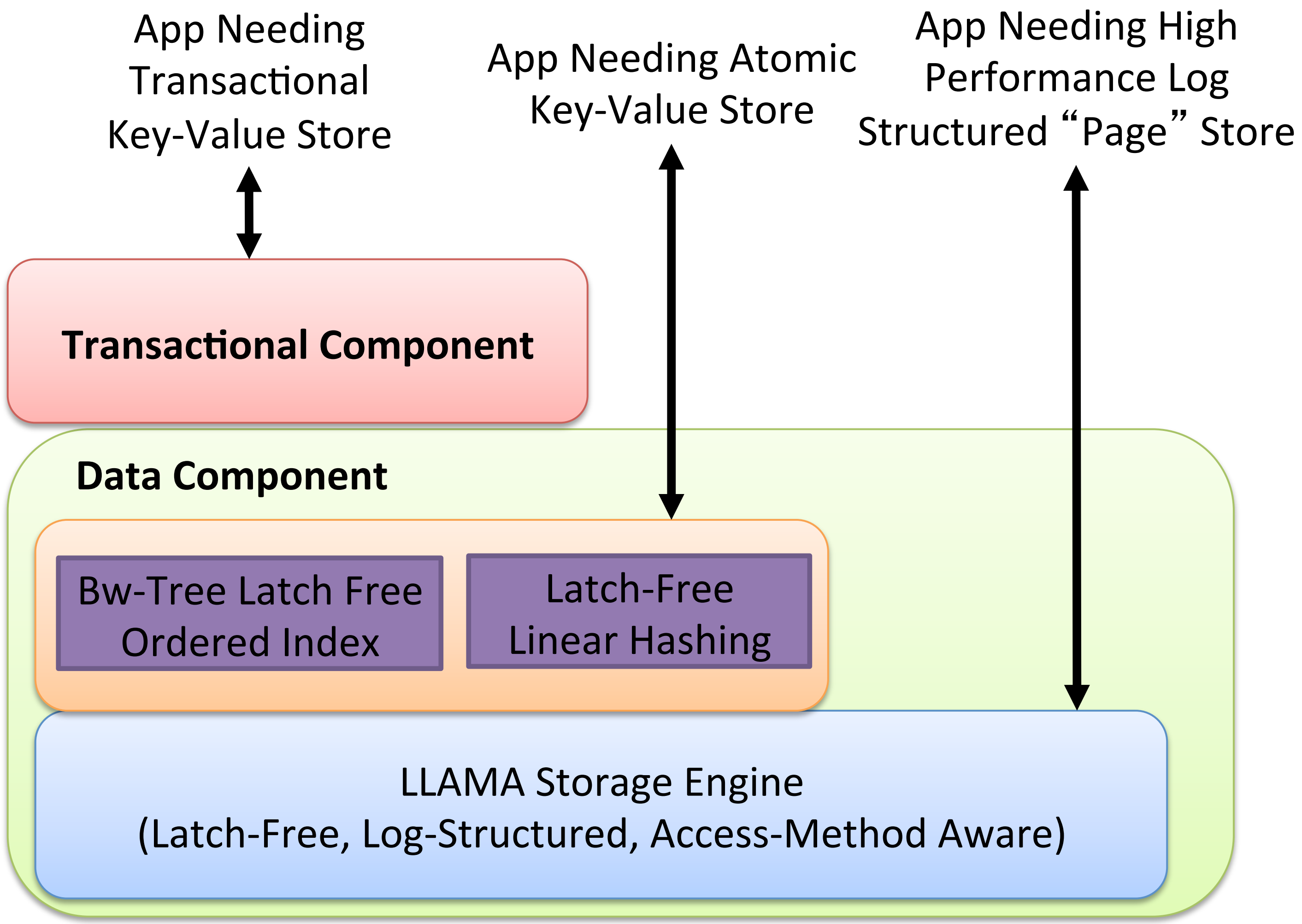
- Swapout drops page from memory.
- Install LSS offset in mapping table.
- Can also perform partial swapout.
- May require random read to retrieve page.



Other items

- LSS Garbage Collection
  - Cleans orphaned data unreachable from mapping table.
  - Relocates entire pages in sequential blocks (to reduce random reads from LSS).
- Access Method Recovery
  - Occasionally checkpoint mapping table.
  - Recover by:
    - Restoring mapping table.
    - Scan LSS forward from position recorded in checkpoint to the end of the log.
    - End result is latest LSS offset for pages in mapping table.

The Big Picture


App Needing Transactional Key-Value Store
App Needing Atomic Key-Value Store
App Needing High Performance Log Structured "Page" Store
Transactional Component
Data Component
Bw-Tree Latch Free Ordered Index
Latch-Free Linear Hashing
LLAMA Storage Engine
(Latch-Free, Log-Structured, Access-Method Aware)