

Modern servers should run analytics (AP) & transactional (TP) workloads together.



Problematically, even in a many core server today ...

1. Heavy AP processing is not performance-optimal in isolation.
2. When mixed with TP, AP's blocking stream-based joins/aggs + thread intensive exec. monopolize CPU, Mem. BW etc., collapsing TP throughput leading to analysis paralysis.

PILLAR I Fast Analysis

✗ **Wasted work:** avoidable copies, redundant IO, duplicate intermediate computation

- 1 **Early bloom filters** built early + pushed to scans
- 2 **Late reads** carry row-IDs & keys + materialize later
- 3 **Direct CU access** skip intermediate copies, operators work directly over storage format
- 4 **Intermediate share** similar sub-plans reuse results

PILLAR II Fast Dominant Operators

✗ **Inefficient core operators:** hash joins & aggs are NOT hardware-aware NOT concurrency-optimal

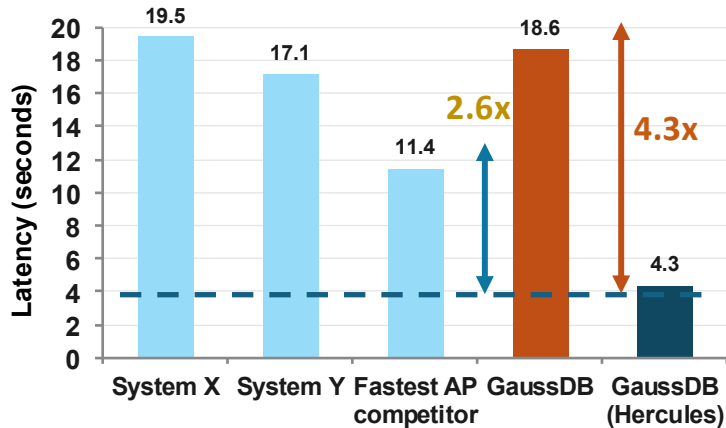
- 5 **CA hash joins** parallel build/probe, no costly streams specialized HW-aware HT (variant of DLHT [HPDC'24])
- 6 **CA hash aggregations** DLHT-specialized HT + no global shared state = eschew costly synch.
- 7 **Non-blocking streams** less stalls + fewer copies
- 8 **Partition-wise execution** skip shuffle + no sync. costs

PILLAR III No Paralysis

✗ **Oblivious scheduler:** OS threads used for concurrency are agnostic to DB context; poor locality, no TP priority

- 9 **Task-based execution** fine-grained, out-of-order
- 10 **Suspendable tasks** self-suspend when blocked
- 11 **User-level scheduling** TP gets priority
- 12 **NUMA & locality-aware** min. coherence + fast access

FAST ANALYSIS: TPC-H (SF100) completion latency

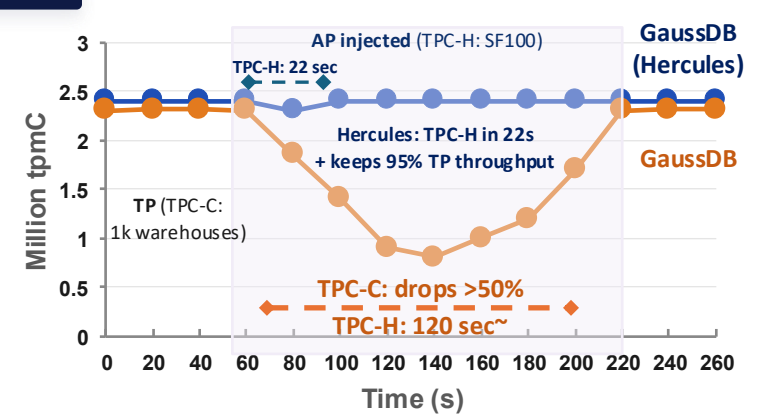


Hercules' 12 Labors Solve the 3 Pillars of Analysis Paralysis!



Results on a single Arm-based Huawei server (Kunpeng 920: 160 cores, 2TB DRAM)

NO PARALYSIS: Mixed TP-AP throughput



Fastest AP
On one Arm server

4.3 s
TPC-H (SF100)

>4x
vs. GaussDB

2.5x
vs. best competitor

22s TPC-H @ 2.4 tpmC
Record-class TP-AP

Record-class TP + AP
More at: bit.ly/hercdb