Status report for MINOS DB test

- Svetlana increased the max connection limit on dev1 and test performed on 05/11/17
  - Submitted 1000 test jobs with dev1 vs. dev2 / Production group used dev3 in parallel
  - Efficiencies were under 10%, found that CPU core processes were not enough to handle
- Increased the number of CPU core processes from 4 to 8 in dev1, dev2, dev3 on 05/15/17
- Tests performed with changes, results summarized next slide

Change in dev1 for the test

1. Disable DNS lookups: it takes time for each connection to reverse the host DNS name

2. We can decrease wait_timeout to control the Idle connections ("sleep" status)
   Currently wait_timeout=300, we can try wait_timeout=60
   Idle connections consume resources and should be either closed promptly in the application or timed out and closed automatically.

3. These parameters are no harm:
   connect_timeout = 900
   net_read_timeout = 600
   net_write_timeout = 600
   they control hanging connections to close automatically.

4. increase tmp_table_size and max_heap_table_size from 32M
   64M to avoid/minimize the disk writes

5. increase query_cache_size from 32M to 64M

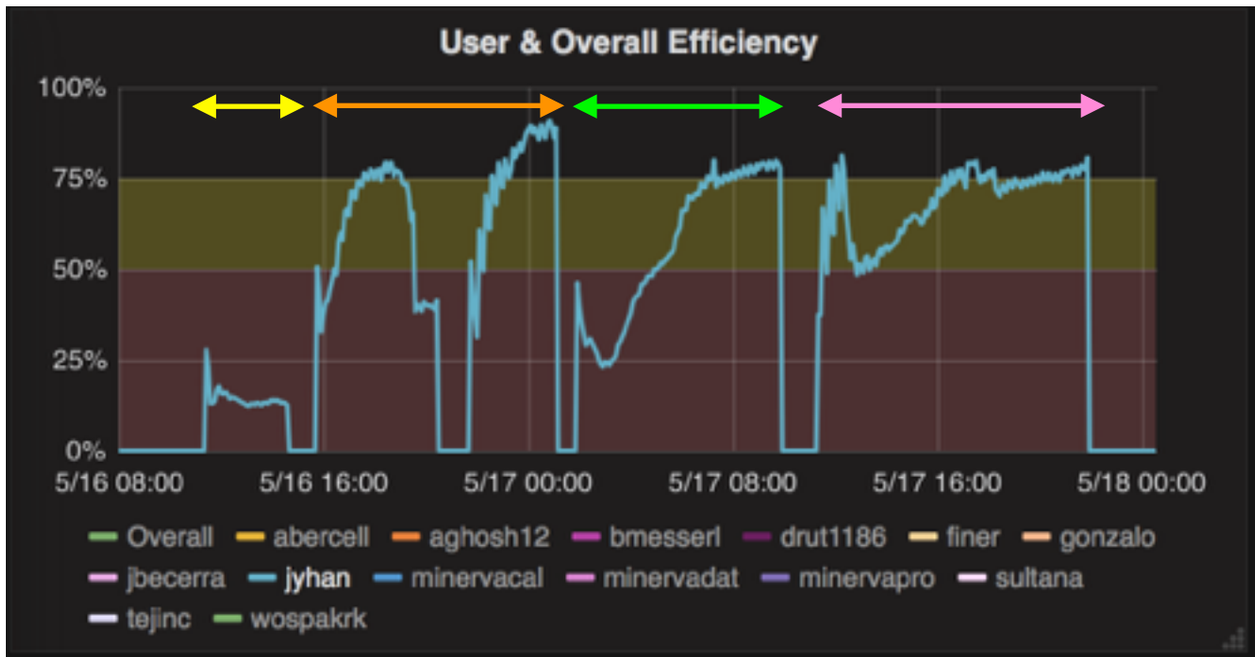
6. We can increase innodb_buffer_pool_size from 122M to 1GB - to ensure that ~60-80% of your working set is in memory.

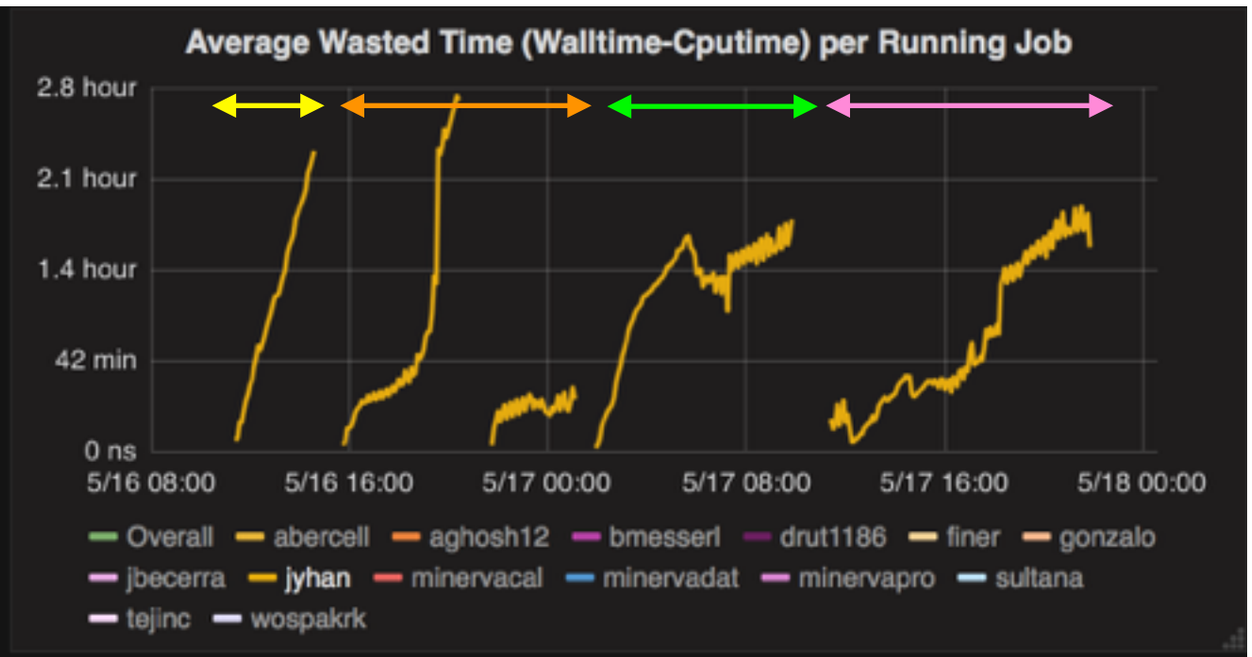
7. We do not use MyISAM explicitly, so we can decrease key_buffer_size from 16M to 64K

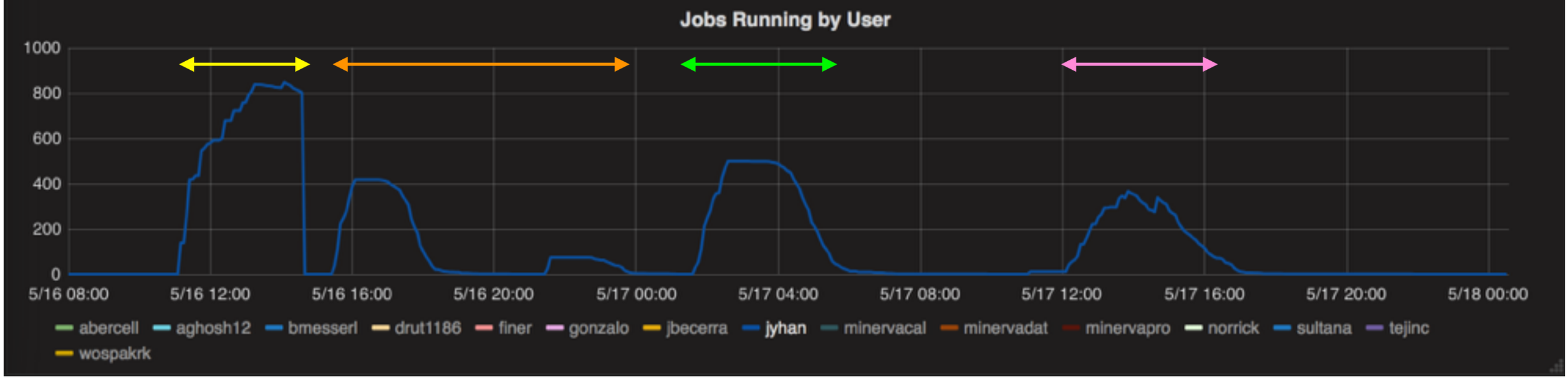
8. We can ask sysadminto disable swap on the disk: sysctl -w vm.swappiness=0
   (I have no permissions to execute this command)

9. And finally increase thread_cache_size from 8 to 16

Color code : 1000 jobs with dev2, 500 jobs with dev(VIP), 500 jobs with dev1, 500 jobs with dev2







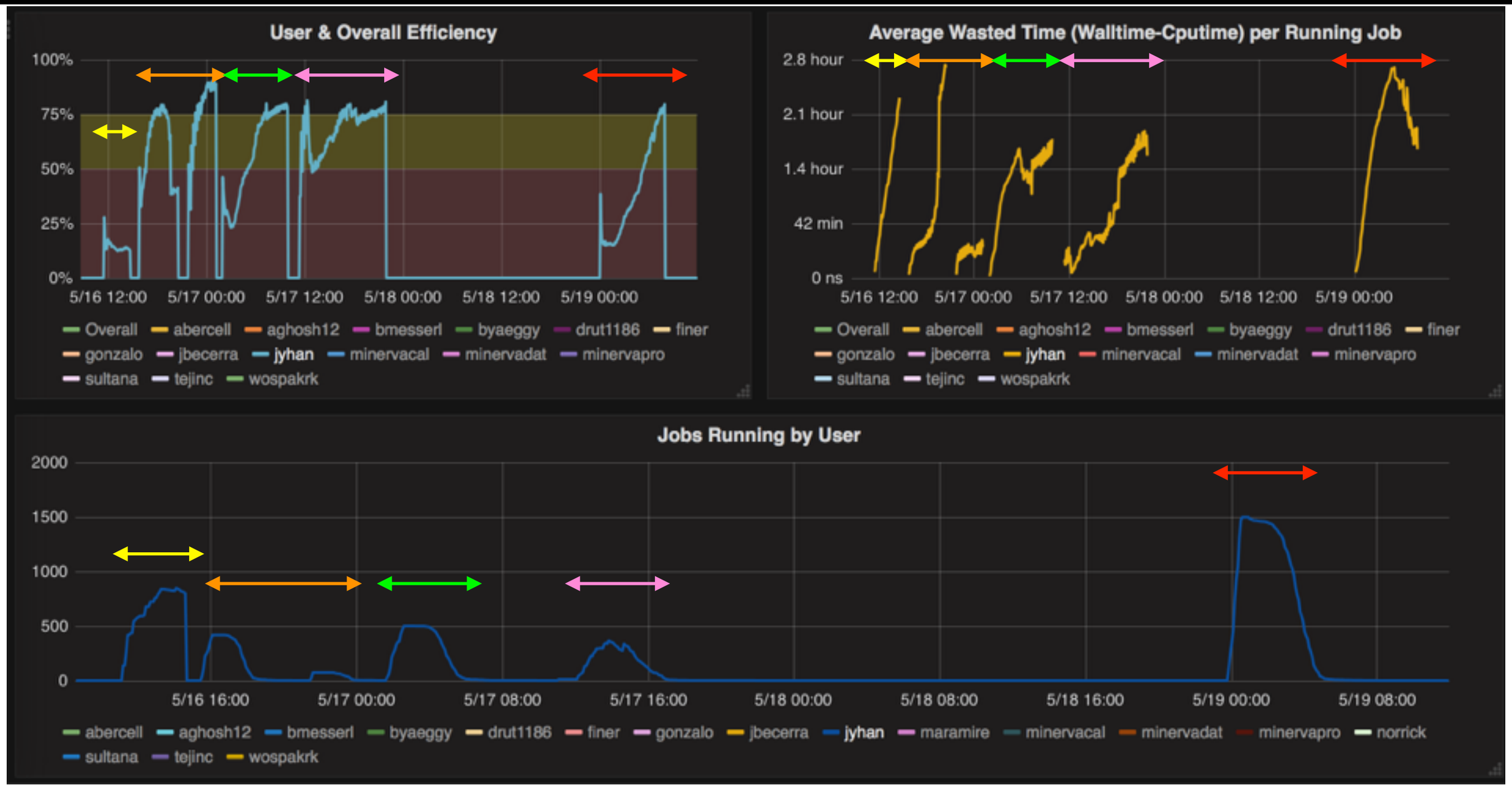
- Test performed after increasing CPU core processes from 4 to 8 in each node
- Submitted 1000 jobs with dev2 → Efficiency was so low (~15%), removed ~400 jobs running
- Decreased jobs down to 500 :
  - Test with dev (Virtual IP) : efficiency started ~50% and went up to 75%, but still wasted time up to ~2.7h
  - Test with dev1 (increased max connection) : efficiency started from 25%, then went up to 75%
  - Test with dev2 (max connection is lower than dev1) : efficiency started from 50% and went up 75%
- It seems that the increased max connection doesn't do much

Test with 1500 jobs for scalability check

- 1500 jobs tested for scalability check
  - 500 jobs submitted with dev1, dev2, dev3, respectively
  - Performance with 500 jobs in each node (dev1, dev2) independently is show p2

1500 jobs with dev1, dev2, dev3

Color code : 1000 jobs with dev2, 500 jobs with dev(VIP), 500 jobs with dev1, 500 jobs with dev2



Even though spread jobs into three nodes, the performance with more than 500 jobs doesn't look good

Summary of test / Question

- 500 jobs test with dev1 vs. dev2 doesn't show much difference

  → Changed parameter might not affect much for the performance (dev2 shows better performance)

- Test with 1500 jobs assigning 500 jobs to dev1, dev2, dev3 each didn't have similar performance with 500 jobs with one node

  ⇒ Is there any limitation to run more than 500 jobs into three Galera clusters?

  What is the limit for the number of jobs in each node?

- What is the best way to connect DB? Connect through Virtual IP address (VIP)?
  - VIP has round-robin method to distribute the jobs into Galera cluster
  - Round-robin method is sufficient for scalability jobs (good way for load balancing)??
    - "Observed" or "Predictive" method is better?
      - We might need to change the connection type of database in the job?
        - Connection type now : hold connection until job is done → open and close connection
- Would it help if we gave DBAs some examples of the queries we're running?

  ⇒ DBAs can give an advice to improve the performance in queries

- Arrange another test with DBAs, so DBAs can diagnose the problem
- We suspect that uploading tables into DB (temp database) would cause the problem

  ⇒ Will try to save tables into DB to avoid uploading procedure which causes the dead lock

- Usage of minervadb01.fnal.gov : used for LE data, read-only replica (v 5.1.41, MYISAM engine)