DAMON 2020

TECHNISCHE UNIVERSITÄT DARMSTADT

THE BEST RUN SAP

Hochschule Reutlingen
Reutlingen University

THE TALE OF 1000 CORES: AN EVALUATION OF CONCURRENCY CONTROL ON REAL(LY) LARGE MULTI-SOCKET HARDWARE

Tiemo Bang

TU Darmstadt & SAP SE

Norman May

SAP SE

Ilia Petrov

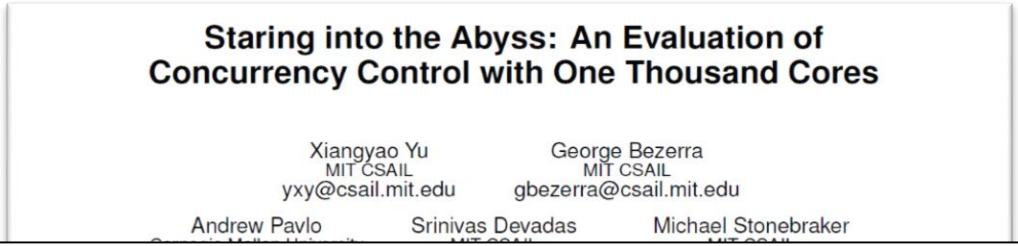
Reutlingen University

Carsten Binnig

TU Darmstadt

PRIOR WORK ON OLTP WITH 1000 CORES

VLDB 2014:



Staring into the Abyss: An Evaluation of Concurrency Control with One Thousand Cores

Xiangyao Yu
MIT CSAIL
yxy@csail.mit.edu

George Bezerra
MIT CSAIL
gbezerra@csail.mit.edu

Andrew Pavlo

Srinivas Devadas

Michael Stonebraker

[1]

The gist:

- Many-core era has begun: more processing power by more cores.
- DBMS will coordinate 1000s of concurrent transactions.

? **How does concurrency control perform for OLTP on many cores in future?**

➢ **No concurrency control scheme scales well on simulated 1000 cores!**

How is the situation for OLTP today, 6 years later?

[1] Yu et al. 2014. Staring into the Abyss: An Evaluation of Concurrency Control with One Thousand Cores. VLDB. https://doi.org/10.14778/2735508.2735511

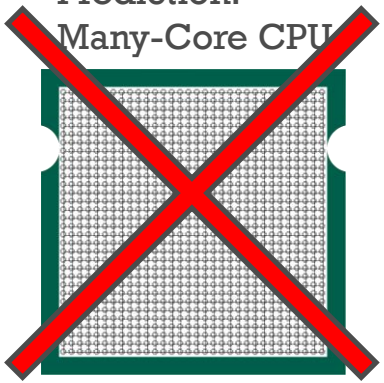
FUTURE IS NOW

Evaluated CC schemes are still common in today's DBMS:

- Deadlock Detect
- No Wait
- Wait Die
- H-STORE
- Multi-Versioned Concurrency Control
- Optimistic Concurrency Control
- Timestamp Ordering

Hardware with 1000 cores is in production and still growing:

Prediction:
Many-Core CPU



Today:
Many CPUs

Burts. 2018. HPE BOOTS UP SANDBOX OF THE MACHINE FOR EARLY USERS. https://www.nextplatform.com/2018/06/21/hpe-boots-up-sandbox-of-the-machine-for-early-users/

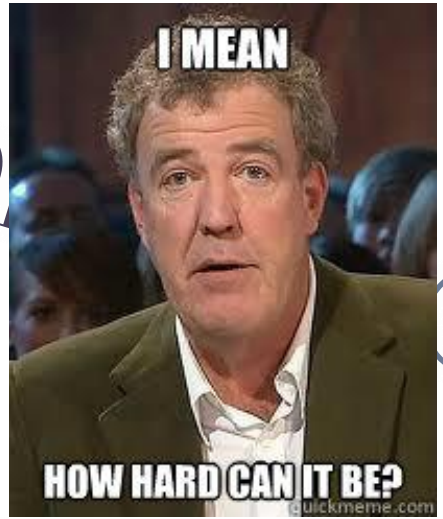
How does concurrency control scale on real 1000 cores today?

THE TALE ON 1000 CORES

How does concurrency control scale on real 1000 cores today?

Plan: Simply re-run original experiments

1. Real(ly) large multi-socket hardware
   (1568 logical cores on 28 Skylake CPUs)
2. Original open-source DBMS, DBx1000
3. TPC-C benchmark for high & low conflict OLTP



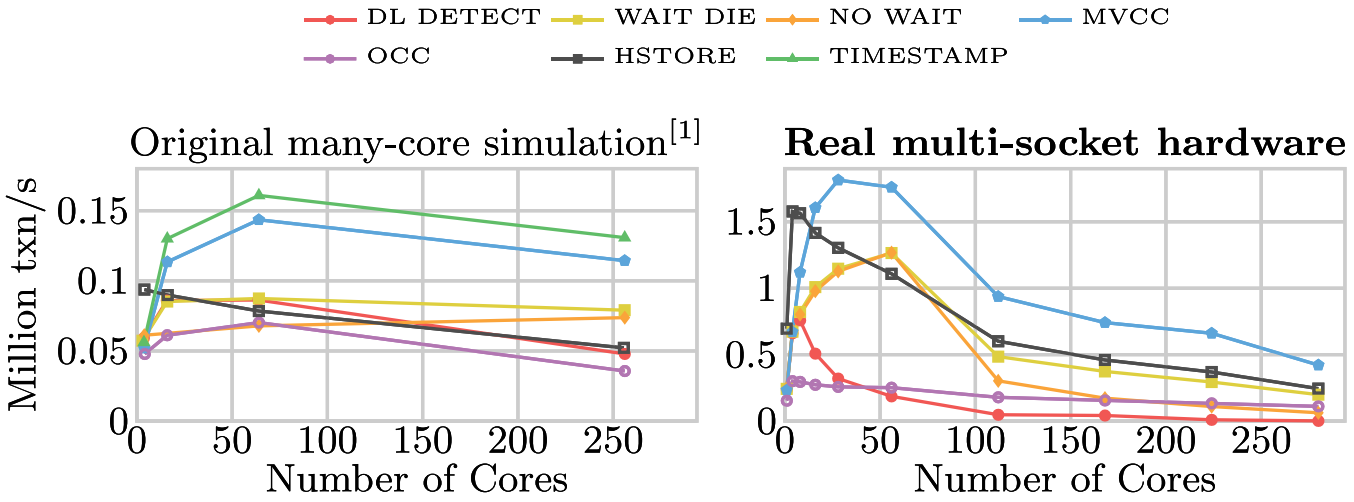
… a longer story is about to unfold
until a clear view on concurrency control with 1000 cores!

A First Look

SIMULATION VS. REALITY

SIMULATION VS. REALITY (1)

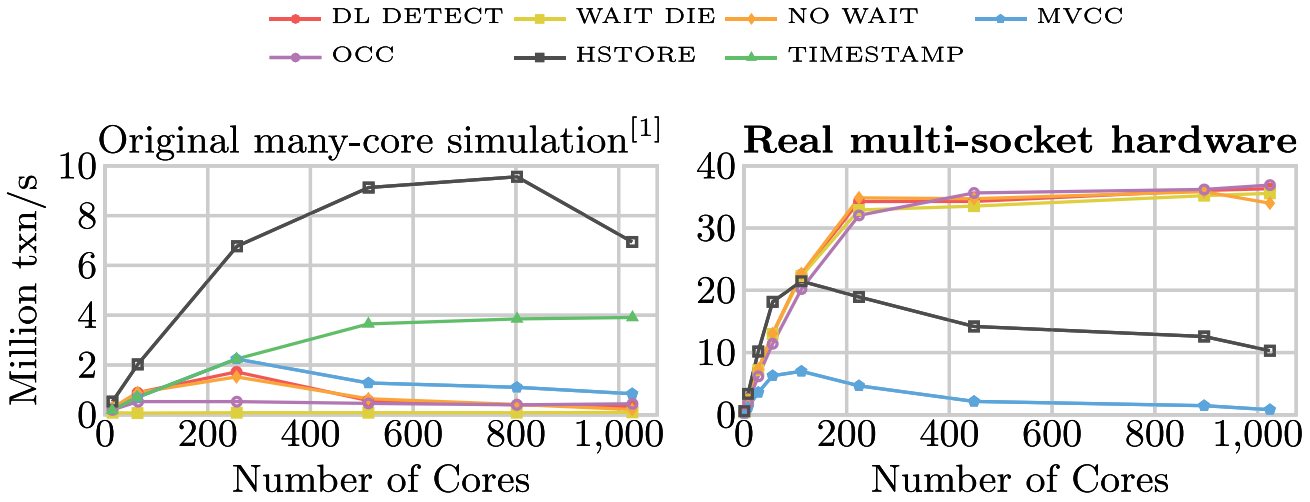
Throughput for **high conflict** TPC-C (4 WH)



➢ **Some similarities in simulation and real hardware**

SIMULATION VS. REALITY (2)

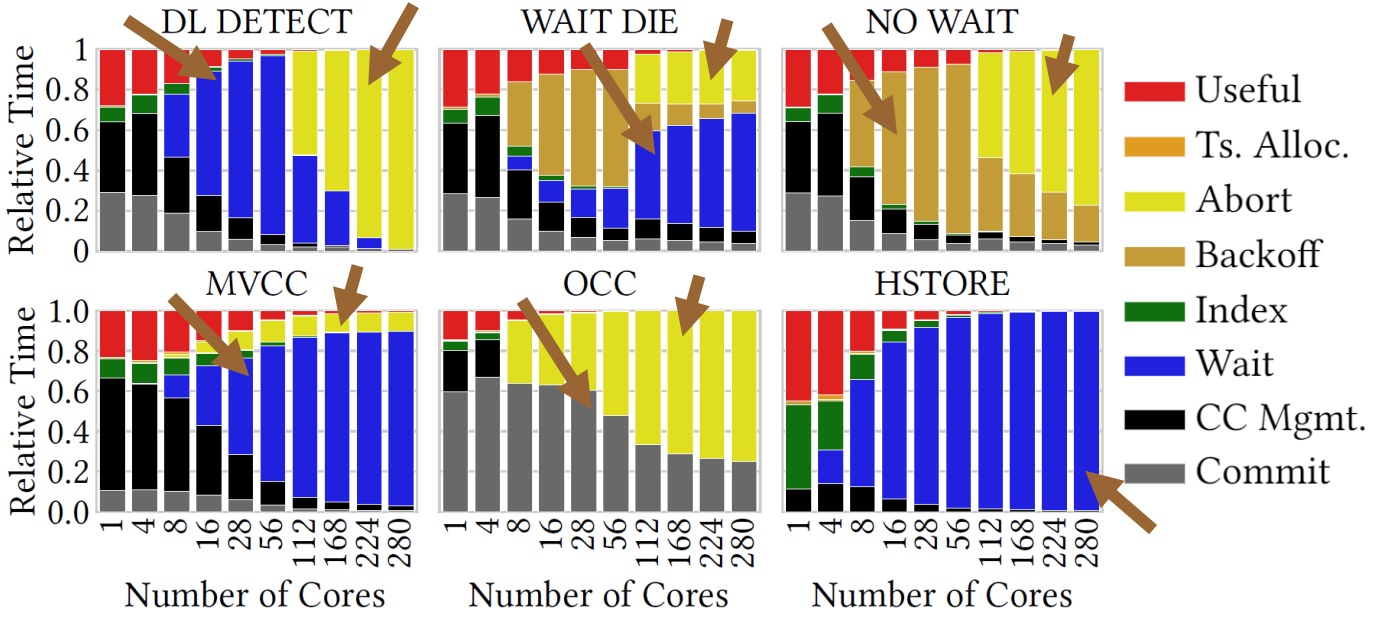
Throughput for **low conflict** TPC-C (1024 WH)



➢ **2PL schemes perform the best under low conflict?**

ANALYSIS ON REAL HARDWARE (1)

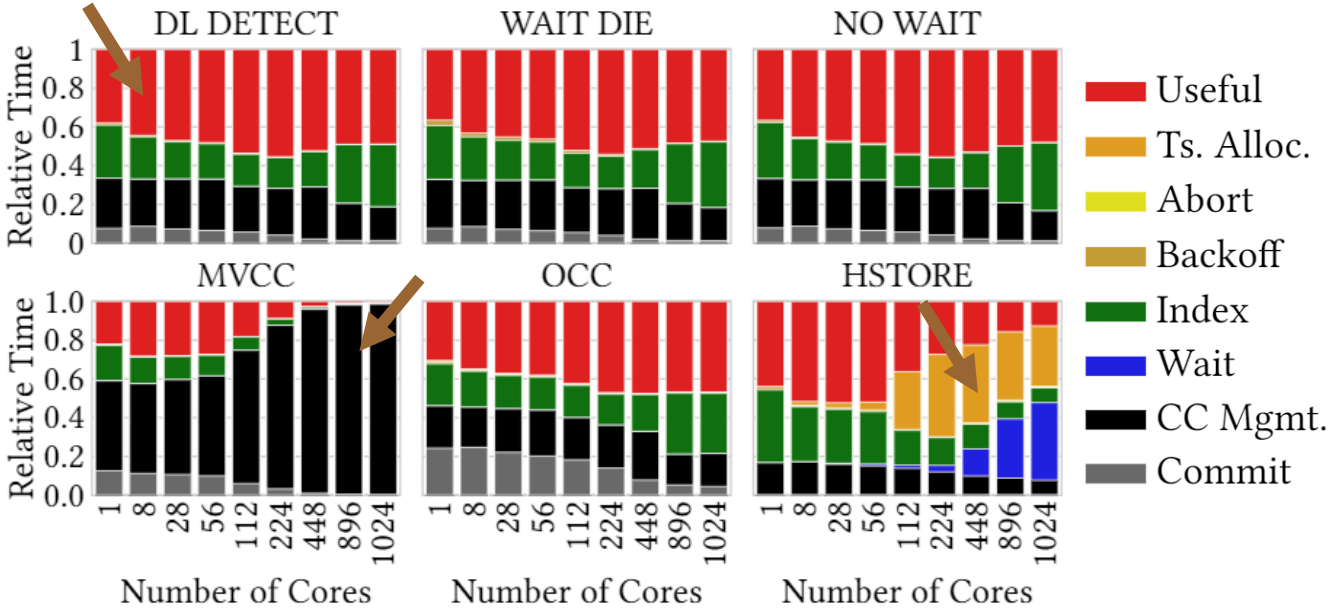
Time breakdown for high conflict TPC-C



✓ **Textbook behaviour of CC on real hardware under high conflict**

ANALYSIS ON REAL HARDWARE (2)

Time breakdown of low conflict TPC-C



Why is there so much overhead of CC despite low conflict?

A Second Look on Real Hardware

HIDDEN SECRETS

ASSUMPTIONS OF PRIOR WORK

Why is there so much overhead of CC despite low conflict?

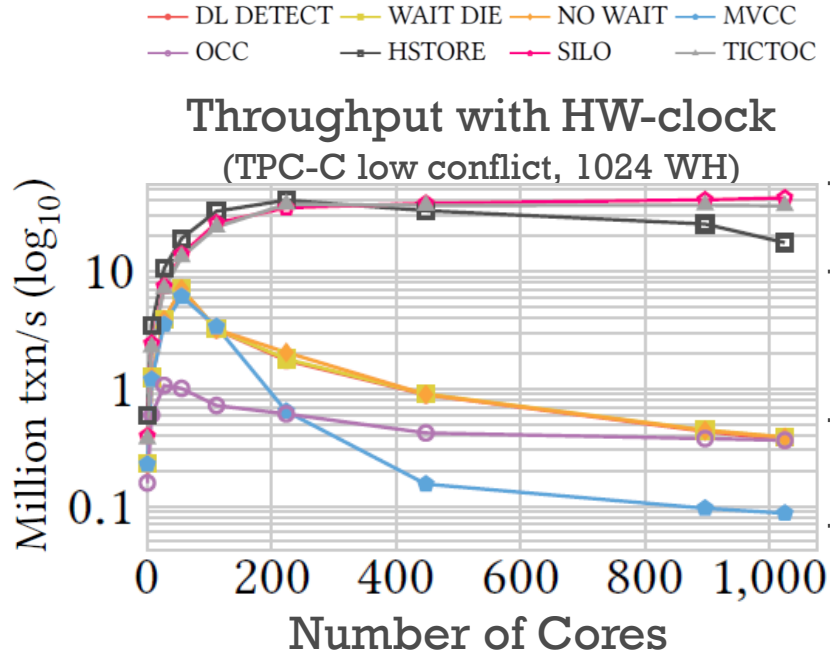
Yu et. al: Timestamp allocation, locking & latches are bottlenecks with many cores.

Their 1. assumption:

Timestamp allocation via hardware clock should scale better, but will not be available.

SCALABILITY VIA HARDWARE CLOCK ? (1)

Hardware clock is available today: *invariant tsc feature*



✓ Better performance:
No contention for timestamps

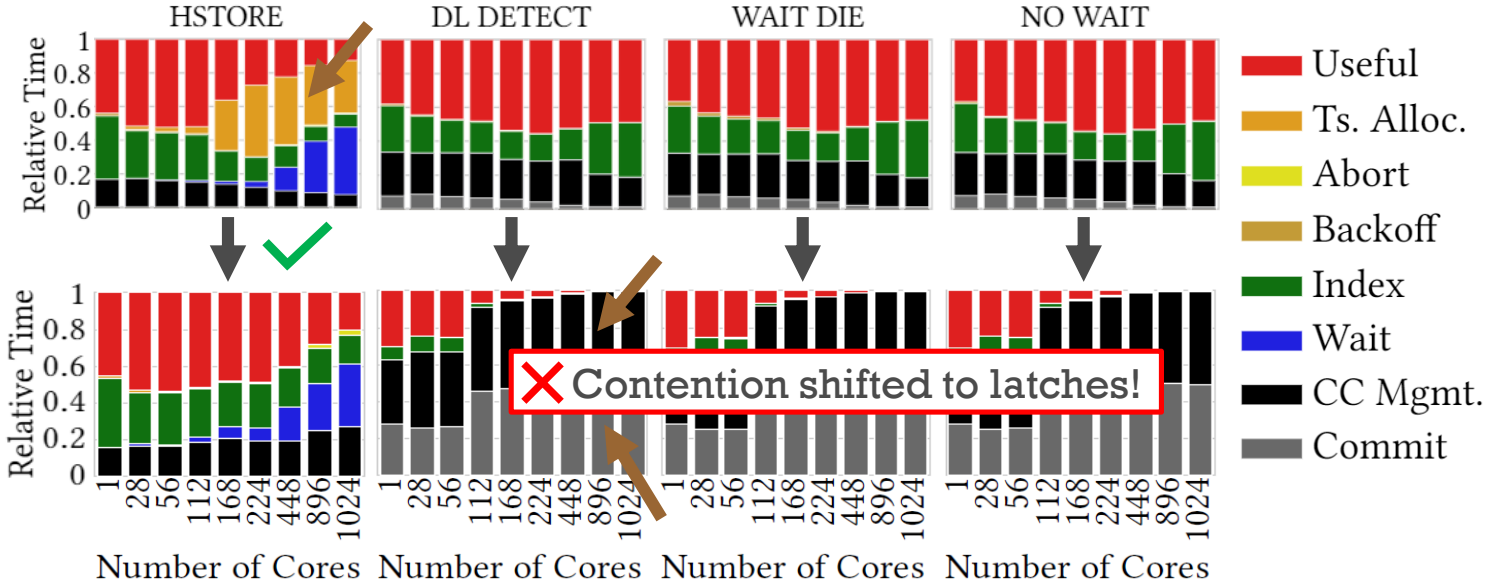
✗ Worse performance ?!

SCALABILITY VIA HARDWARE CLOCK ? (2)

Analysis of timestamp allocation via HW-clock (TPC-C low conflict, 1024 WH)

Before:
(Atomic increment)

After:
(HW-clock)



➢ **Contention pressures other components even more!**

ASSUMPTIONS OF PRIOR WORK CONT.

Why is there so much overhead of CC despite low conflict?

Yu et. al: ~~Timestamp allocation~~, locking & latches are bottlenecks for many cores under low conflict workload.

Their 1. assumption:

~~Timestamp allocation via hardware clock should scale better, but will not be available.~~

➢ **Insight for today's hardware:**
HW-clock may help scaling, but contention shifts

SOURCE OF CONTENTION

"Due to memory constraints [of the simulator] [...], we reduced the size of [the] database".

"This does not affect our measurements because the number of tuples accessed is independent."

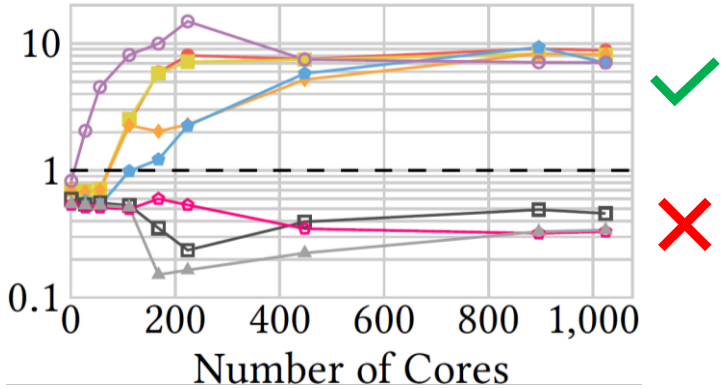
Yu et al., Staring into the Abyss: An Evaluation of Concurrency Control with One Thousand Cores, Section 5.6

Their 2. assumption:

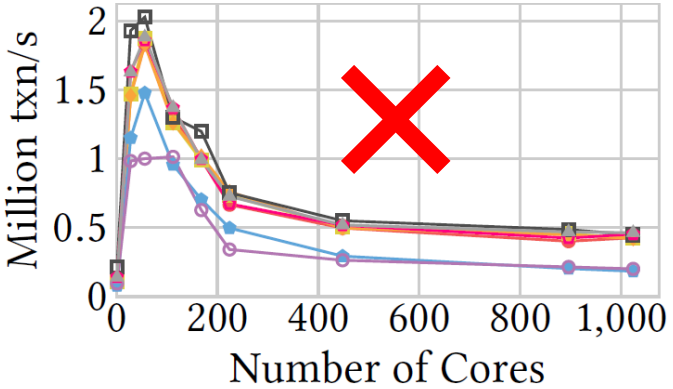
Smaller database & absent inserts do not affect CC.

EFFECT OF SIMULATION CONSTRAINTS

Speedup of complete database



Throughput with inserts



Ambivalent effect

× Slower data movement

✓ Relief physical contention

Orthogonal bottlenecks

× Catalogue lookups

× Memory management

OPTIMISATION: BACK TO THE FUTURE

State-of-the-art in-memory optimisations[2-4]:

- ✓ Hardware-assisted timestamp allocation
- ✓ Scalable synchronisation (Queuing latch & backoff)
- ✓ Efficient data movement (Reordering & prefetching)
- ✓ NUMA-awareness (Placement & replication)
- ✓ Query compilation (Meta data compiled)
- ✓ Efficient memory management (Thread local & pre-allocation)
- ✓ Exploit transaction context (No concurrency control for read-only relations)
- ✓ Less communication for deadlock (Wait-Die prevention, no active detection)

[2] Huang et al. 2020. Opportunities for Optimism in Contended Main-Memory Multicore Transactions. VLDB. https://doi.org/10.14778/3377369.3377373
[3] Kersten et al. 2018. Everything You Always Wanted to Know about Compiled and Vectorized Queries but Were Afraid to Ask. VLDB. https://doi.org/10.14778/3275366.3284966
[4] Kimura. 2015. FOEDUS: OLTP Engine for a Thousand Cores and NVRAM. SIGMOD. https://doi.org/10.1145/2723372.2746480

The Final Look on CC today

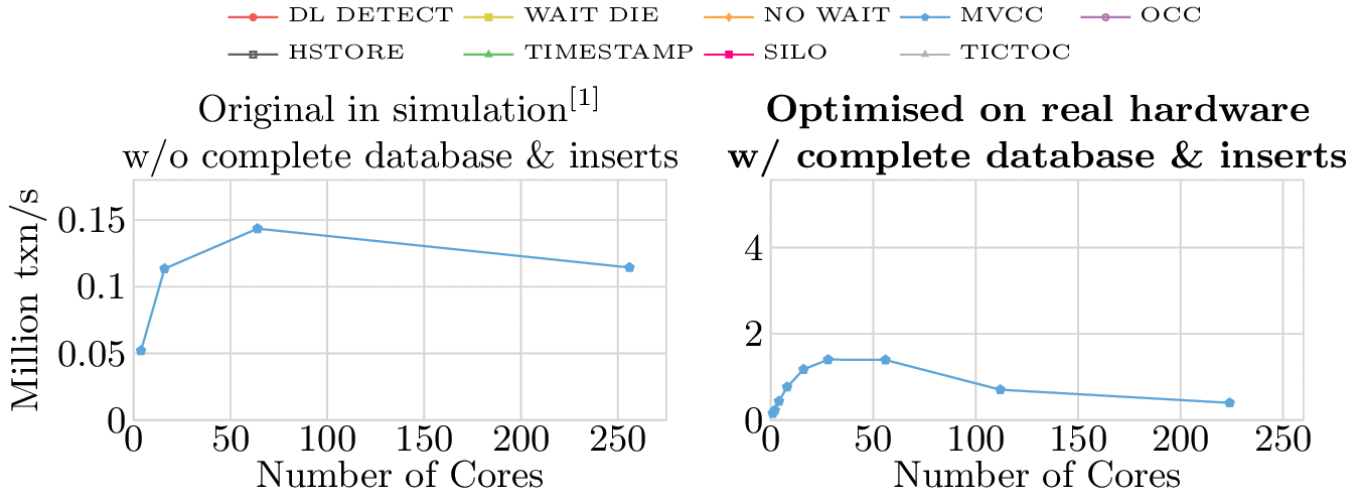
CLEARING SKIES

CLEAR VIEW OF CC ON 1000 CORES (1)

Throughput **of optimised DBx1000** for **high conflict TPC-C** (4 WH)

CLEAR VIEW OF CC ON 1000 CORES (1)

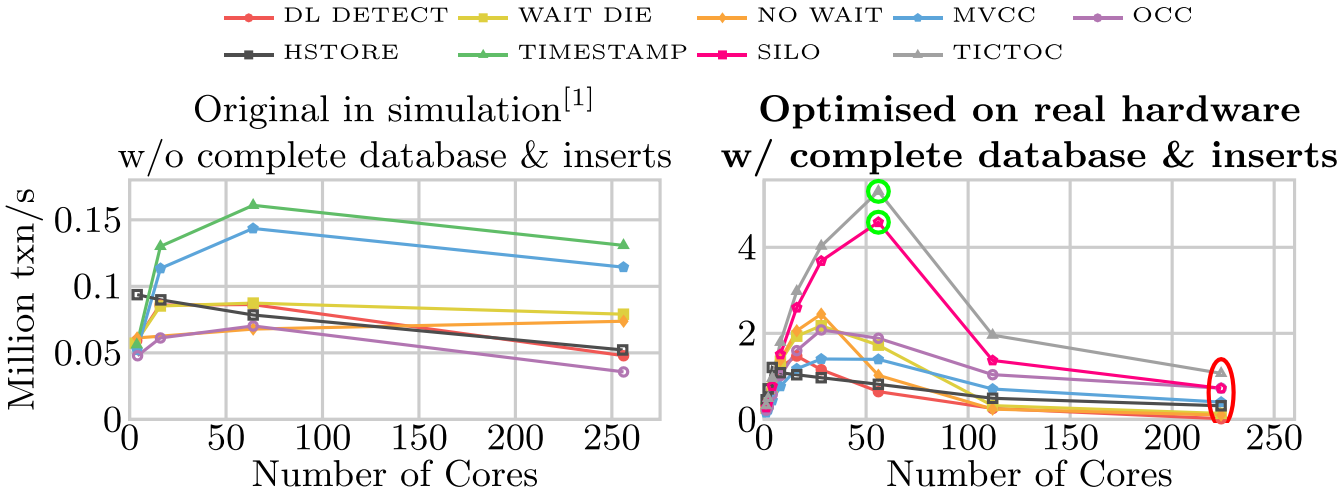
Throughput **of optimised DBx1000** for **high conflict TPC-C** (4 WH)



➢ **Fine-grained coordination may help, but cannot evade conflicts**

CLEAR VIEW OF CC ON 1000 CORES (1)

Throughput **of optimised DBx1000** for **high conflict TPC-C** (4 WH)



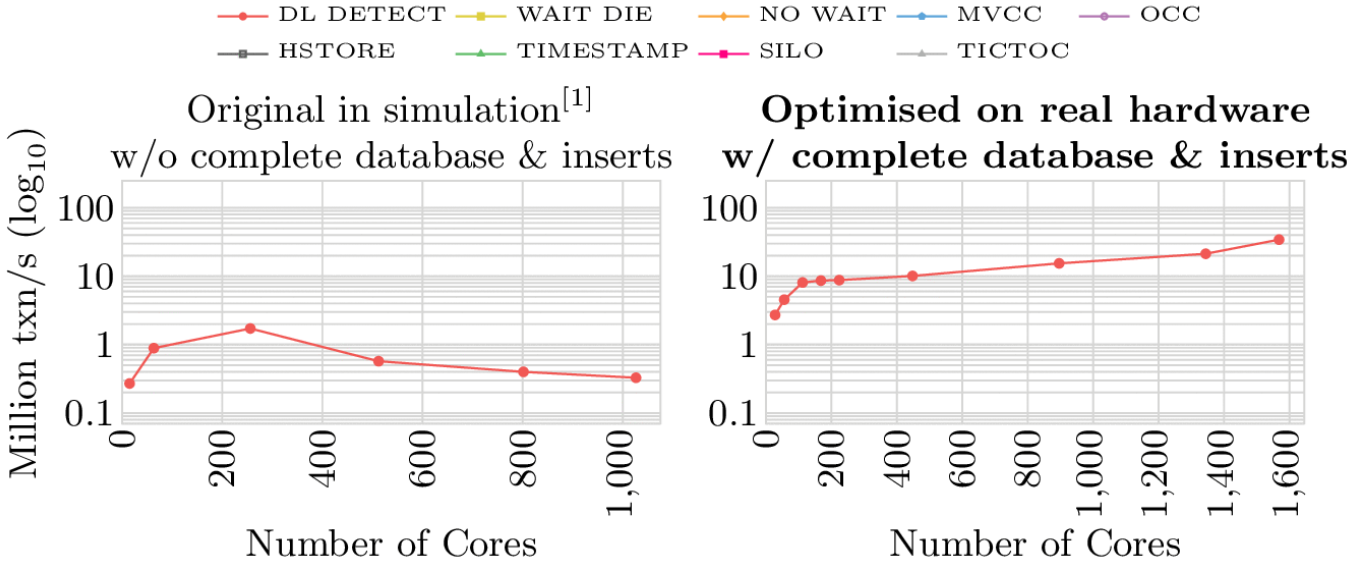
➢ **Fine-grained coordination may help, but cannot evade conflicts**

CLEAR VIEW OF CC ON 1000 CORES (2)

Throughput **of optimised DBx1000** for **low conflict TPC-C** (1024 WH)

CLEAR VIEW OF CC ON 1000 CORES (2)

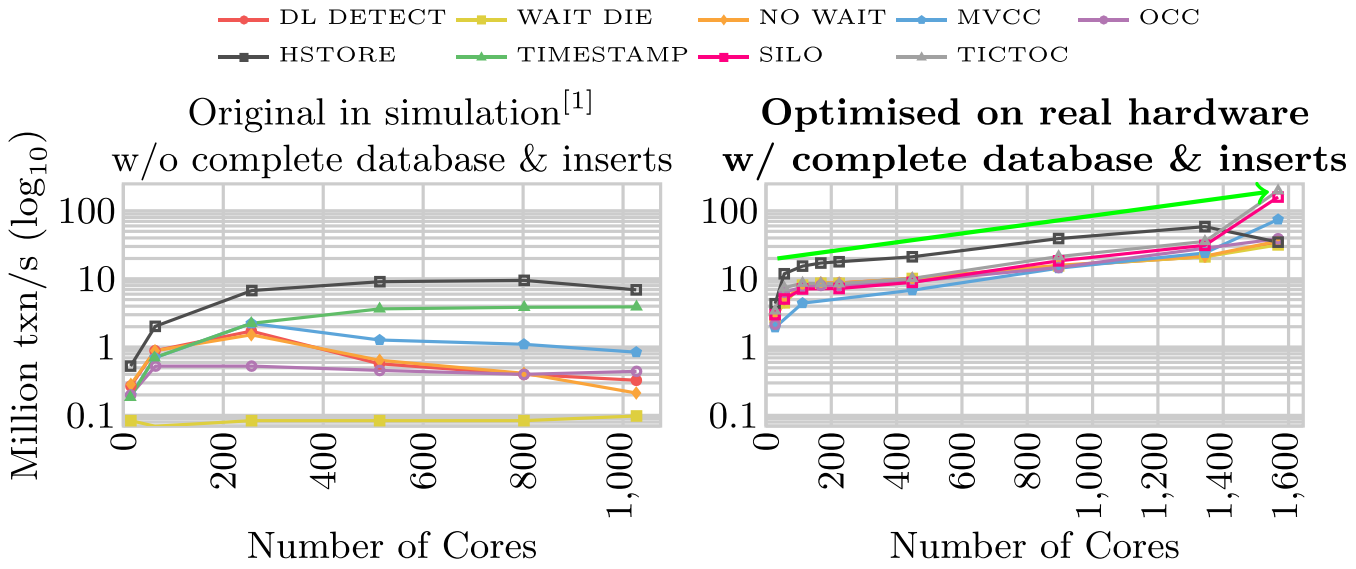
Throughput **of optimised DBx1000** for **low conflict TPC-C** (1024 WH)



➢ **All CC schemes scale with state-of-the-art engineering!**

CLEAR VIEW OF CC ON 1000 CORES (2)

Throughput **of optimised DBx1000** for **low conflict TPC-C** (1024 WH)



➢ **All CC schemes scale with state-of-the-art engineering!**

CONCLUDING THE TALE OF 1000 CORES

OLTP on many cores is not as grim as forecasted, CC can scale!

But the entire architecture needs to scale (not only CC)!

- How to manage physical contention?
- How to address diverse hardware?

New concepts needed for robustly scalable architectures, e.g., optimally configure DBMS architectures![5]
(see Research 18, Wednesday ~11:45AM)

Many thanks to Yu et al. for open sourcing DBx1000!

[5] Bang et al. 2020. Robust Performance of Main-Memory Data Structures by Configuration. SIGMOD. https://doi.org/10.1145/3318464.3389725