

# Monopedia: Staying Single is Good Enough

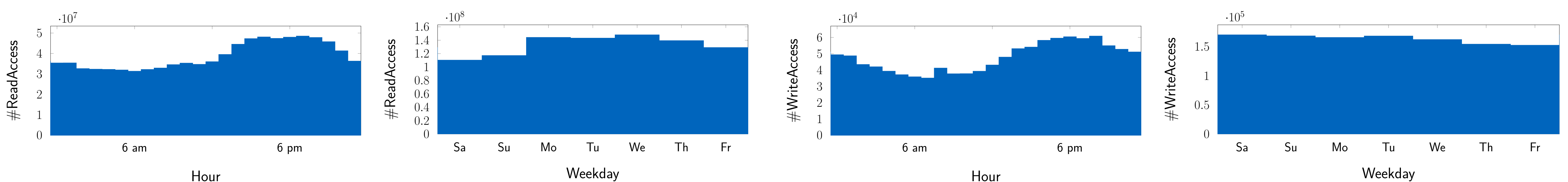
## The HyPer Way for Web Scale Applications

Maximilian E. Schüle, Pascal M. N. Schliski, Thomas Hutzelmann, Tobias Rosenberger  
 Viktor Leis, Dimitri Vorona, Alfons Kemper, Thomas Neumann

{m.schuele,p.schliski,t.hutzelmann,tobias.rosenberger}@tum.de, {leis,vorona,kemper,neumann}@in.tum.de

	#req	traffic [MiB]	#err	req/s	KiB/s	On Time	min	avg	max	last request	title	Rate
Σ Reads	2705	64782007	0	496.9	12766...	91.26%	803	3297.97	298488	3826		
Σ Writes	7	171728	0	0.0	0.0	100.00%	1426	64192....	403219	6487		
R0	2202	53455458	0	332.9	88998...	82.54%	803	3221.30	298488	3759	Isla_Fisher	
R1	503	11326549	0	164.0	38665...	99.99%	987	3374.64	298468	3826	Pope_Sergius_I	
W2	7	171728	0	0.0	0.0	100.00%	1426	64192....	403219	6487	A_Tale_of_Two_C...	

Interactive webinterface of Monopedia Benchmark: summarized informations and informations for each worker thread (request frequency, duration, timeliness, name of last article accessed); rate can be adjusted.



Traffic Data Analysis: March 25-31, 2017: aggregated by week and hour to find out the traffic peaks and the time-dependent load distribution; more read requests during weekdays; nearly constant number of updates.  
 left: SELECT SUM(reads) FROM week WHERE date BETWEEN '2017/03/25' AND '2017/03/31', right: SELECT SUM(writes) FROM week WHERE data BETWEEN '2017/03/25' AND '2017/03/31'

### Abstract

- **web scale applications:** cluster of database servers plus caching layer
- **approach:** only one MM-DBMS instead
- **Monopedia Benchmark:** for web scale applications modeled after Wikipedia

### Results

- one database server suffices
- even without external caching
- **all:** up to 70 % more requests (2665 mixed requests per second)
- **no-wait:** 520 % more requests
- **all-on-time:** 150 % more load

### Sourcecode

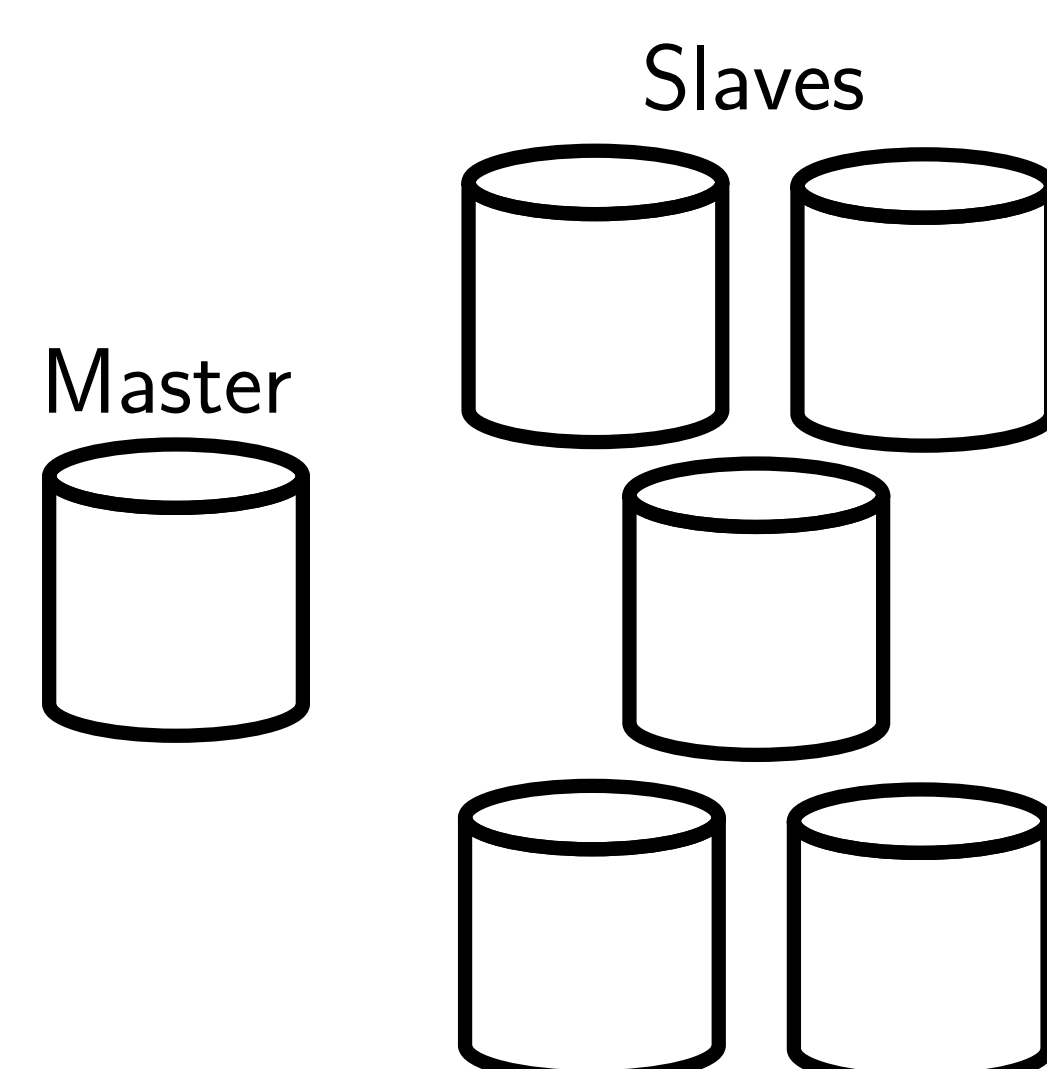


Apache License 2.0

<https://dbkemper4.informatik.tu-muenchen.de/monopedia>

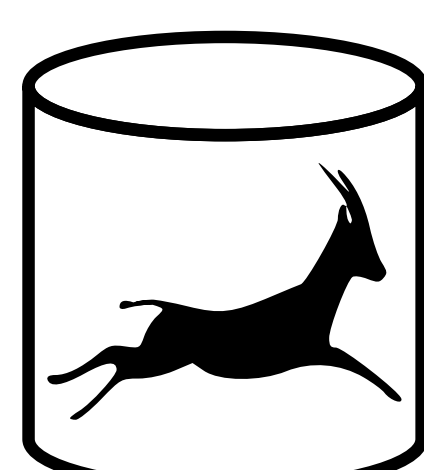
### Conventional Setup

English Wikipedia:  
 6 database servers (MariaDB)



### Centralized Approach

One HyPer server



### Monopedia Benchmark

- **idea:** prove and measure the capability of web scale databases
- **solution:** replay read/update database queries of Wikipedia (load test)
- **benchmark:** load test configuration for simulating concrete traffic

### Traffic Data Analysis: March 25-31, 2017

- **reads (en):** 932,228,284 articles
- **updates (en):** 1,143,247 articles
- **average frequency:** 1541.38 read requests per second  
1.89 write requests per second
- **download the dumps:**

<https://hyper-db.de/monopedia/>

**HyPer**

```
SELECT page_id, page_namespace,
page_title, page_restrictions,
page_is_redirect, page_is_new,
page_random, page_touched,
page_links_updated,
page_latest, page_len,
page_content_model
FROM mediawiki.page
WHERE page_namespace = '0'
AND page_title = $1
LIMIT 1

SELECT old_text,old_flags
FROM mediawiki.pagecontent
WHERE old_id = $1
LIMIT 1
```

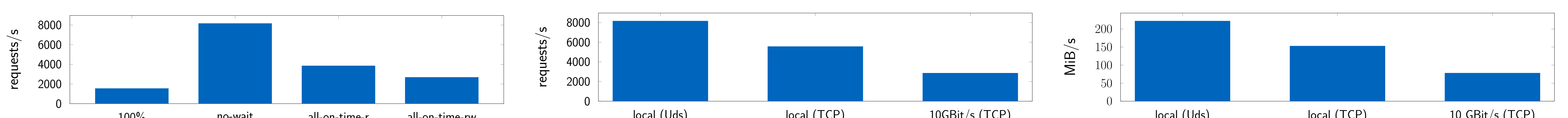
**loadtest**

↑ input: Wikimedia pageview data

```
en Main_memory 3 0
en Main_motion 1 0
en Main_page 54 0
...
```

	#entries	#requests
all	13.3 M	932.2 M
articles	6.9 M	764.1 M
project related	4.7 M	52.0 M
non existent	1.7 M	116.1 M

Summarized Read Traffic of English Wikipedia



Results: First Tests. *100%*: current average load; *no-wait*: possible limits by sending many requests and counting the highest possible throughput; *all-on-time*: highest possible level in respect of the answer's timeliness.  
 Results: Network Tests. *local (Uds)*: database and load test on the same machine, communicating via Unix domain sockets; *local (TCP)*: like the first, but communicating via TCP; *10 Gbit/s (TCP)*: remotely, TCP.