

In-Memory Accelerator for Hadoop™



www.gridgain.com

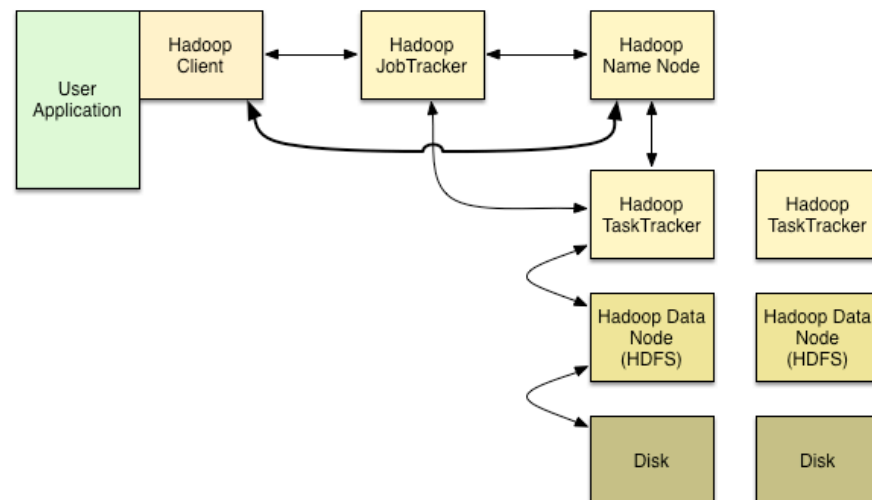


#gridgain

Hadoop: Pros and Cons

What is Hadoop?

- > Hadoop is a batch system
- > HDFS - Hadoop Distributed File System
- > Data must be ETL-ed into HDFS
- > Parallel processing over data in HDFS
- > Hive, Pig, HBase, Mahout...
- > Most popular data warehouse



Pros:

- > Scales very well
- > Fault tolerant and resilient
- > Very active and rich eco-system
- > Process TBs/PBs in parallel fashion

Cons:

- > Batch oriented - **real time not possible**
- > Complex deployment
- > Significant execution overhead
- > HDFS is IO and network bound

In-Memory Accelerator For Hadoop: Overview

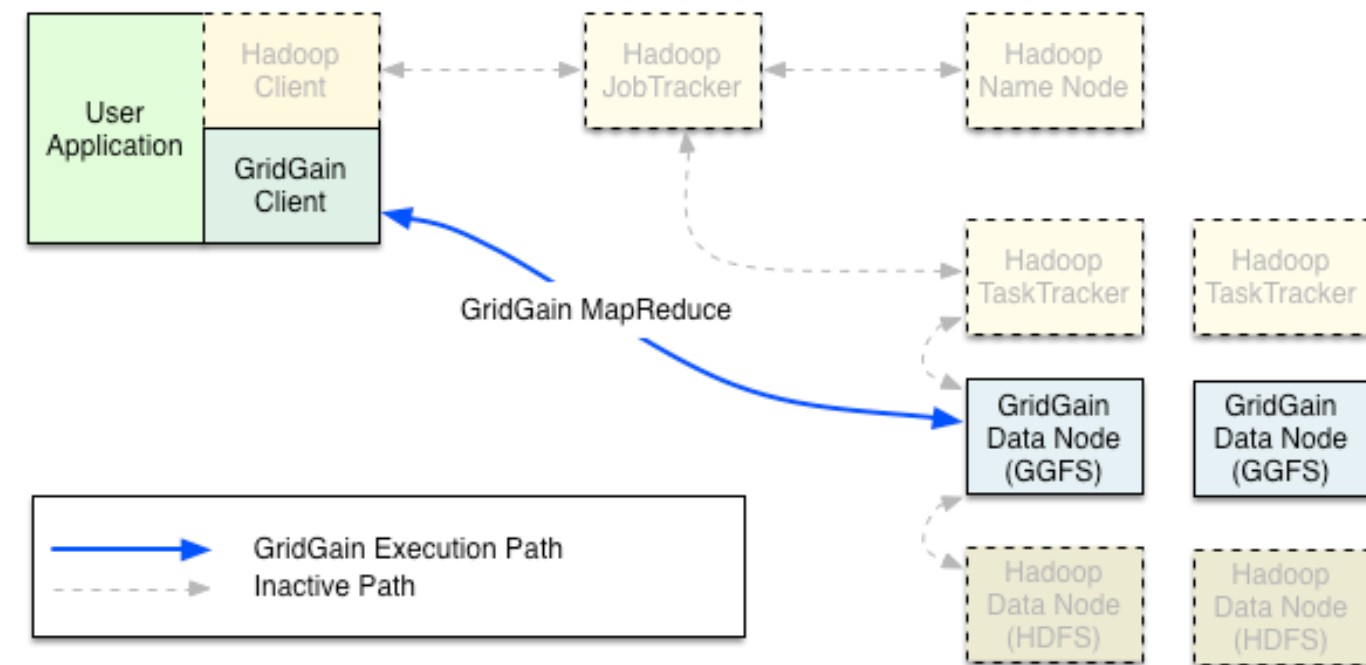
Up To 100x Faster:

1. In-Memory File System

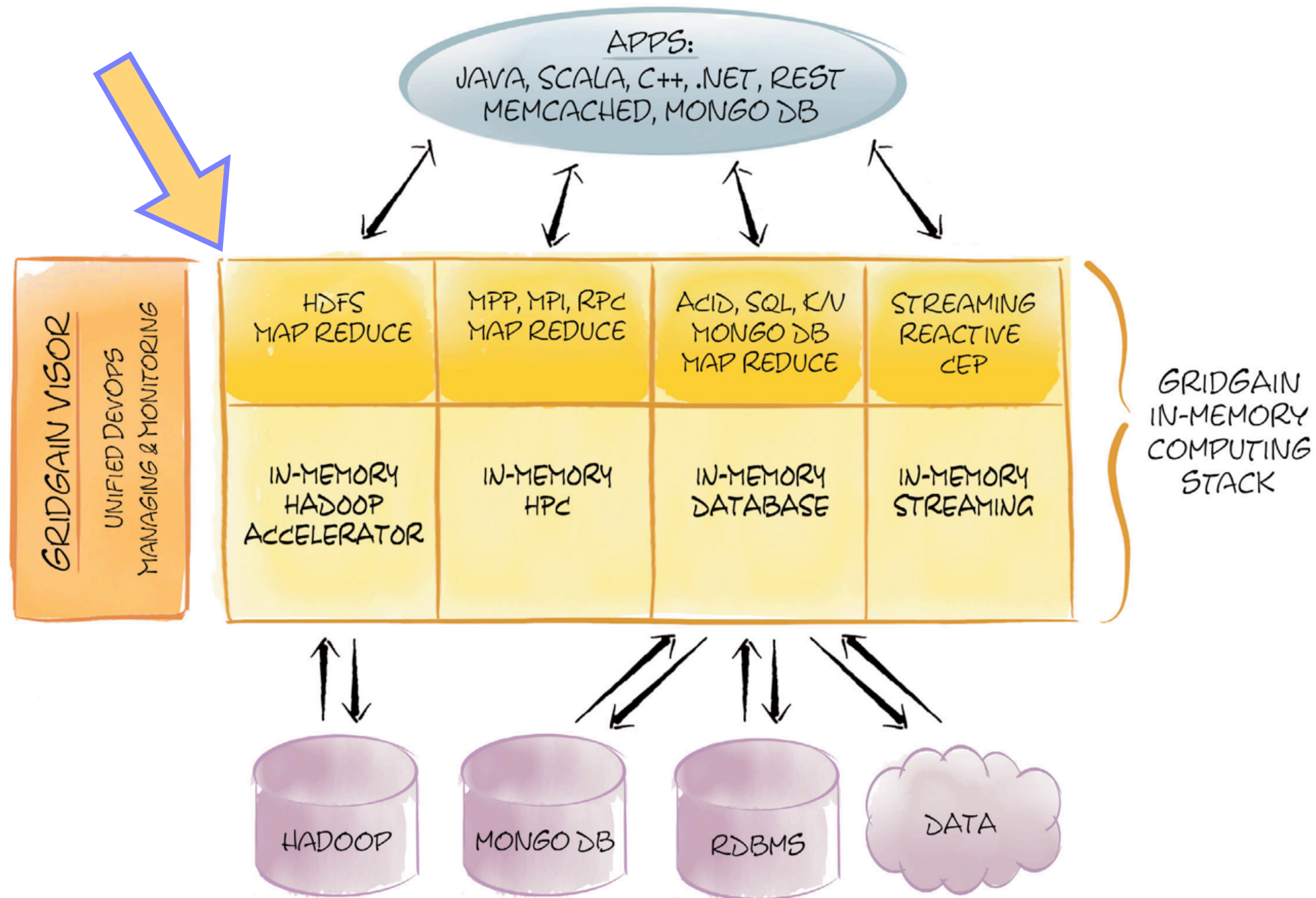
- 100% compatible with HDFS
- Boost HDFS performance by removing IO overhead
- Dual-mode: standalone or caching
- Blend into Hadoop ecosystem

2. In-Memory MapReduce

- Eliminate Hadoop MapReduce overhead
- Allow for embedded execution
- Record-based

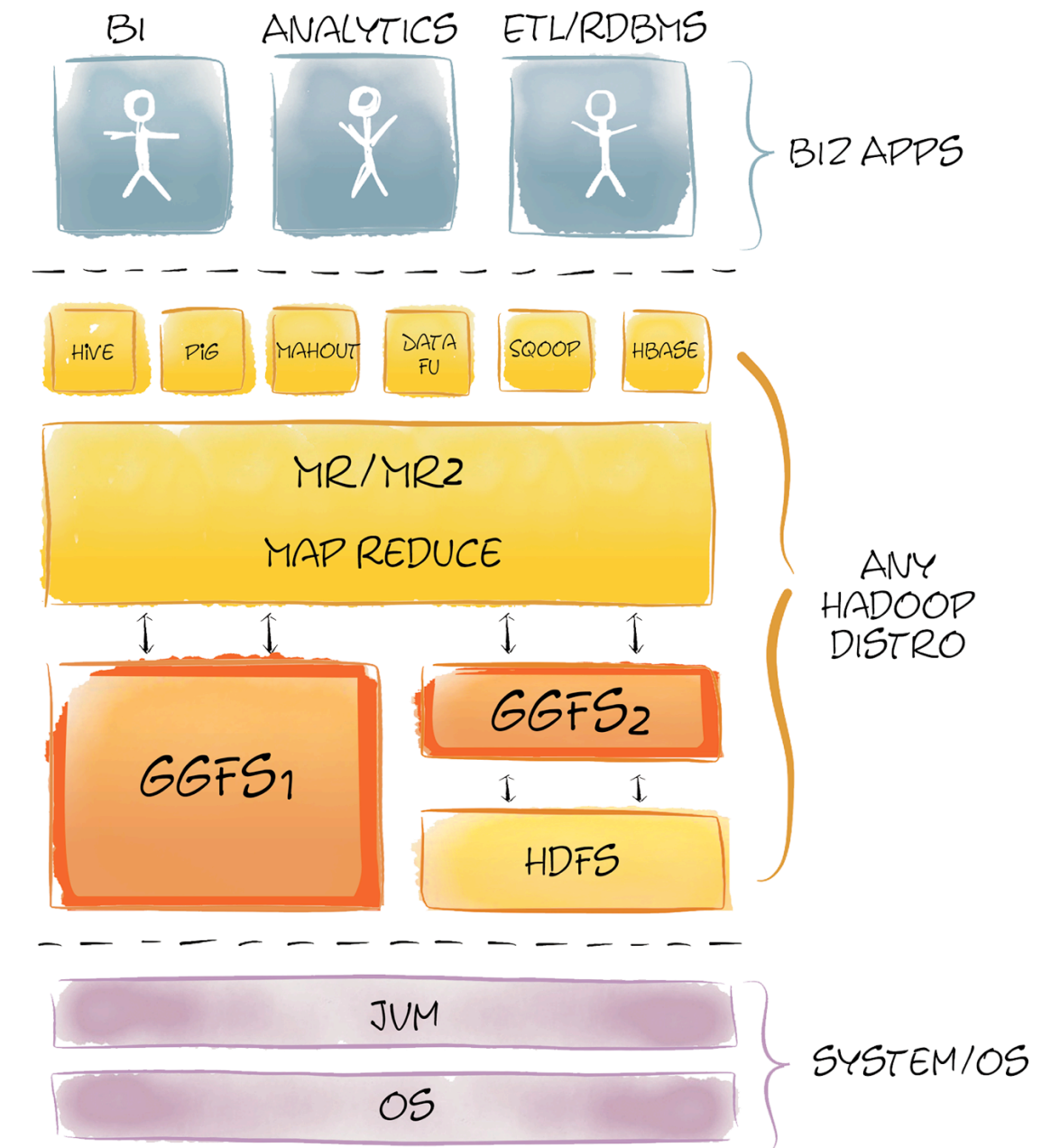


GridGain: In-Memory Computing Platform

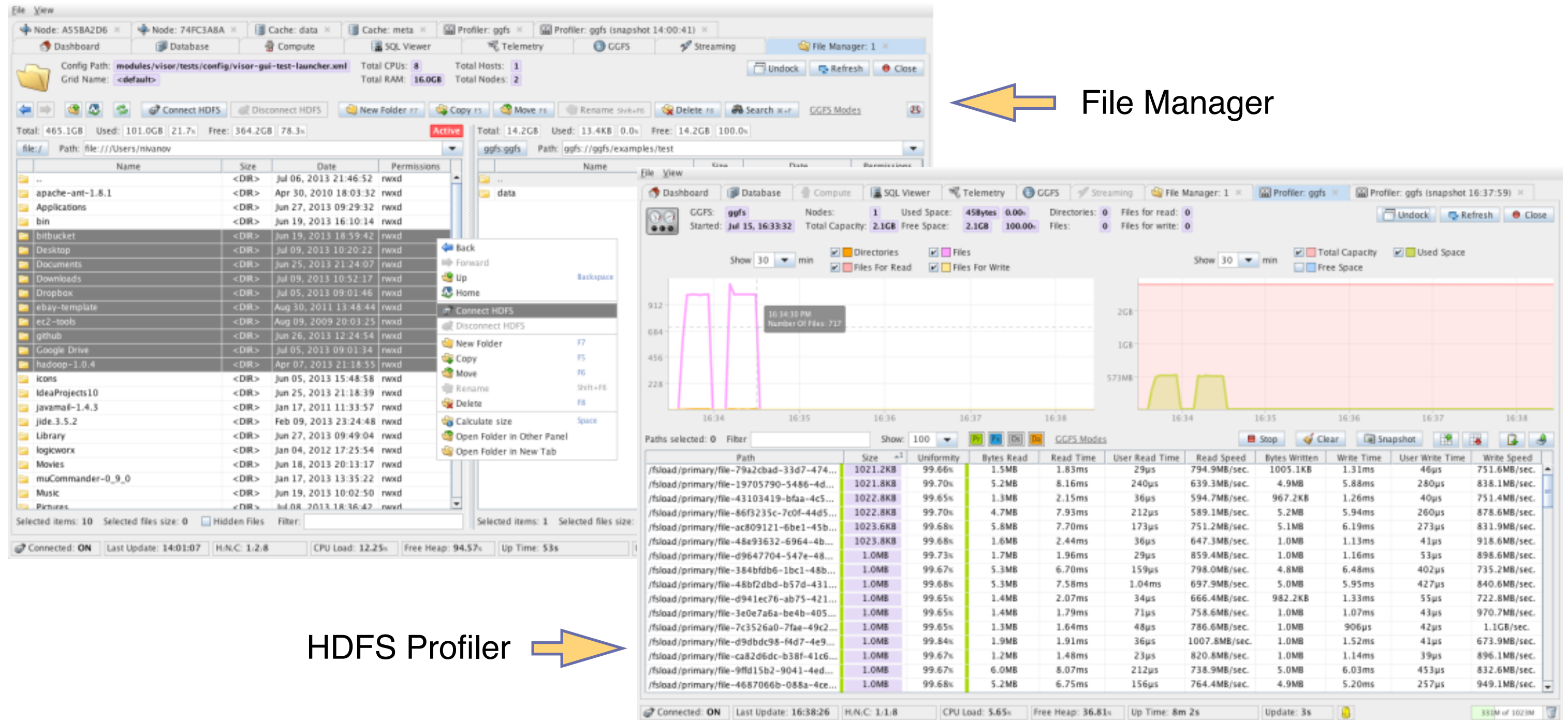


In-Memory Accelerator For Hadoop: Details

- > **PnP Integration**
Minimal or zero code change
- > **Any Hadoop distro**
Hadoop v1 and v2
- > **In-Memory File System**
100% compatible with HDFS
Dual-mode: no ETL needed, read/write-through
Block-level caching & smart eviction
Automatic pre-fetching
Background fragmentizer
On-heap and off-heap memory utilization
- > **In-Memory MapReduce**
In-process co-located computations - access GGFS in-process
Eliminate unnecessary IPC
Eliminate long task startup time
Eliminate mandatory sorting and re-shuffling on reduction



GridGain Visor: Unified DevOps



The image displays two screenshots of the GridGain Visor interface. The top screenshot shows the File Manager, which lists files and directories in a table with columns for Name, Size, Date, and Permissions. A context menu is open over the 'bin' directory, showing options like Back, Forward, Up, Home, Connect HDFS, Disconnect HDFS, New Folder, Copy, Move, Rename, Delete, Calculate size, Open Folder in Other Panel, and Open Folder in New Tab. The bottom screenshot shows the HDFS Profiler, which displays a graph of file activity over time and a table of file statistics. A yellow arrow points from the File Manager to the HDFS Profiler.

File Manager

HDFS Profiler

Path	Size	Uniformity	Bytes Read	Read Time	User Read Time	Read Speed	Bytes Written	Write Time	User Write Time	Write Speed
/fsload/primary/file-79a2cbad-33d7-474...	1021.2KB	99.66%	1.5MB	1.83ms	29µs	794.9MB/sec.	1005.1KB	1.31ms	46µs	751.6MB/sec.
/fsload/primary/file-19705790-5486-4d...	1021.8KB	99.70%	5.2MB	8.16ms	240µs	639.3MB/sec.	4.9MB	5.88ms	280µs	838.1MB/sec.
/fsload/primary/file-43103419-bfaa-4c5...	1022.8KB	99.65%	1.3MB	2.15ms	36µs	594.7MB/sec.	967.2KB	1.26ms	40µs	751.4MB/sec.
/fsload/primary/file-86f3235c-7c0f-44d5...	1022.8KB	99.70%	4.7MB	7.93ms	212µs	589.1MB/sec.	5.2MB	5.94ms	260µs	878.6MB/sec.
/fsload/primary/file-ac809121-6be1-45b...	1023.6KB	99.68%	5.8MB	7.70ms	173µs	751.2MB/sec.	5.1MB	6.19ms	273µs	831.9MB/sec.
/fsload/primary/file-48e93632-6964-4b...	1023.8KB	99.68%	1.6MB	2.44ms	36µs	647.3MB/sec.	1.0MB	1.13ms	41µs	918.6MB/sec.
/fsload/primary/file-d9647704-547e-48...	1.0MB	99.73%	1.7MB	1.96ms	29µs	859.4MB/sec.	1.0MB	1.16ms	53µs	898.6MB/sec.
/fsload/primary/file-384bfb6-1bc1-48b...	1.0MB	99.67%	5.3MB	6.70ms	159µs	798.0MB/sec.	4.8MB	6.48ms	402µs	735.2MB/sec.
/fsload/primary/file-48bf2dbd-b57d-431...	1.0MB	99.68%	5.3MB	7.58ms	1.04ms	697.9MB/sec.	5.0MB	5.95ms	427µs	840.6MB/sec.
/fsload/primary/file-d941ec76-ab75-421...	1.0MB	99.65%	1.4MB	2.07ms	34µs	666.4MB/sec.	982.2KB	1.33ms	55µs	722.8MB/sec.
/fsload/primary/file-3e0e7a6a-be4b-405...	1.0MB	99.65%	1.4MB	1.79ms	71µs	758.6MB/sec.	1.0MB	1.07ms	43µs	970.7MB/sec.
/fsload/primary/file-7c3526a0-7fae-49c2...	1.0MB	99.65%	1.3MB	1.64ms	48µs	786.6MB/sec.	1.0MB	906µs	42µs	1.1GB/sec.
/fsload/primary/file-d9dbdc98-f4d7-4e9...	1.0MB	99.84%	1.9MB	1.91ms	36µs	1007.8MB/sec.	1.0MB	1.52ms	41µs	673.9MB/sec.
/fsload/primary/file-ca82d6dc-b38f-41c6...	1.0MB	99.67%	1.2MB	1.48ms	23µs	820.8MB/sec.	1.0MB	1.14ms	39µs	896.1MB/sec.
/fsload/primary/file-9ffd15b2-9041-4ed...	1.0MB	99.67%	6.0MB	8.07ms	212µs	738.9MB/sec.	5.0MB	6.03ms	453µs	832.6MB/sec.
/fsload/primary/file-4687066b-088a-4ce...	1.0MB	99.68%	5.2MB	6.75ms	156µs	764.4MB/sec.	4.9MB	5.20ms	257µs	949.1MB/sec.

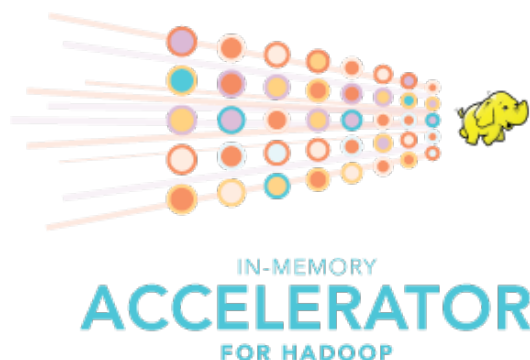
Benchmarks: GGFS vs HDFS

BENCHMARK	GGFS, MS.	HDFS, MS.	BOOST, %
File Scan	27	667	2470%
File Create	96	961	1001%
File Random Access	413	2931	710%
File Delete	185	1234	667%

10 nodes cluster of Dell R610

- > Each has dual 8-core CPU
- > Ubuntu 12.4, Java 7
- > 10 GBE network
- > Stock Apache Hadoop 2.x

Comparison: Hadoop Accelerator vs. Spark



- > **No ETL required**
Automatic HDFS read-through and write-through
Data is loaded on demand
- > **Per-block file caching**
Only hot data blocks are in memory
- > **Strong management capabilities**
GridGain Visor - Unified DevOps

- > **Requires data ETL-ed into Spark**
Changes to data do not get propagated to HDFS
Explicit ETL step consumes time
- > **Needs to have full file loaded**
If does not fit - gets offloaded to disk
- > **No management capabilities**

Customer Use Case: Task & Challenge

Task:

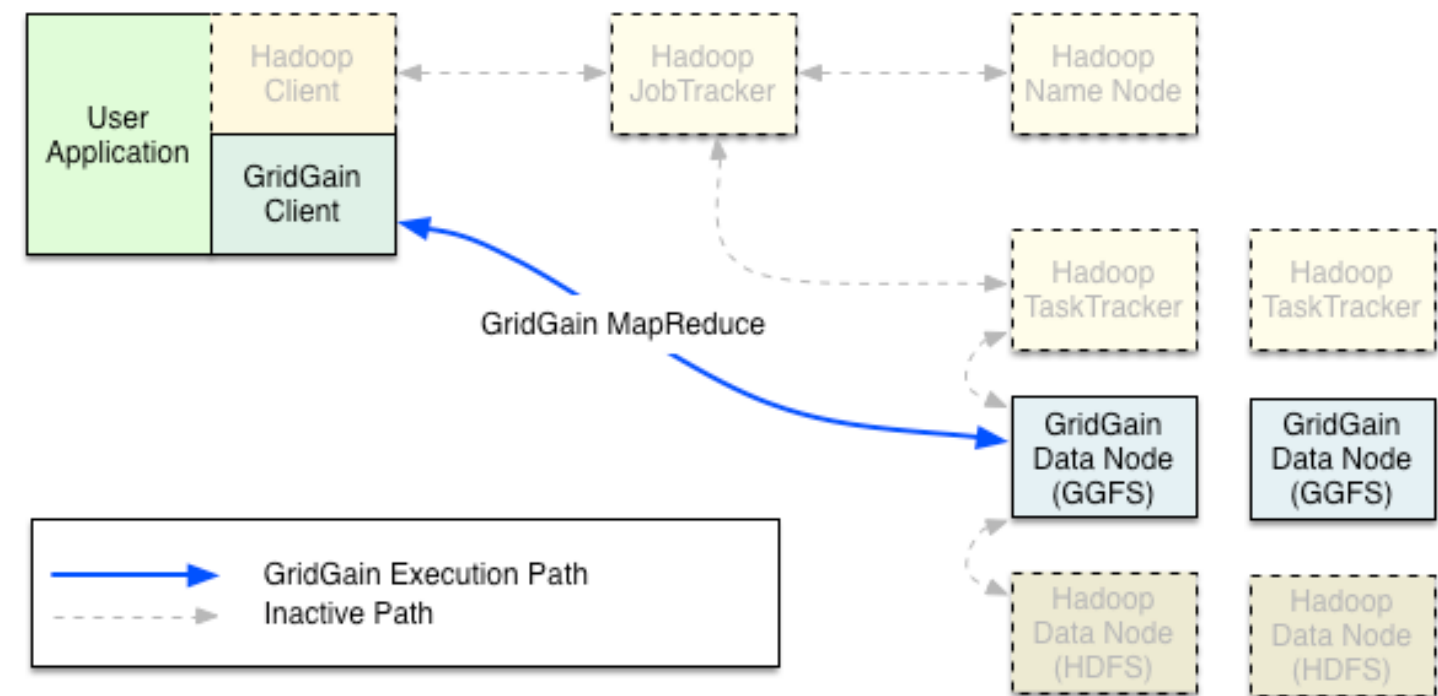
- > Real time search with MapReduce
- > Dataset size is 5TB
- > Writes 80%, reads 20%
- > Perceptual real time SLA (few seconds)

Challenge:

- > Hadoop MapReduce too slow (> 30 sec)
- > Data scanning slow due to constant IO
- > Overall job takes > 1 minute

Customer Use Case: **Solution**

- > **Utilize existing servers**
Start GridGain data node on every server
- > **Only put highly utilized files in GGFS**
User controlled caching
- > **In-Memory MapReduce over GGFS**
Embedded processing
- > **Results under 3 seconds**



GridGain

IN-MEMORY COMPUTING



@gridgain

GridGain Systems

www.gridgain.com

1065 East Hillsdale Blvd, Suite 230
Foster City, CA 94404