

## Mixed Workload CH-benCHmark

**Michael Seibold** #    Florian Funke #    Alfons Kemper #

#Technische Universität München (TUM)

# Outline

Overview

Data Scaling

Benchmark Parameters

Performance Metrics

Conclusion

# Outline

Overview

Data Scaling

Benchmark Parameters

Performance Metrics

Conclusion

# Operational (Real-Time) Business Intelligence

- Process analytical queries directly on operational database
- Without impeding mission-critical transaction processing
- Enabled by advances in hardware architecture

Use cases: for low latency analytics<sup>1</sup>

- BI dashboard: low-latency monitoring
- Interactive customer response (e.g. upsell, antifraud)

---

<sup>1</sup>Curt Monash's Blog (April 10, 2011), Teradata

# Mixed Workload CH-benCHmark

- Outcome of the Dagstuhl Seminar "Robust Query Processing" Fall 2010
  - Organized by Goetz Graefe et al.
  - Breakout Working Group: Workload Management (headed by: Harumi Kuno)
- Research project at Technische Universität München  
<http://www-db.in.tum.de/research/projects/CH-benCHmark/>
- Publications

**DBTest 2011** Richard Cole, Florian Funke, Leo Giakoumakis, Wey Guy, Alfons Kemper, Stefan Krompass, Harumi A. Kuno, Raghunath Othayoth Nambiar, Thomas Neumann, Meikel Poess, Kai-Uwe Sattler, Michael Seibold, Eric Simon and Florian Waas. **The mixed workload CH-benCHmark**. In Proceedings of the 4th International Workshop on Testing Database Systems (DBTest), June 13, 2011, Athens, Greece.

**TPC TC 2011** Florian Funke, Alfons Kemper, Stefan Krompass, Harumi Kuno, Thomas Neumann, Anisoara Nica, Meikel Poess and Michael Seibold. **Metrics for Measuring the Performance of the Mixed Workload CH-benCHmark**. In Proceedings of the 3rd TPC Technology Conference on Performance Evaluation & Benchmarking (TPC TC), August 29, 2011, Seattle WA, USA.

- On-going work: integration into [oltpbenchmark.com](http://oltpbenchmark.com)  
(open-source benchmark framework developed by Carlo Curino, Yahoo! Research)

# Mixed Workload

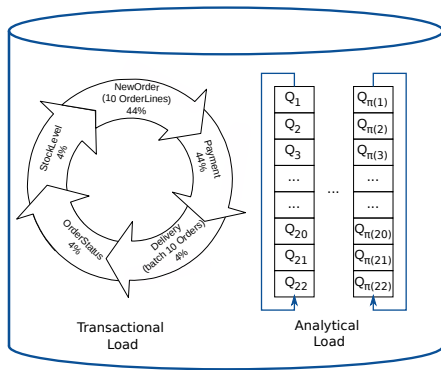
Analytical + transactional loads are

- run **in parallel**
- on the **same tables**
- in a **single database** system

# Single-workload benchmarks

- TPC-C: Transactional workload
- TPC-H:
  - Analytical load and refresh functions
  - BUT: data modification operations performed in bulk when no queries are running
  - No mixed workload
- TPC-C and TPC-H can be combined
  - Installed on a single database instance and run in parallel
  - BUT: different loads are run on separate data
  - No mixed workload

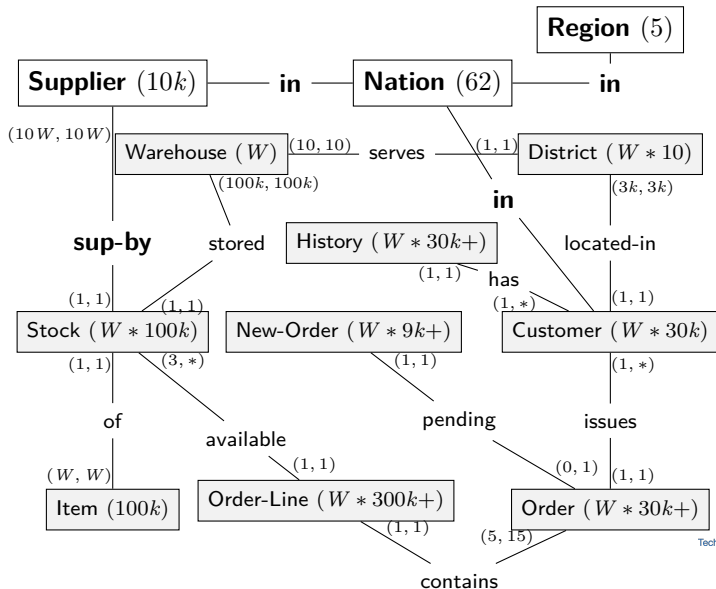
# CH-benCHmark



- Derived from two standardized and widely accepted benchmarks: TPC-C and TPC-H
- Alternative: Composite Benchmark for Transaction processing and operational Reporting (CBTR)
  - Installed on a single database instance and run in parallel
  - BUT: different loads are run on separate data
  - uses the actual data of a real enterprise



# Combine TPC-C and TPC-H: Schema and Initial Database Population



# CH-benCHmark

- Transactional load
  - Unchanged TPC-C business transactions
  - Processed on unchanged TPC-C tables
  - Initial database population according to TPC-C spec
  - But: CH-Benchmark does not simulate terminals
  - Transactional sessions (no think times or keying times)
- Analytical load
  - Read-only query suite modeled after TPC-H
  - All 22 TPC-H queries reformulated to match extended TPC-C schema
  - Goal: Preserve business semantics and syntactical structure
- Transactional load continuously updates (and expands) database
- TPC-H refresh functions are omitted

# Outline

Overview

Data Scaling

Benchmark Parameters

Performance Metrics

Conclusion

# Data Scaling

- Scaling model: Maintains the ratio between
  - Transactional load presented to the system under test
  - Cardinality of the tables accessed by the transactions
  - Required space for storage
  - Number of terminals or sessions generating the system load

# Data Scaling: TPC-C

## Continuous scaling model

- Data volume has to be increased for higher transaction load
- Number of warehouses determines not only the cardinality of the other tables, but also the number of terminals
- Each terminal generates a limited load due to think times and keying times

### *Increase transaction load*

- *Increase number of terminals*
- *Requires a higher number of warehouses*
- *Results in a larger data volume*

### PROBLEM:

- Size of initial database population depends on maximum transactional load
- Mixed workload: Analytical queries on different data volumes

# Data Scaling: TPC-H

## Fixed scale factor model

- Database size is set by a scale factor
- Regardless of system performance

# Data Scaling: CH-benCHmark

## Fixed scale factor model (like TPC-H)

- Fixed number of warehouses, no wait times
- Not necessary to increase the number of warehouses for achieving higher throughput rates
- Allow for high transaction rates on small database sizes
- Scenario: Emerging main-memory database systems (e.g. HyPer, VoltDB, SAP HANA)

## *Different transactional throughput rates*

→ *Difference in data volume growth*

→ *Normalization of response times for analytical queries*

# Outline

Overview

Data Scaling

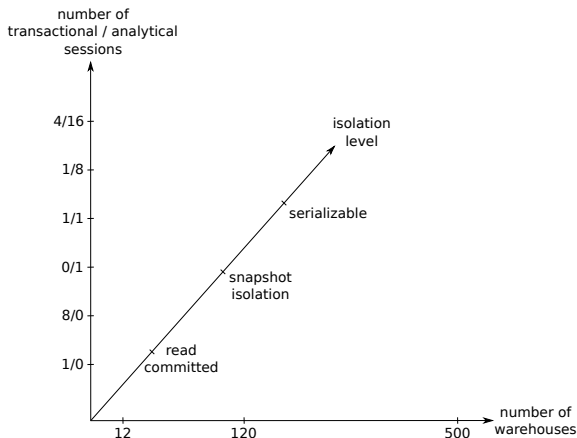
**Benchmark Parameters**

Performance Metrics

Conclusion



# Benchmark Parameters



*Model different workload characteristics and application scenarios:  
e.g. data volumes that fit in main-memory  
or require secondary storage*

# Outline

Overview

Data Scaling

Benchmark Parameters

**Performance Metrics**

Conclusion

# Performance Metrics

Transactional Throughput (tpmCH)	Total number of New-Order transactions completed during the measurement interval divided by the elapsed time of the interval in minutes; New-Order transactions that rollback due to simulated user data entry errors must be included; Similar to the Maximum Qualified Throughput metric of TPC-C
Geometric Mean (ms)	For each query type the average response times of queries completed during the measurement interval is determined and the geometric mean of the average response times of all query types is reported.
Duration Per Query Set (s)	Query set consists of 22 queries, one query per query type; Sum of the average response times of all query types; Reported in seconds
Queries Per Hour (QphCH)	Completed queries per hour; Can be deduced from <i>Duration Per Query Set</i> metric as follows: $\frac{60 \text{ minutes}}{\text{Duration Per Query Set (in seconds)}} \times 22$

# Performance Metrics

*Insert throughput metric of the transactional component interferes with the response-time metric of the analytic component of the mixed workload*

*Transactional Throughput metric and the Queries Per Hour metric are not combined in a single metric*

- Transactional load generated by transactional sessions is not limited by sleep times or keying times, but can only be throttled by the system under test
  - Competing systems under test may prioritize transactions and analytical queries differently
- Aspect would get lost if a single metric were used

# Reporting

Q#	1 Q. stream on data set with increasing data volume			
	System "V"		System "P"	
	average response times (ms)	average normalized response times	average response times (ms)	average normalized response times
Q1	368	42	63632	6001
Q2	126	126	890	890
Q3	1012	121	2606	281
Q4	3368	319	15202	1403
Q5	5722	818	17104	1624
Q6	168	20	22549	2125
Q7	1782	260	177	65
Q8	729	131	4270	517
Q9	703	105	4829	551
Q10	6760	725	141634	7510
Q11	57	57	538	538
Q12	2497	254	39550	3619
Q13	120	27	1437	314
Q14	1430	163	42819	4165
Q15	2876	368	141051	14804
Q16	525	525	7767	7767
Q17	651	79	23405	1926
Q18	7828	587	332413	30598
Q19	614	70	27786	2683
Q20	517	73	6863	819
Q21	2552	280	31812	2710
Q22	245	53	1847	396
Geometric mean (ms)	146		1621	
Normalized geometric mean	859		10814	
Duration per query set (s)	41		931	
Normalized duration per query set	6		92	
Queries per hour (QphH)	1949		86	
Normalized queries per hour (QphH)	15222		868	

# Outline

Overview

Data Scaling

Benchmark Parameters

Performance Metrics

Conclusion

# Conclusion

- Introduced mixed workload CH-benCHmark
- Performance metrics for mixed workload benchmarks
  - Normalization
  - Alternative:
    - Not measure peak transactional and analytical performance
    - But measure how much analytical throughput can be achieved while a fixed transactional throughput rate is maintained
- Future Work: Measure resource usage or energy consumption while fixed transactional and analytical performance is maintained

***Thank you for your attention***