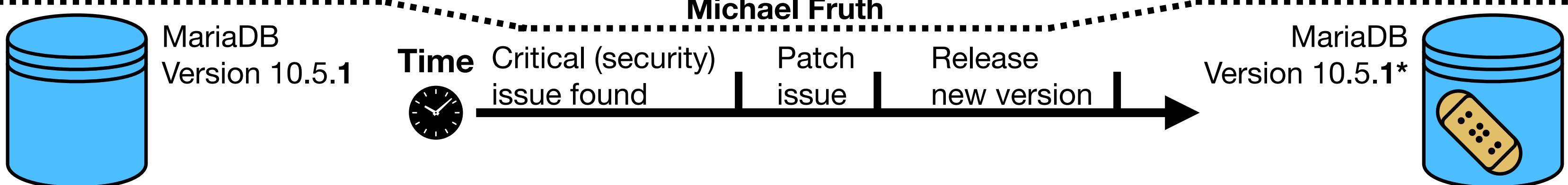


DATABASE MANAGEMENT SYSTEMS



Subject: [PATCH] MDEV-21665: Server crashes in my_qsort2 / Filesort_buffer::sort_buffer

```

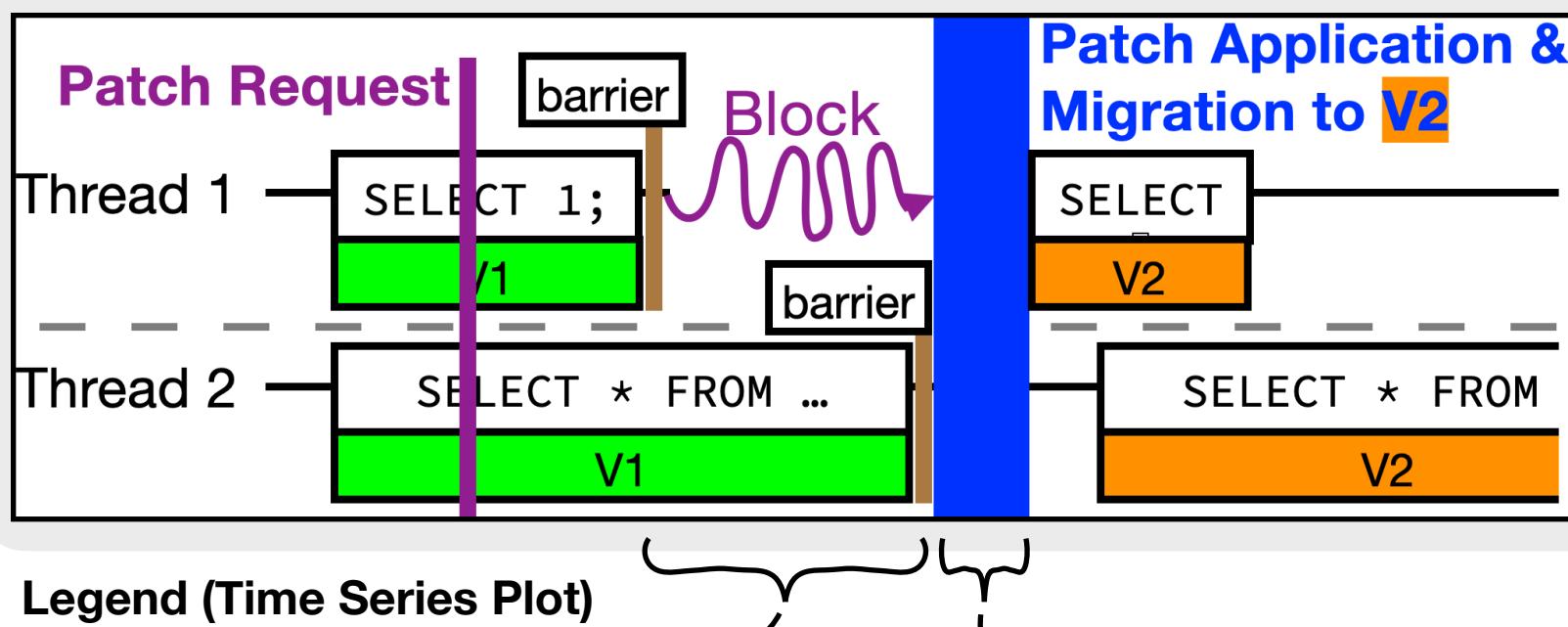
git diff
- set_if_bigger(buff_size, (record_length +sizeof(uchar*)) * MERGEBUFF2);
+ set_if_bigger(buff_size,
+               ALIGN_SIZE((record_length +sizeof(uchar*)) * MERGEBUFF2));
  
```

How to update the DBMS as cloud provider? Challenges:

- 1 A sense of urgency as security leaks might be exploited.
- 2 Rolling updates or hot standby is very expensive.
- 3 For self-service products offered to the customer: When are downtimes acceptable to the customer?

Solution: Live patching allows to patch a running process without restart. Two live patching strategies:

Global Quiescence Each thread needs to reach its barrier before a patch can be applied.



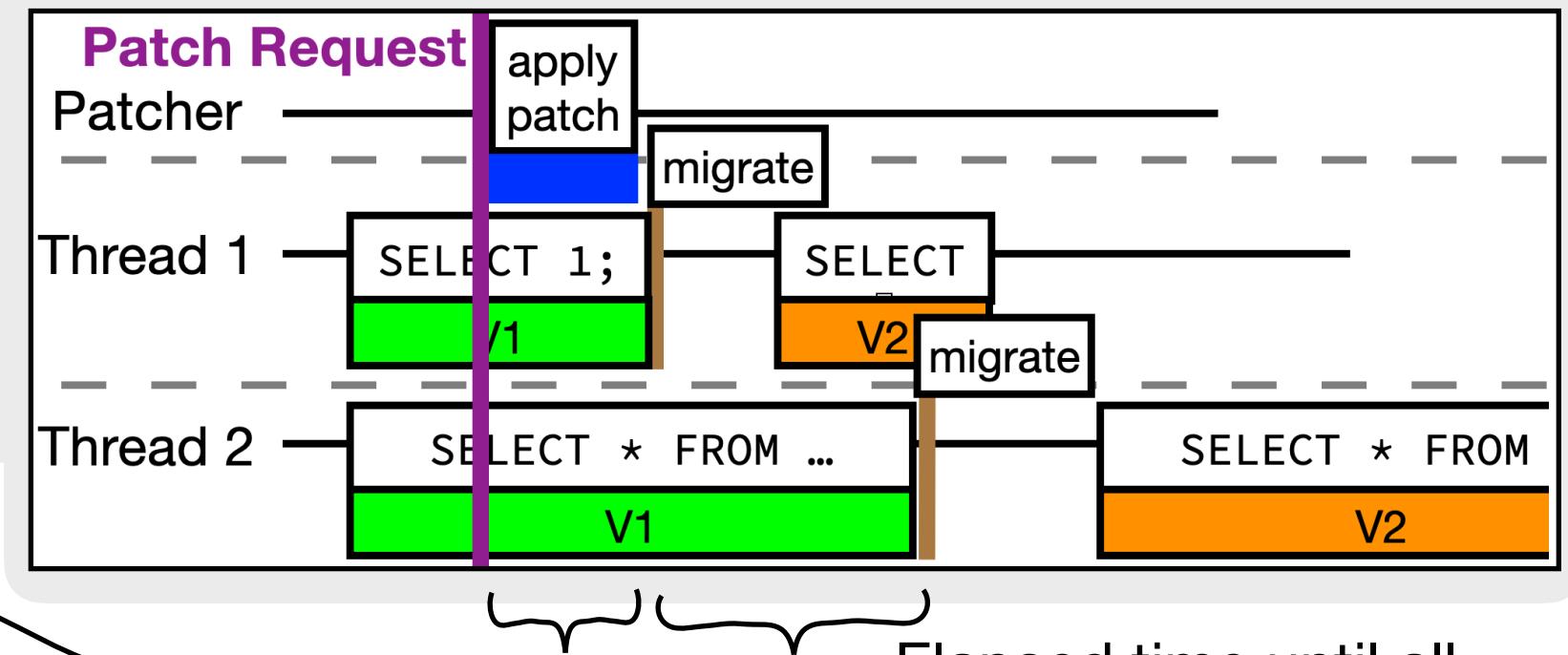
Elapsed time until all threads reached their barrier

Elapsed time for preparing the patch

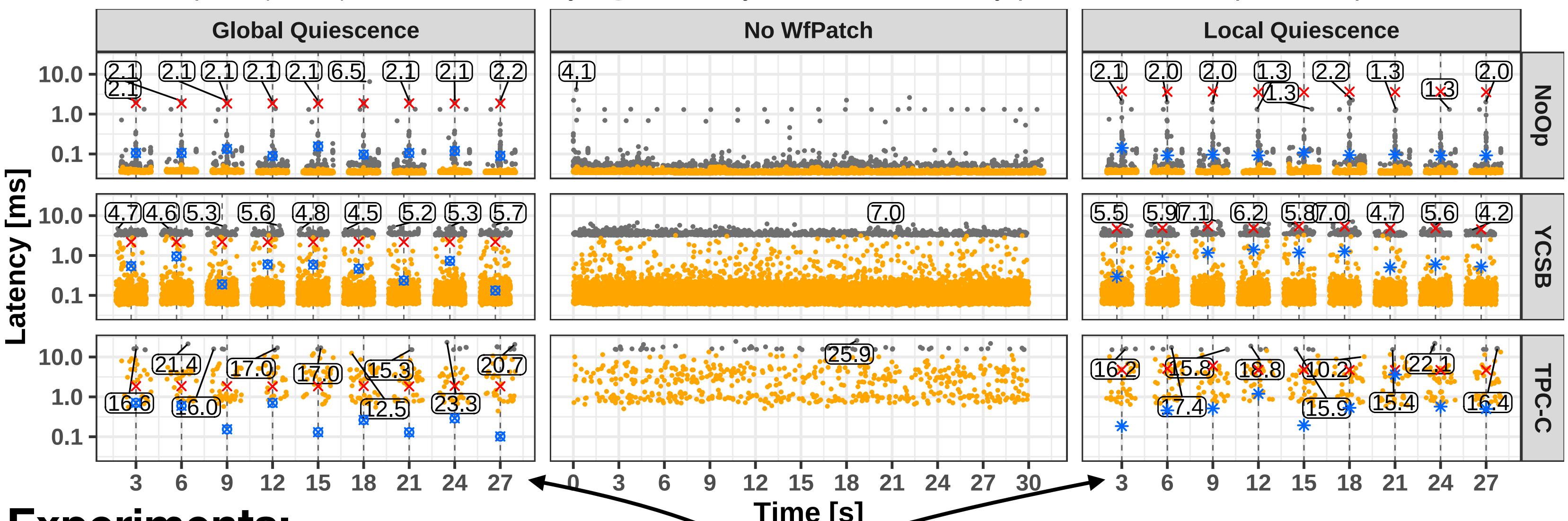
Down-sampled (0.5%) standard latency

Latency >= 99.95th latency percentile

Local Quiescence Each thread is “patched” individually, regardless of the status of the other threads. Live patching facility: WFPATCH [1]



Elapsed time until all threads migrated to the patched process version



Experiments:

- Benchmarking Framework: BenchBase
- Benchmarks: NoOp (No Operation), YCSB, TPC-C
- Database: MariaDB (10.5.1)
- 9 runs; applying patch after “round number × 3 seconds”
- 1 dry-run (without applying a patch; column No WfPatch)

Acknowledgments: I thank Stefanie Scherzinger (University of Passau), Wolfgang Mauerer, Ralf Ramsauer (Technical University of Applied Sciences Regensburg), Daniel Lohmann, Florian Rommel (Leibniz University of Hannover) and Christian Dietrich (Hamburg University of Technology).

References: [1] Florian Rommel, Christian Dietrich, Daniel Friesel, Marcel Köppen, Christoph Borchert, Michael Müller, Olaf Spinczyk, and Daniel Lohmann. 2020. From Global to Local Quiescence: Wait-Free Code Patching of Multi-Threaded Processes. In Proc. OSDI.