

Firebird Tour 2017: Performance

Vlad Khorsun, Firebird Project



About Firebird Tour 2017

- Firebird Tour 2017 is organized by [Firebird Project](#), [IBSurgeon](#) and [IBPhoenix](#), and devoted to Firebird Performance.
- The Platinum sponsor is [Moscow Exchange](#)
- Tour's locations and dates:
 - October 3, 2017 – Prague, Czech Republic
 - October 5, 2017 – Bad Sassendorf, Germany
 - November 3, 2017 – Moscow, Russia





MOSCOW EXCHANGE

- Platinum Sponsor
- Sponsor of
 - «Firebird 2.5 SQL Language Reference»
 - «Firebird 3.0 SQL Language Reference»
 - «Firebird 3.0 Developer Guide»
 - «Firebird 3.0 Operations Guide»
- Sponsor of Firebird 2017 Tour seminars
- www.moex.com





- Replication, Recovery and Optimization for Firebird since 2002
- Platinum Sponsor of Firebird Foundation
- Based in Moscow, Russia

www.ib-aid.com

Performance related changes in FB3

- Engine internals
 - SMP-friendly Super Server
 - Hash Join algorithm
 - Improved nbackup synchronization
 - Reduced number of Pointer Page fetches
 - Delayed Header and/or TIP Page writes for read-only transactions



Performance related changes in FB3

- ODS12
 - Primary and Secondary Data Pages
 - Flag Swept for Data and Pointer Pages
 - Allocate and free Data Pages by extents
 - Extent is a group of 8 consecutive pages
 - SCN Pages for physical backup



Benchmarks

- Test machine
 - CPU: Intel i4770, 4/8 cores
 - HDD: 2 SATA drives in RAID 1 (mirror)
 - SSD: Samsung 850 Pro, SATA
 - RAM: 16 GB
 - OS: Windows 7



Benchmarks

- Test database
 - Generated by TPCC load tool
 - Physical parameters
 - 8 KB page size
 - 8.75 GB
 - Forced Writes = ON
 - Restored from the file copy before each test run

Table	Rows	Data Pages
CUSTOMER	3`000`000	235`808
DISTRICT	1`000	24
ITEM	100`000	1`520
STOCK	10`000`000	421`528
WAREHOUSE	100	2



Read-only, single table

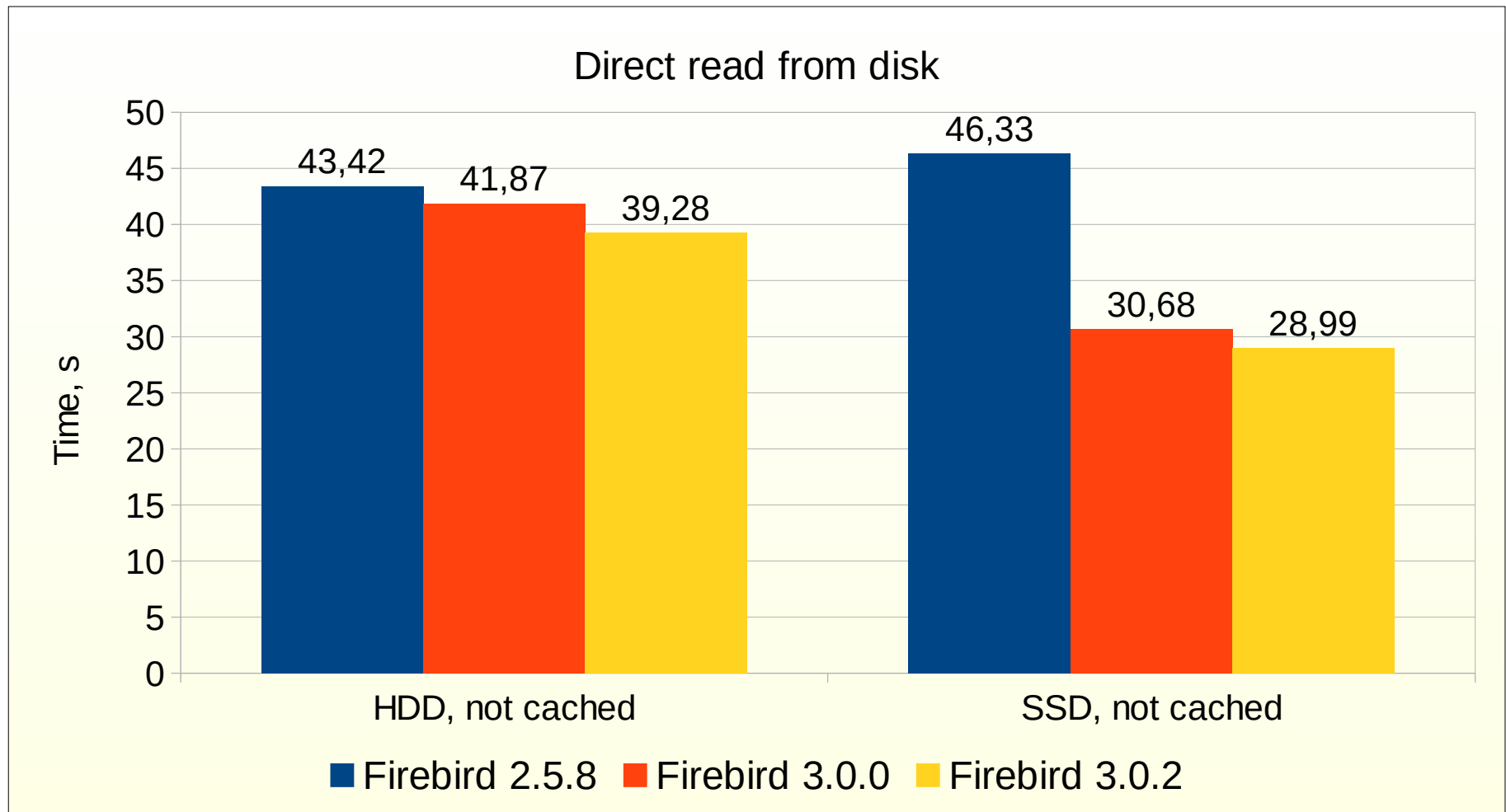
- Test Query:
 - `SELECT COUNT (*) FROM STOCK`
- Page cache
 - 500`000 buffers
- Caching effect
 - Direct read from disk, not cached by OS nor by Firebird
 - HDD
 - SSD
 - Cached by OS, Firebird cache is empty
 - all reads are from OS cache, no read from physical disk
 - Fully cached by Firebird
 - no reads at all



Read-only, single table

- Test Query:

- `SELECT COUNT (*) FROM STOCK`



Read-only, single table

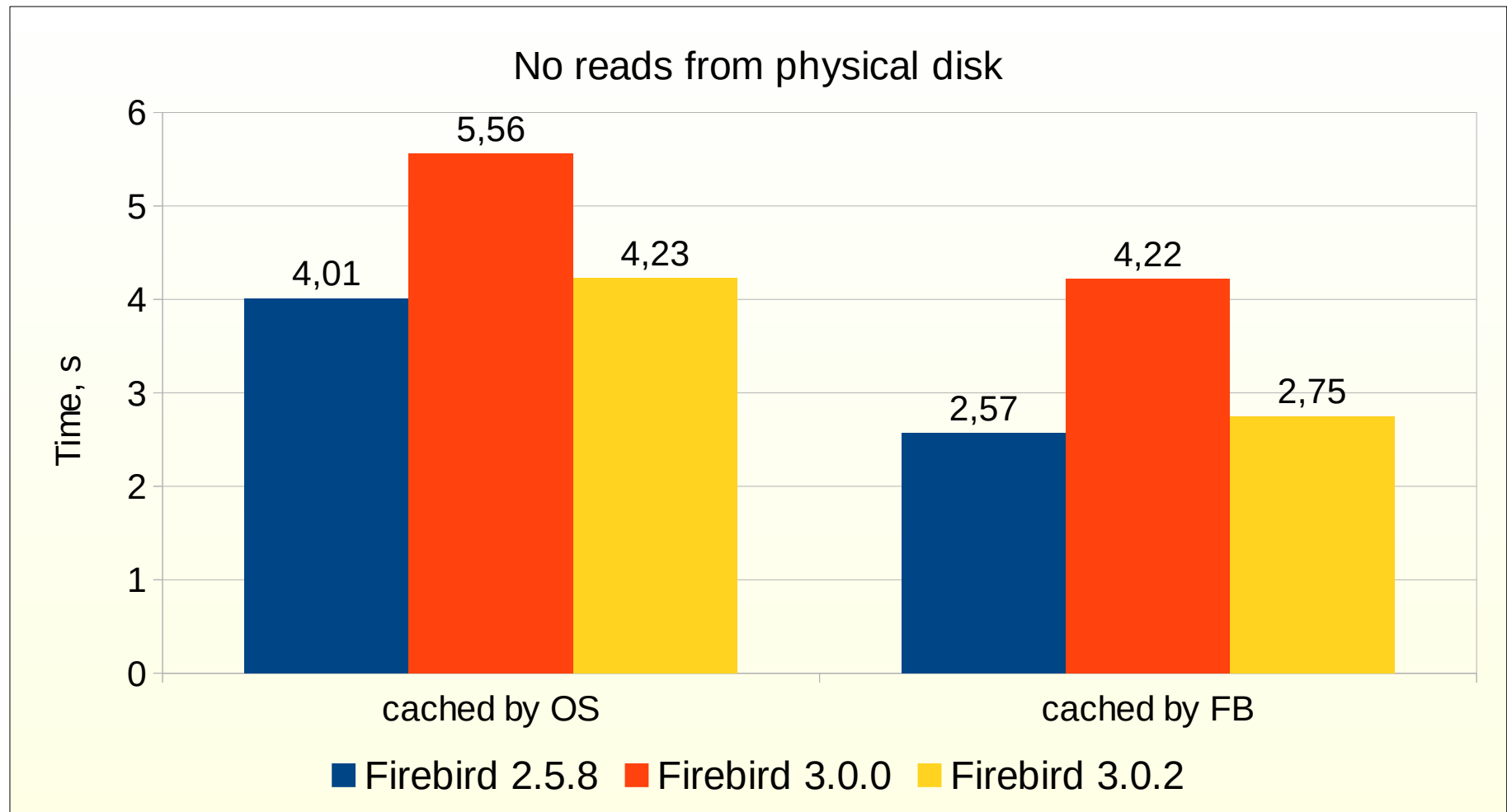
- Direct read from disk
 - HDD
 - Almost identical speed while FB3 is a bit faster than FB 2.5
 - Due to extents allocation?
 - SSD
 - FB 2.5 slower than on HDD!
 - It requires extra investigation
 - FB 3 30% faster on SSD
 - As expected



Read-only, single table

- Test Query:

- `SELECT COUNT (*) FROM STOCK`



Read-only, single table

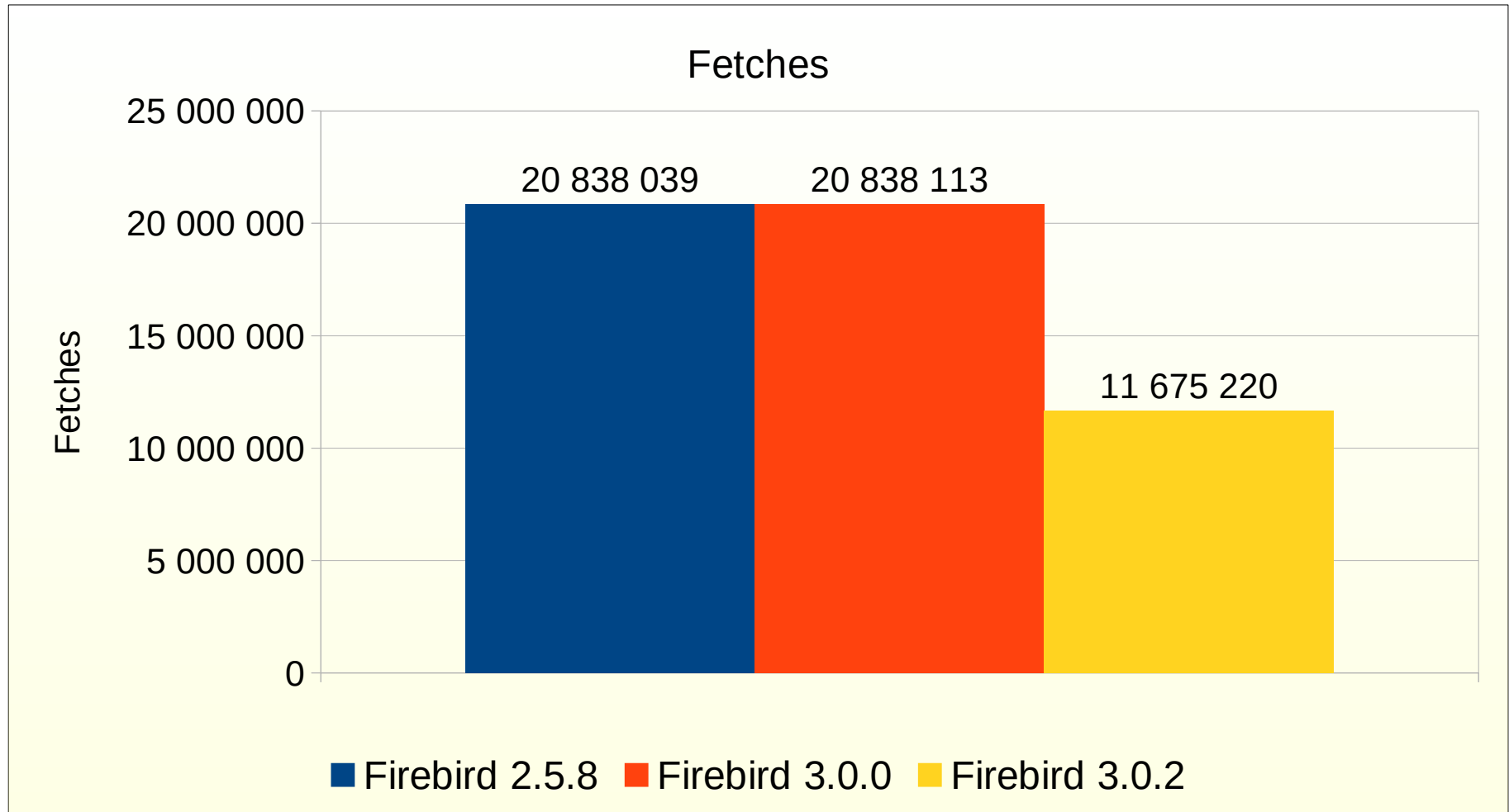
- Firebird 3.0.0 is much slower than Firebird 2.5.8
 - Minimal cost of page access in Super Server before Firebird 3 is very cheap : just increment and decrement
 - Minimal cost of page access in Super Server in Firebird 3 is much larger : four interlocked operations
- Firebird 3.0.2 is almost as fast as Firebird 2.5.8
 - Small internal cache of physical numbers of Data Pages
 - Allows to avoid most accesses of Pointer Pages
 - Reduces number of page accesses (fetches) almost two times



Read-only, single table

- Test Query:

- `SELECT COUNT (*) FROM STOCK`



Read-only, LOOP JOIN

- Query

```
SELECT W.W_NAME, SUM(S.S_QUANTITY),  
       SUM(I.I_PRICE * S.S_QUANTITY)  
FROM WAREHOUSE W JOIN STOCK S  
      ON W.W_ID = S.S_W_ID  
JOIN ITEM I  
      ON S.S_I_ID = I.I_ID  
GROUP BY W.W_NAME
```

- PLAN

```
SORT (JOIN (  
      W NATURAL,  
      S INDEX (STOCK_PK),  
      I INDEX (ITEM_PK)))
```

- Result set

- 100 rows



Read-only, LOOP JOIN

Execution statistics

	Firebird 2.5.8	Firebird 3.0.2	Firebird 2.5.8	Firebird 3.0.2
Buffers	50 000	50 000	500 000	500 000
Fetches	70 015 390	60 413 829	70 015 390	60 413 828
Reads	10 188 968	10 300 878	0	0
Time	74,99	83,53	35,93	38,55

Per-table statistics

Table	Indexed	Non-indexed
ITEM	10 000 000	0
STOCK	10 000 000	0
WAREHOUSE	0	100



Read-only, LOOP JOIN

- Firebird 3 still 7-11% slower than Firebird 2.5
- Firebird 3 makes a bit less fetches but not enough less to catch up Firebird 2.5
- Table STOCK read all 10M records
 - Using index – far not efficient
- Table ITEMS read all 100K records 100 times!
 - Using index – far not efficient
- Plan and performance is very far from optimal

- Try to disable index access to avoid LOOP join?



Read-only, MERGE JOIN

- Query

```
SELECT W.W_NAME, SUM(S.S_QUANTITY),  
       SUM(I.I_PRICE * S.S_QUANTITY)  
FROM WAREHOUSE W JOIN STOCK S  
      ON W.W_ID +0 = S.S_W_ID +0  
JOIN ITEM I  
      ON S.S_I_ID = I.I_ID +0  
GROUP BY W.W_NAME
```

- Plan, Firebird 2.5.8

```
SORT (MERGE (  
    SORT (MERGE (  
        SORT (S NATURAL),  
        SORT (I NATURAL))  
    ),  
    SORT (W NATURAL))  
)
```



Read-only, MERGE JOIN

- MERGE JOIN algorithm
 - Requires both input datasources to be sorted on the join columns
 - Sort both inputs
 - Join sorted data within one pass



Read-only, HASH JOIN

- Query

```
SELECT W.W_NAME, SUM(S.S_QUANTITY),  
       SUM(I.I_PRICE * S.S_QUANTITY)  
FROM WAREHOUSE W JOIN STOCK S  
      ON W.W_ID +0 = S.S_W_ID +0  
   JOIN ITEM I  
      ON S.S_I_ID = I.I_ID +0  
GROUP BY W.W_NAME
```

- Plan, Firebird 3.0.2

```
SORT (  
  HASH (  
    HASH (S NATURAL, I NATURAL),  
    W NATURAL  
  )  
)
```



Read-only, HASH JOIN

- HASH JOIN algorithm
 - Set larger datasource as left input, smaller datasource as right input
 - Build hash table for the right (smaller) datasource
 - Scan left datasource in natural order and probe hash table for matching rows



Read-only, MERGE vs HASH

Execution statistics

	Firebird 2.5.8	Firebird 3.0.2	Firebird 2.5.8	Firebird 3.0.2
Buffers	50 000	50 000	500 000	500 000
Fetches	21 040 678	12 027 694	21 040 678	12 027 694
Reads	420 343	423 272	0	0
Time	29,48	23,02	28,27	21,54

Per-table statistics

Table	Indexed	Non-indexed
ITEM	0	100 000
STOCK	0	10 000 000
WAREHOUSE	0	100

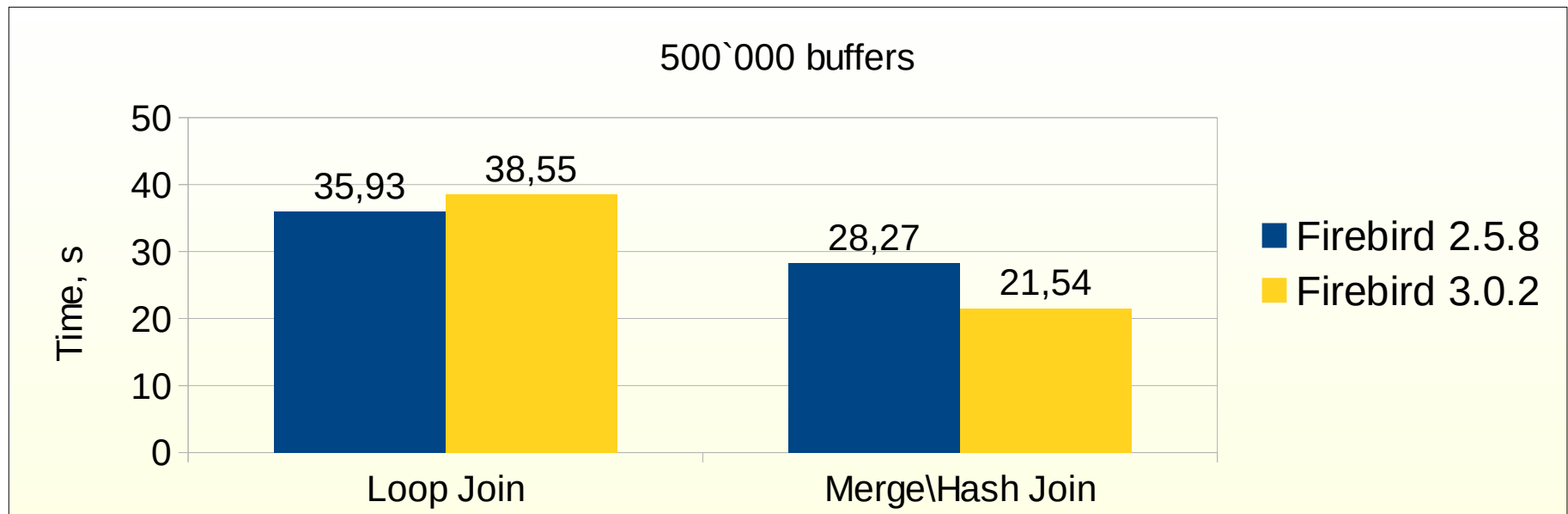
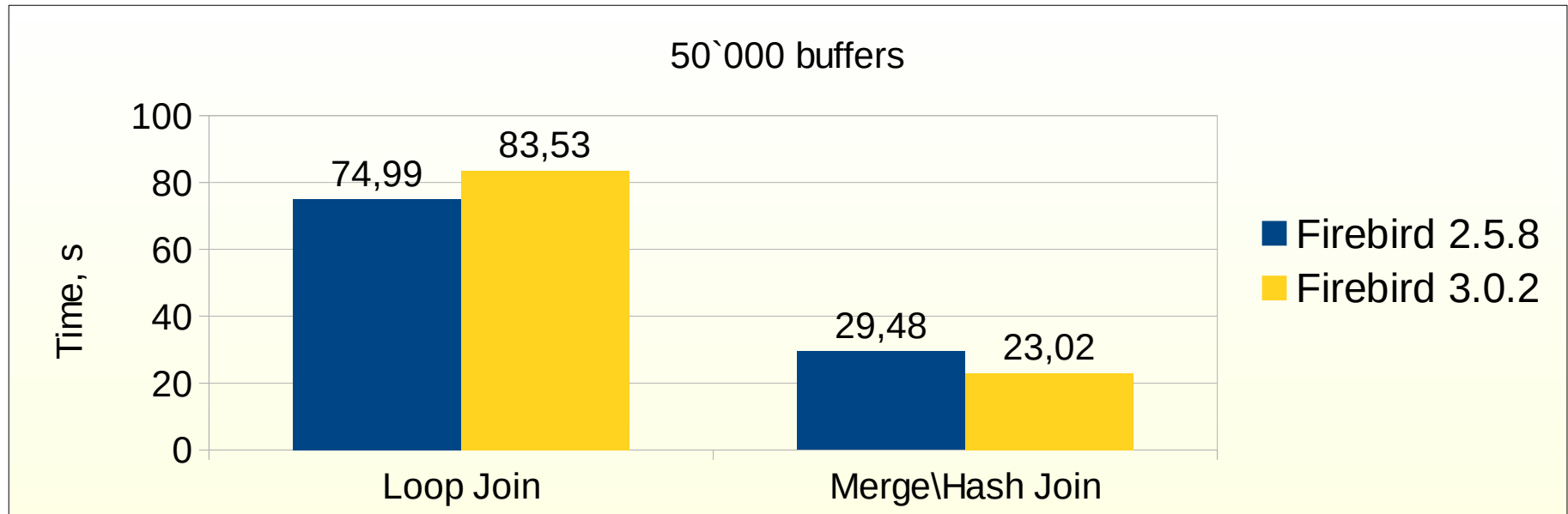


Read-only, MERGE vs HASH

- Firebird 2.5, MERGE vs LOOP
 - >23 times less reads (420K vs 10M)
 - 3.5 times less fetches (21M vs 70M)
- Firebird 3, HASH vs LOOP
 - >23 times less reads (423K vs 10M)
 - 5 times less fetches (12M vs 60M)
- Both versions
 - Much faster
 - Almost independent on Firebird cache size
- Firebird 3 now >22% faster than Firebird 2.5



Read-only, LOOP vs MERGE vs HASH



Read-only, multi-threaded

- Multi-threaded read-only benchmark
 - Table STOCK have 10'000'000 records
 - 100 warehouses, 100'000 items in each
 - Each reader reads random items in own warehouse
 - We don't test network subsystem
 - client application executes procedure, which selects a number of random items (100 by default) of the same warehouse
- ```
SELECT * FROM BENCH_1 (:W_ID, 100)
```
- We don't test IO subsystem
    - before each test series we ensure whole table STOCK is fully cached by filesystem
  - We do test Firebird scalability



# Read-only, multi-threaded

```
CREATE OR ALTER PROCEDURE BENCH_1 (
 W_ID INTEGER,
 NUM INTEGER)
RETURNS (
 I_ID INTEGER)
AS
DECLARE I1 INTEGER;
BEGIN
 WHILE (NUM > 0) DO
 BEGIN
 I1 = RAND() * (100000 - 1);

 SELECT S.S_I_ID
 FROM STOCK S
 WHERE S.S_W_ID = :W_ID AND S.S_I_ID = :I1
 INTO :I_ID;

 SUSPEND;

 NUM = NUM - 1;
 END
 END
END
```



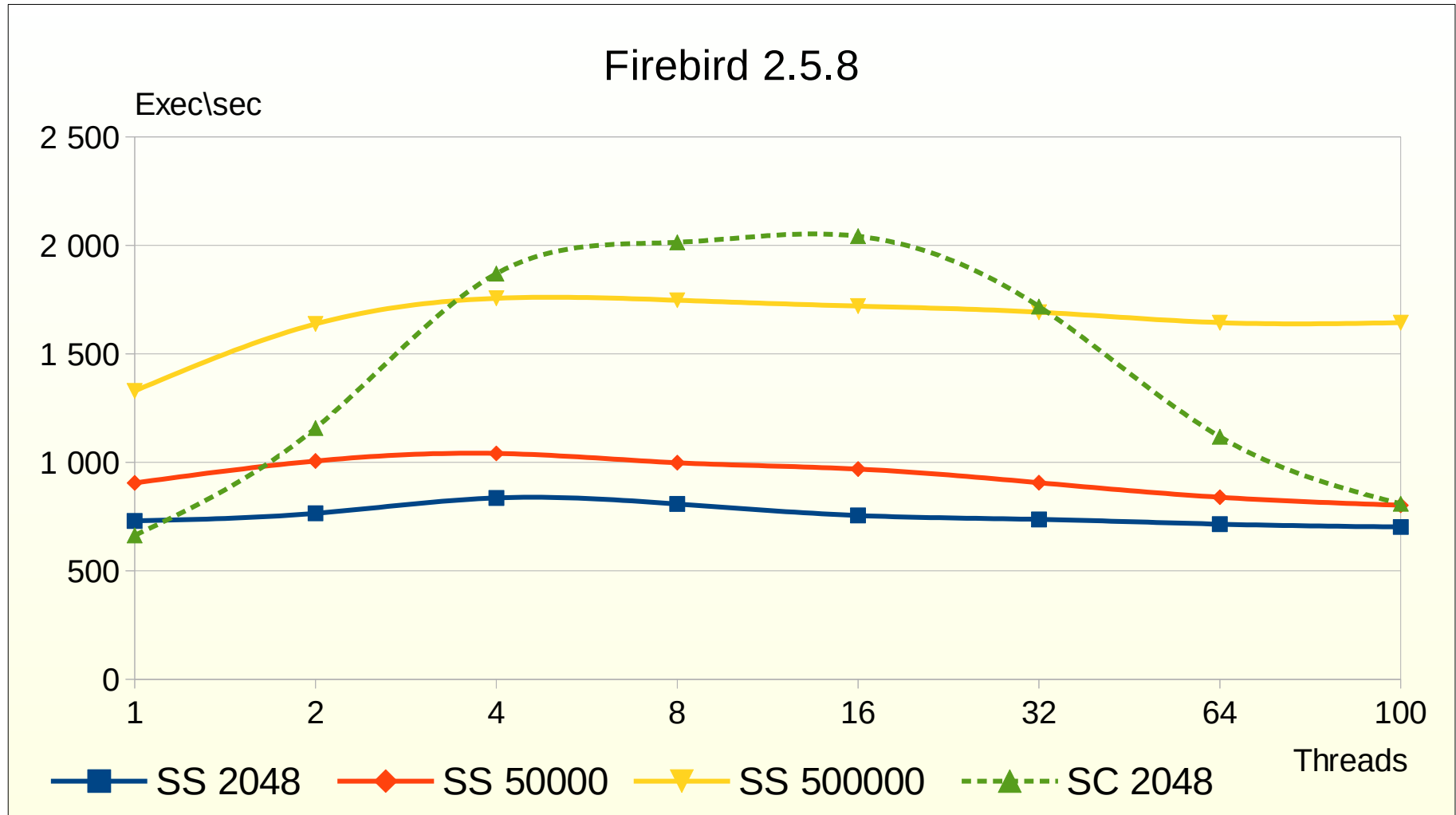
# Read-only, multi-threaded

- Multi-threaded read-only benchmark
  - Firebird versions
    - 2.5.8 base version for comparison
    - 3.0.0 first release of v3, have some perf problems
    - 3.0.2 current release of v3, have some improvements
  - Super Server
    - default cache size (2 048)
    - medium cache size (50 000)
    - huge cache size (500 000)
  - Classic
    - big cache size (2 048)
    - SuperClassic for Firebird 2.5
    - Classic mode for Firebird 3



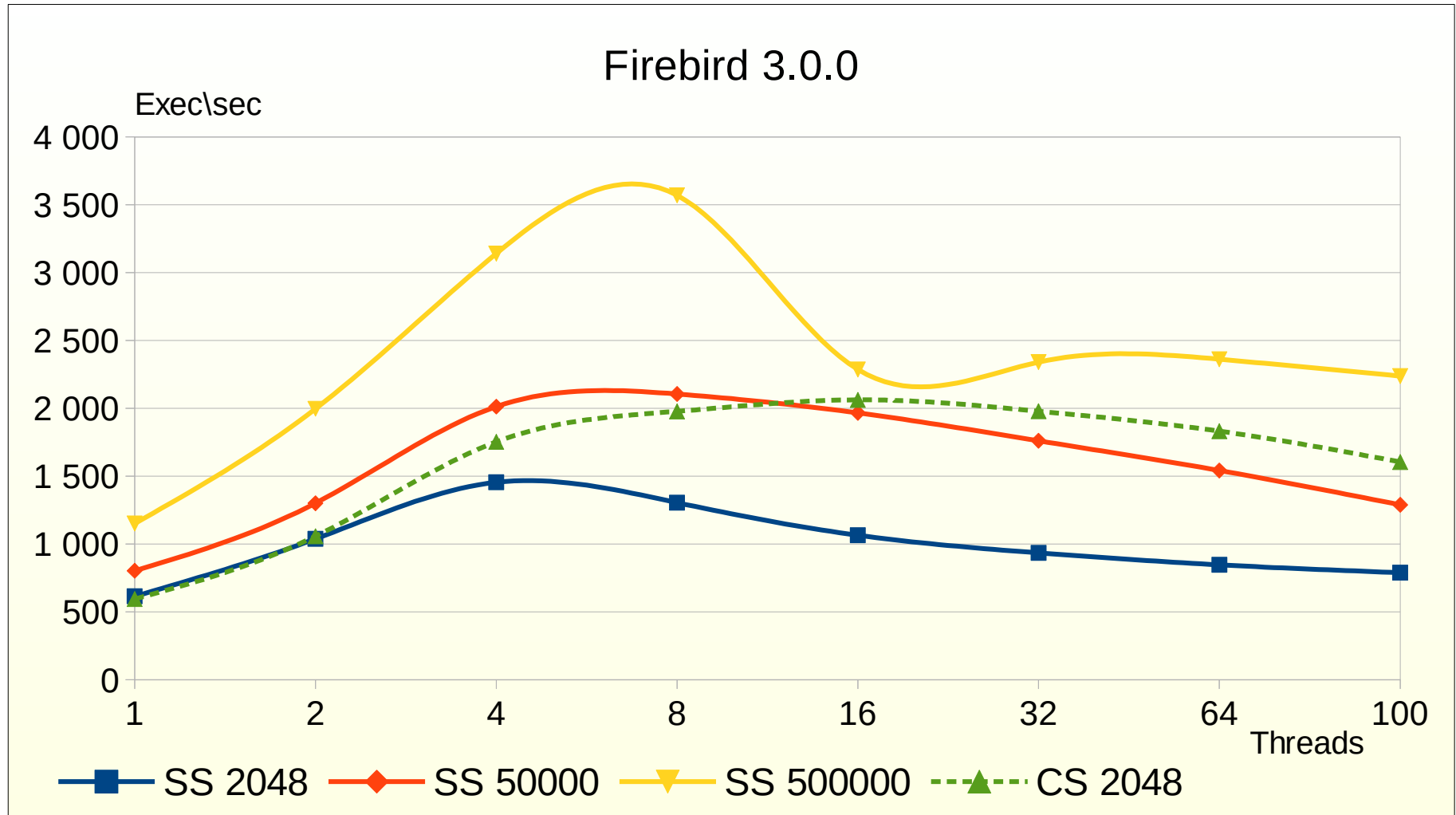
# Read-only, multi-threaded

- Different modes within same version



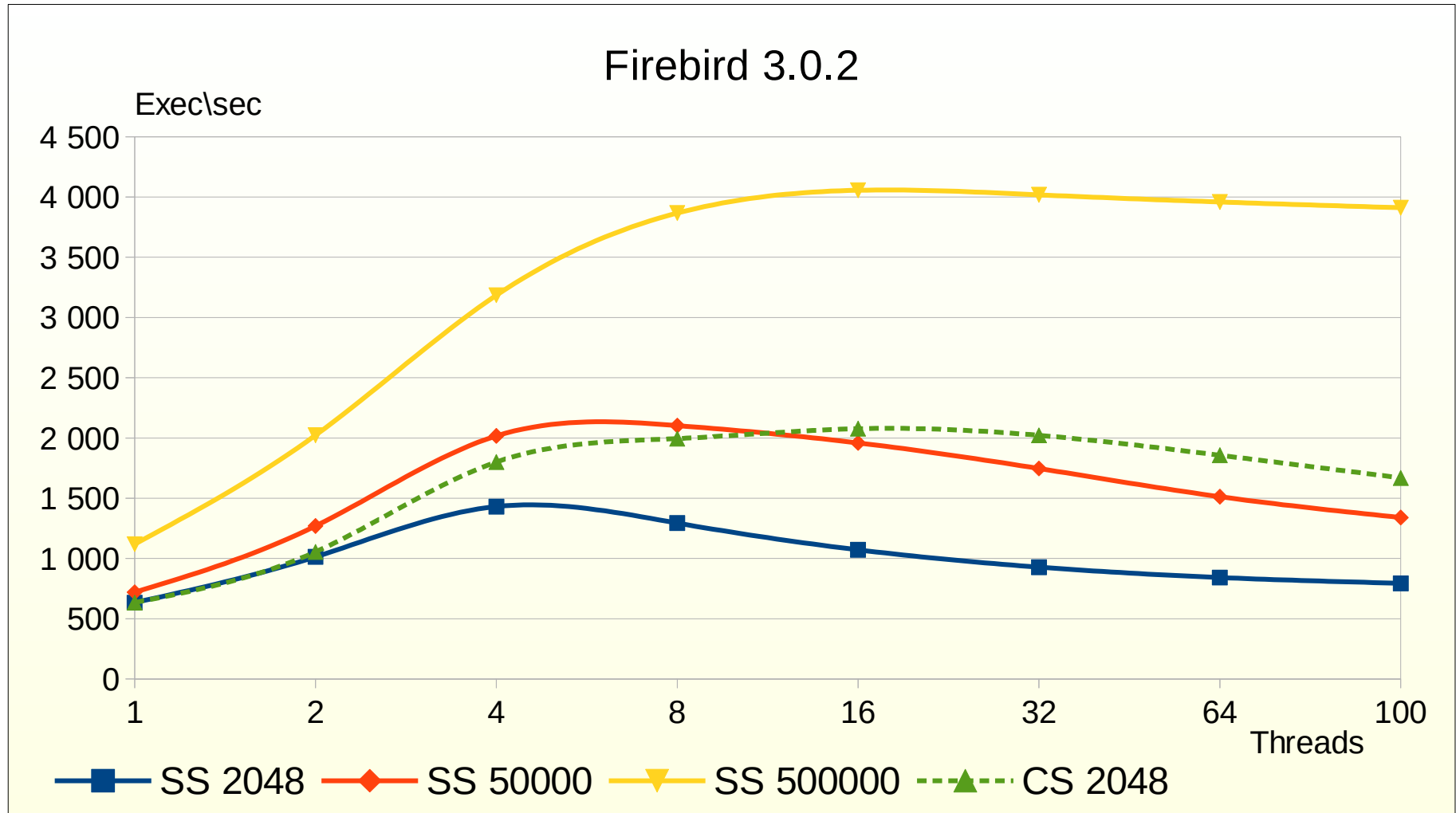
# Read-only, multi-threaded

- Different modes within same version



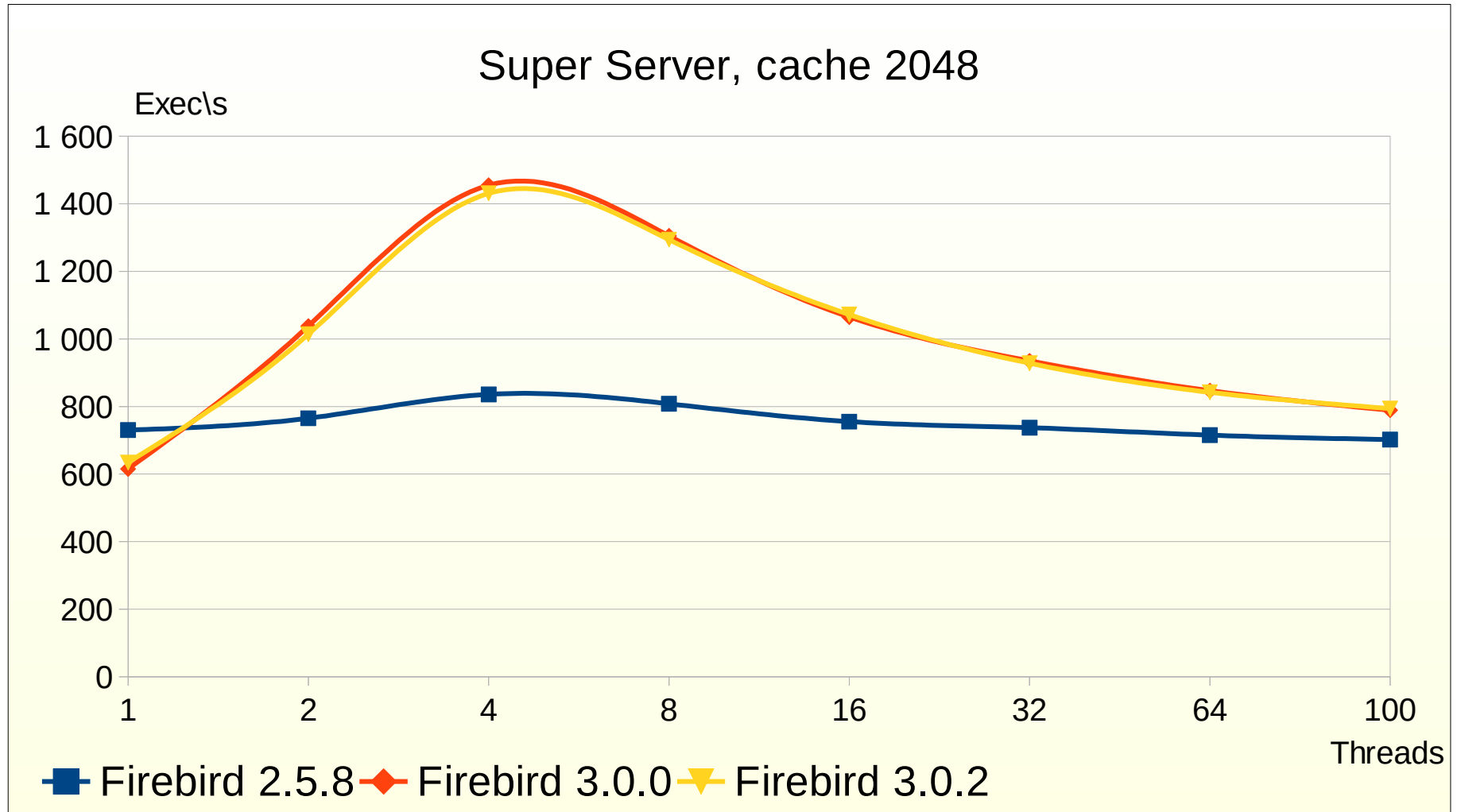
# Read-only, multi-threaded

- Different modes within same version



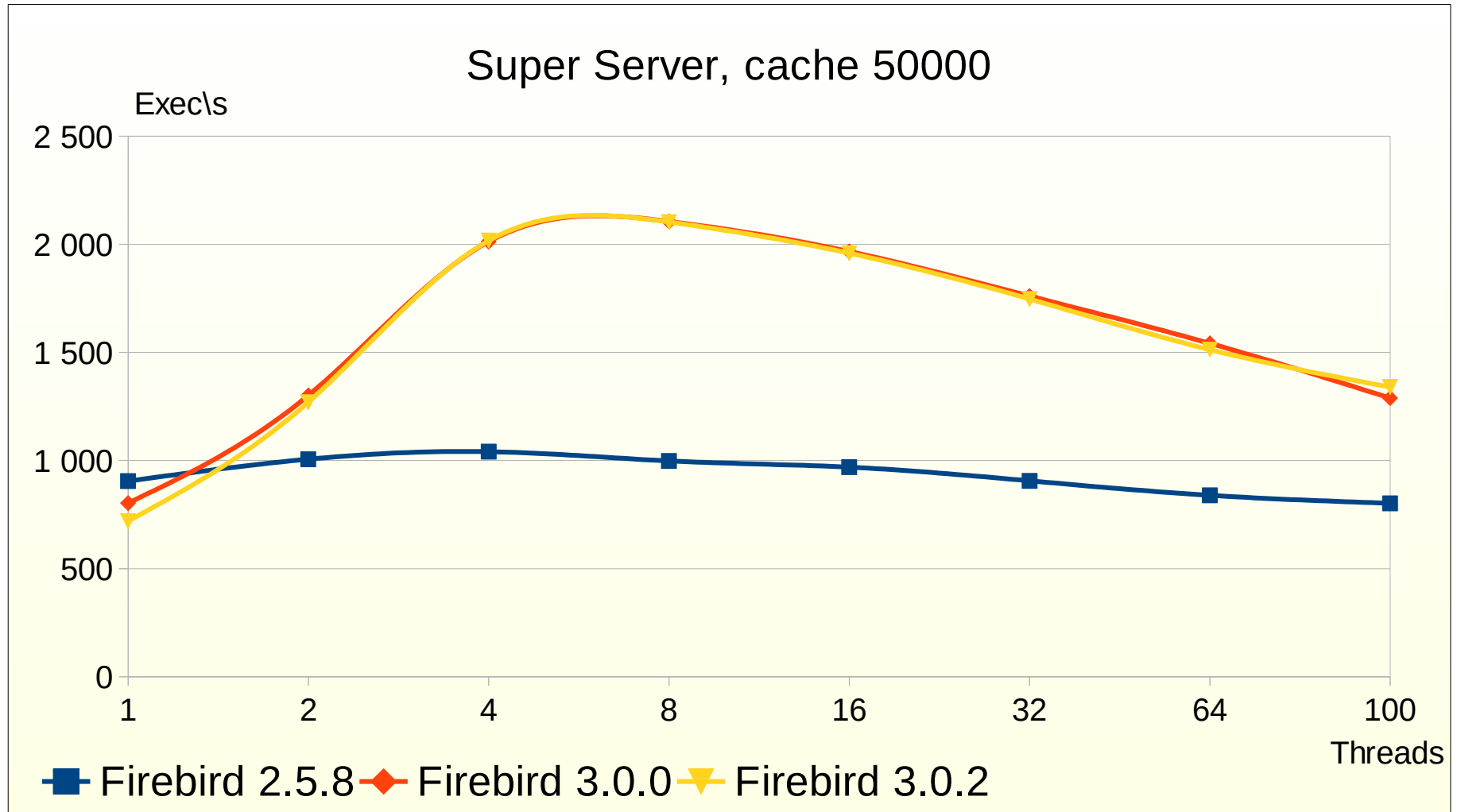
# Read-only, multi-threaded

- Different versions, same mode



# Read-only, multi-threaded

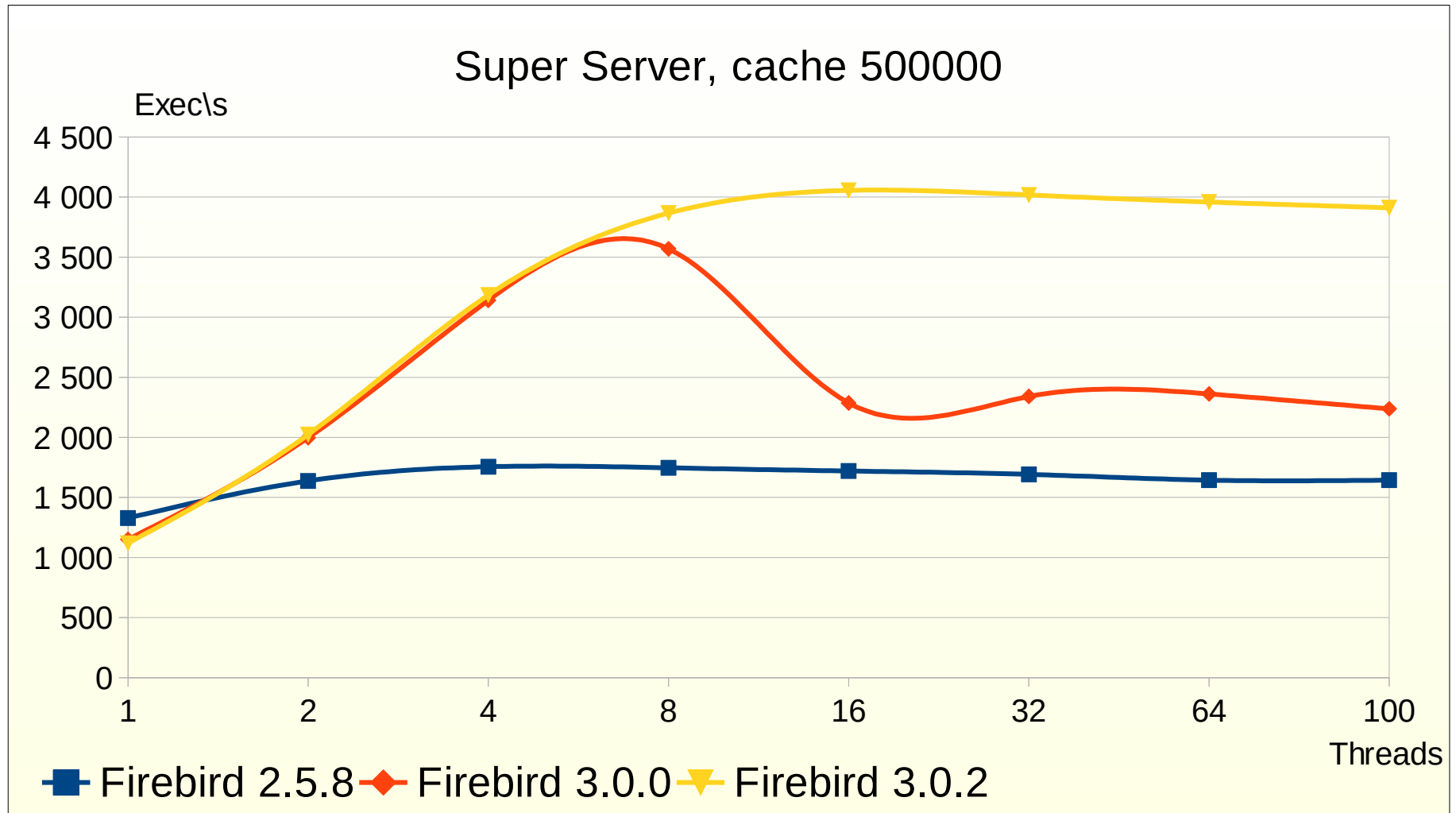
- Different versions, same mode





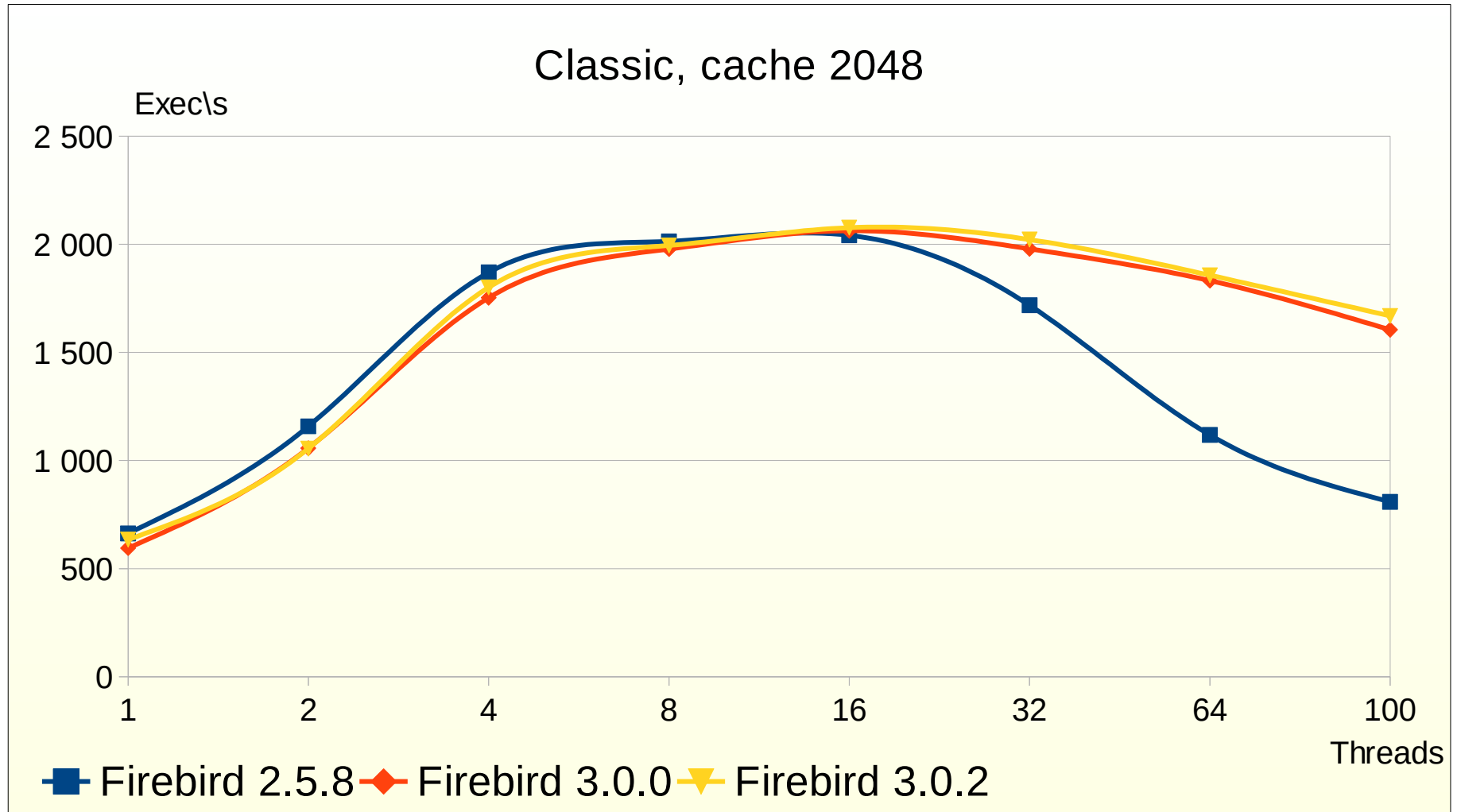
# Read-only, multi-threaded

- Different versions, same mode



# Read-only, multi-threaded

- Different versions, same mode



# Read-only, multi-threaded

- Super Classic 2.5.8 scales well up to 8 threads, keeps the performance at 16 threads and then performance drops
- Classic v3 scales to the almost same level as 2.5.8 but degrades in less degree
- Super v3 with small and medium cache run a bit worse than Classic v3
- Super 3.0.0 with huge cache scales up to 8 threads, then performance drops significantly
- Super 3.0.2 with huge cache scales well up to 8 threads, slightly adds at 16 threads and then performance drops less than by 4% (for 100 threads)
- Super 3.0.2 with huge cache works two times faster than Classic



# TPCC benchmark

- TPCC database
  - Generated by TPCC load tool
    - 100 warehouses
    - Standard scaling factor
  - Physical parameters
    - 8 KB page size
    - 8.75 GB
    - Forced Writes = ON
  - Restored from the file copy before each test run



- To be continued — full version of presentations from Firebird Tour 2017 will be published in November 2017
- Any questions? [ak@ib-aid.com](mailto:ak@ib-aid.com)

