



# Advanced replication in Firebird 4 and beyond

Dmitry Yemanov  
[dimitr@firebirdsql.org](mailto:dimitr@firebirdsql.org)

Firebird Project  
[www.firebirdsql.org](http://www.firebirdsql.org)



# Firebird Conference 2019

## Berlin, 17-19 October



YOUR PREMIER SOURCE OF FIREBIRD SUPPORT

# IBSurgeon



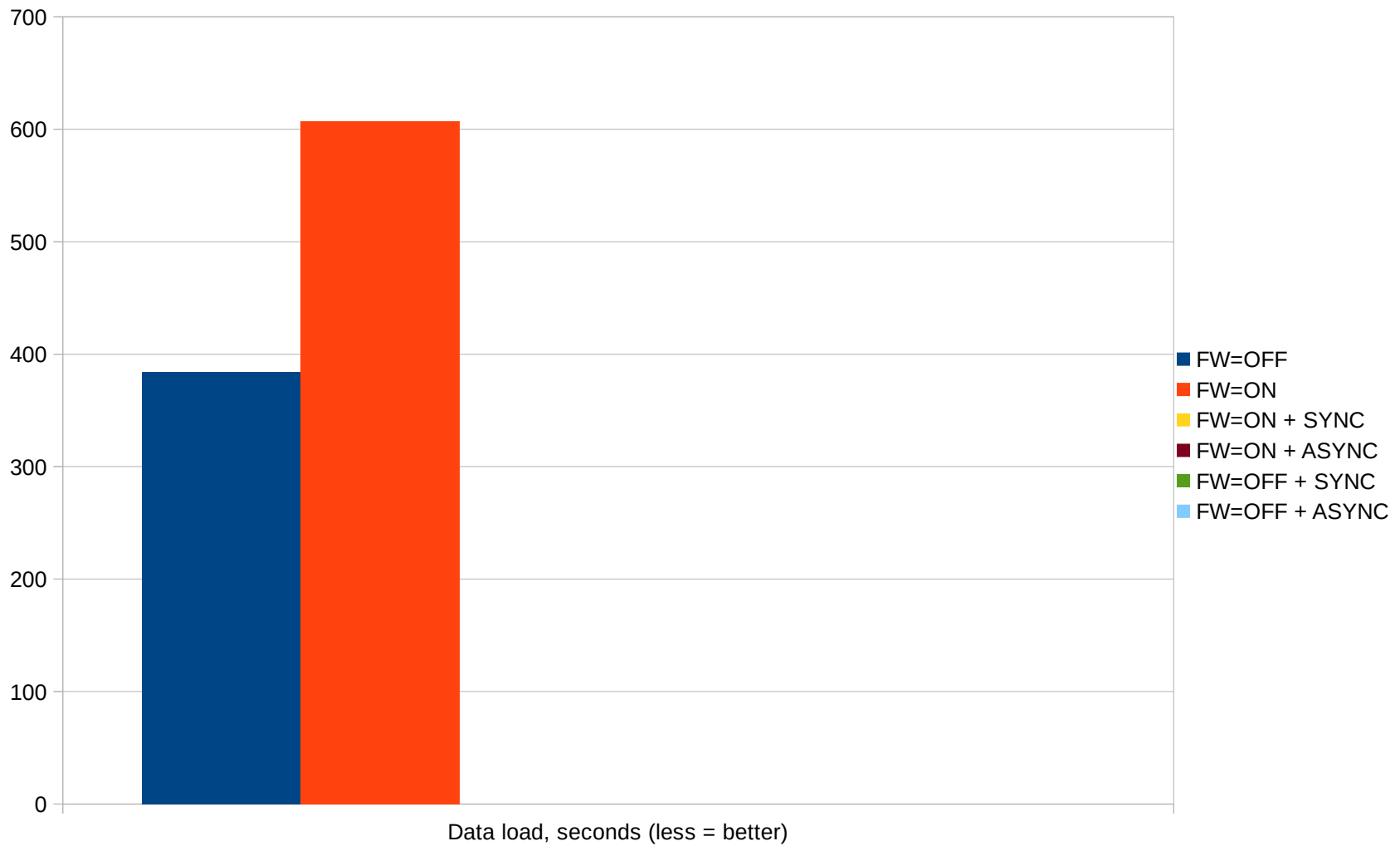
**MOSCOW  
EXCHANGE**



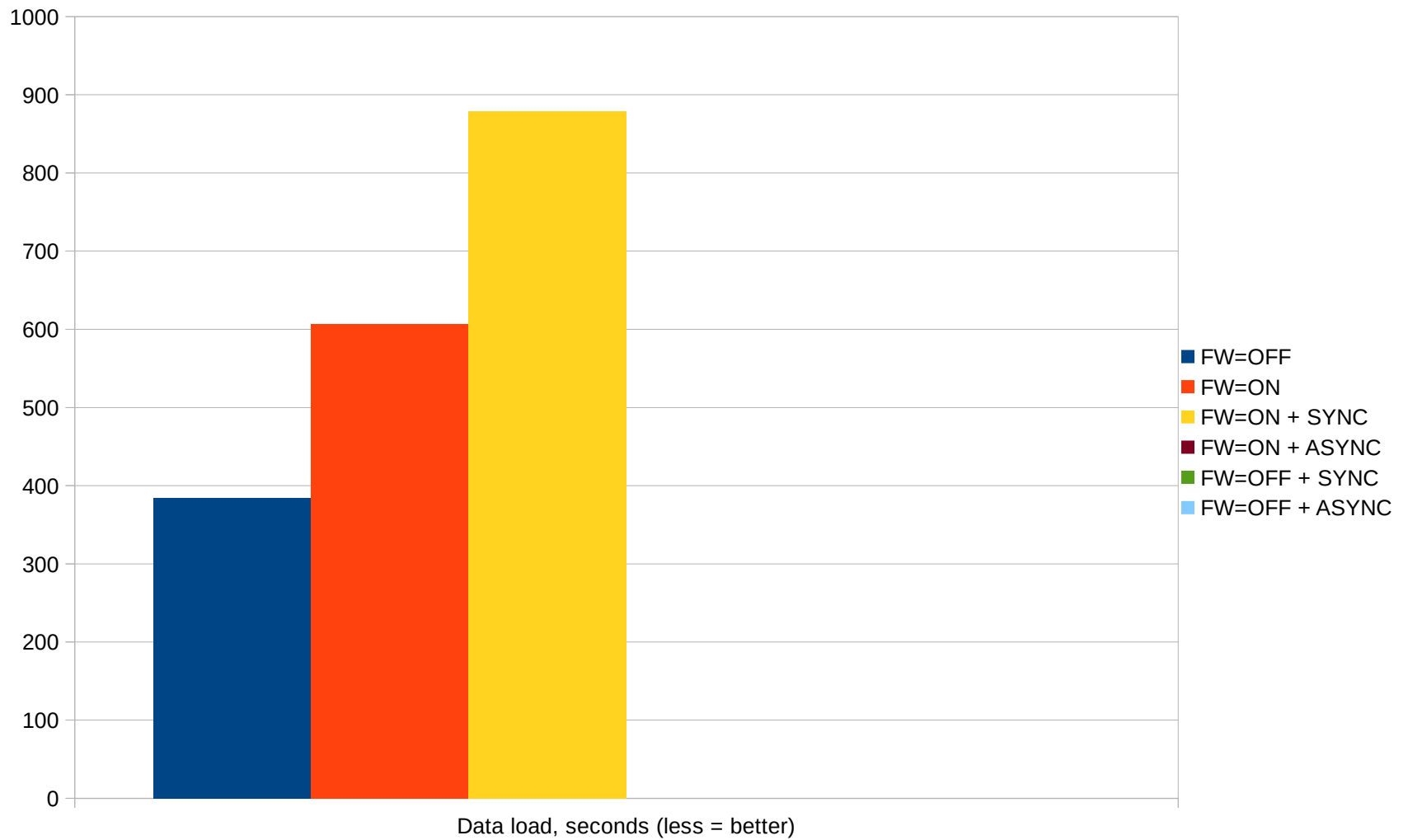
**Fast Reports**  
Reporting must be fast!



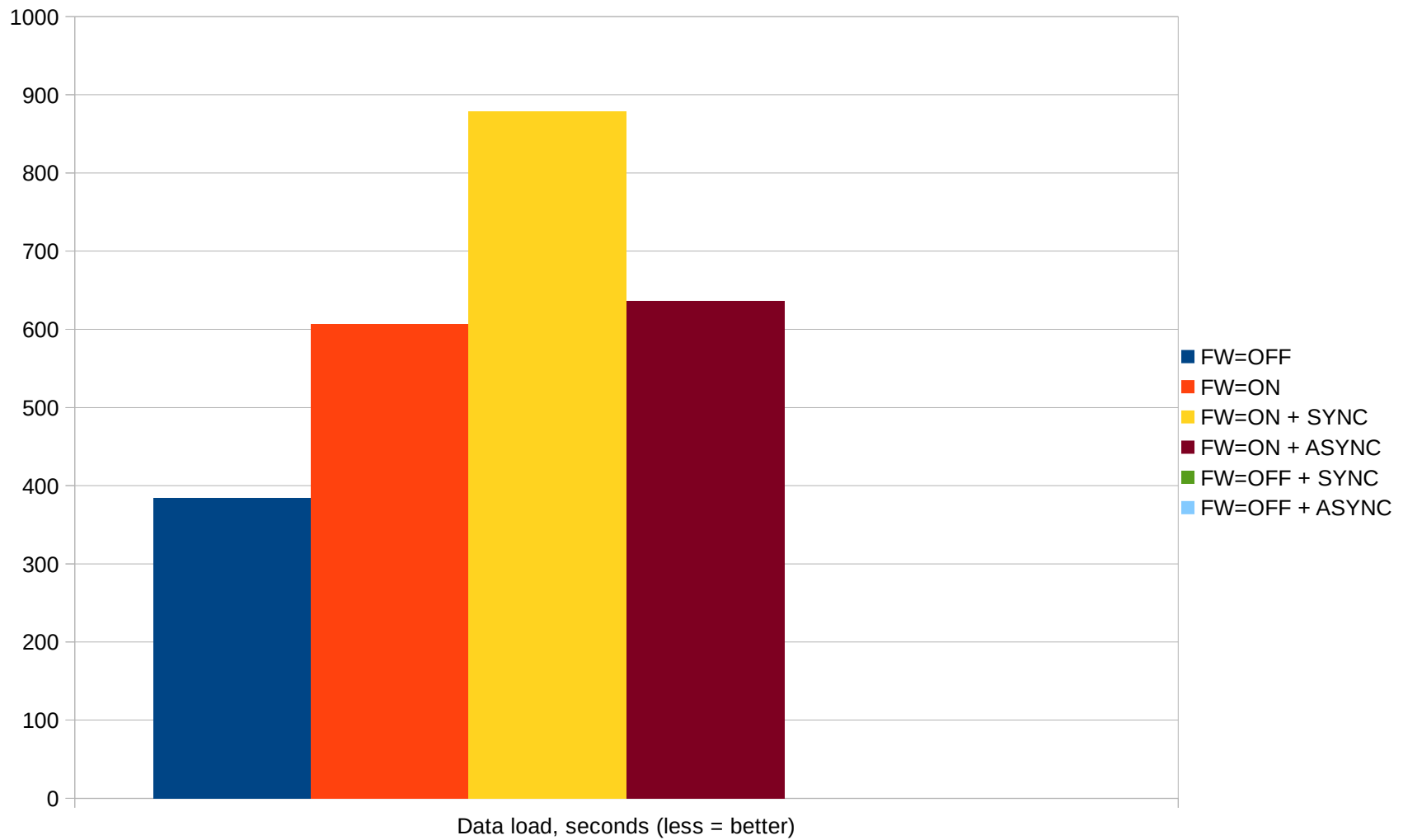
# Replication cost



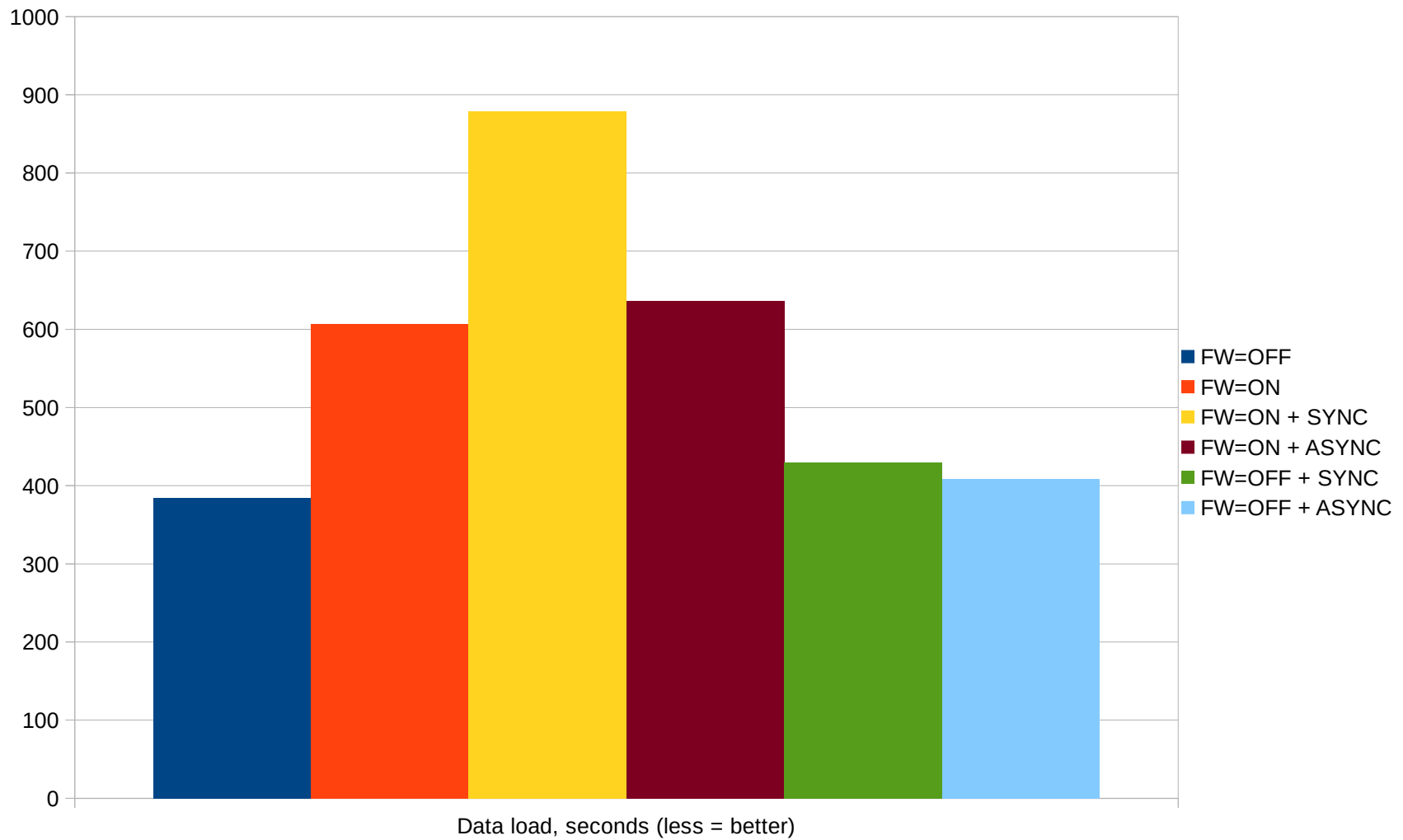
# Replication cost



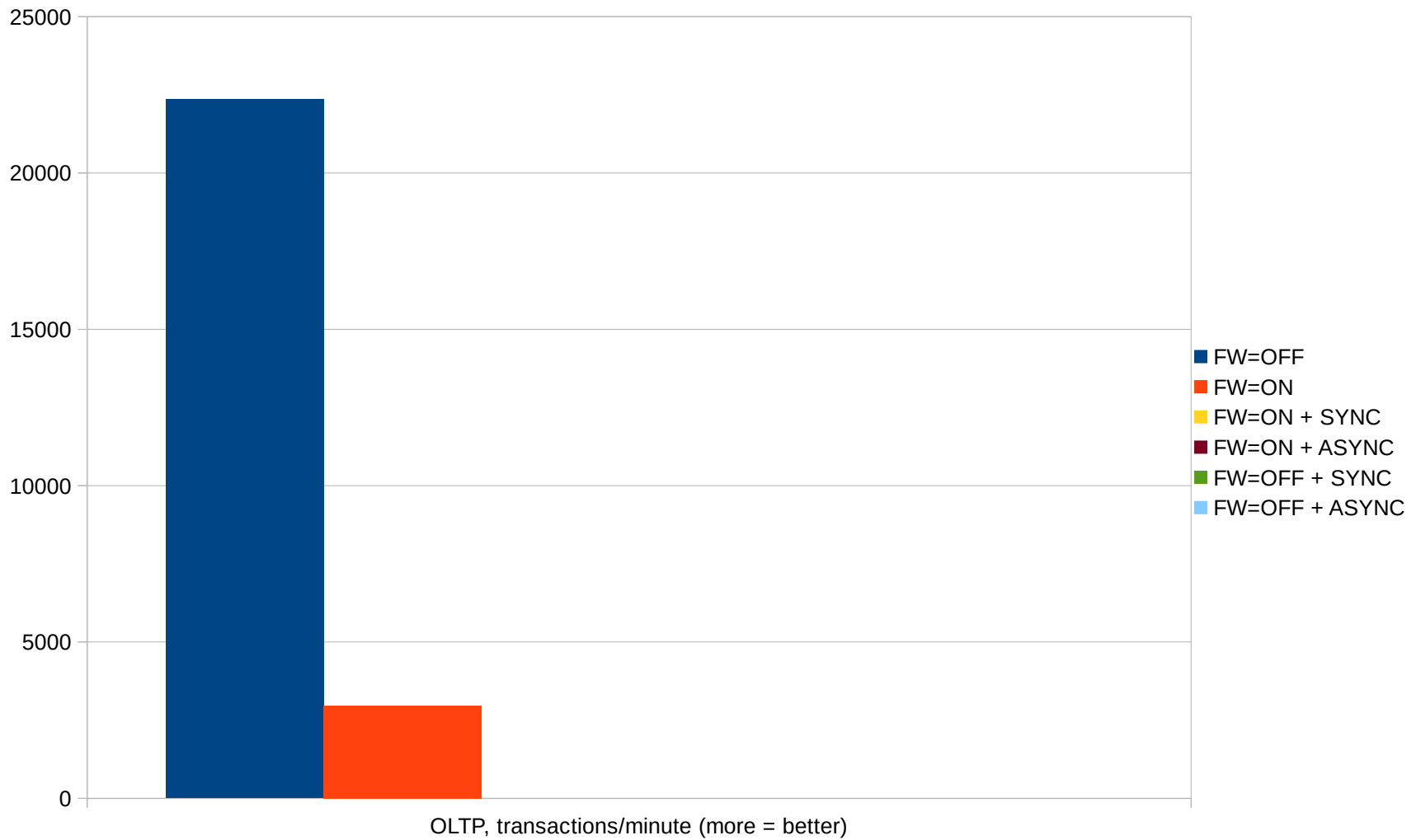
# Replication cost



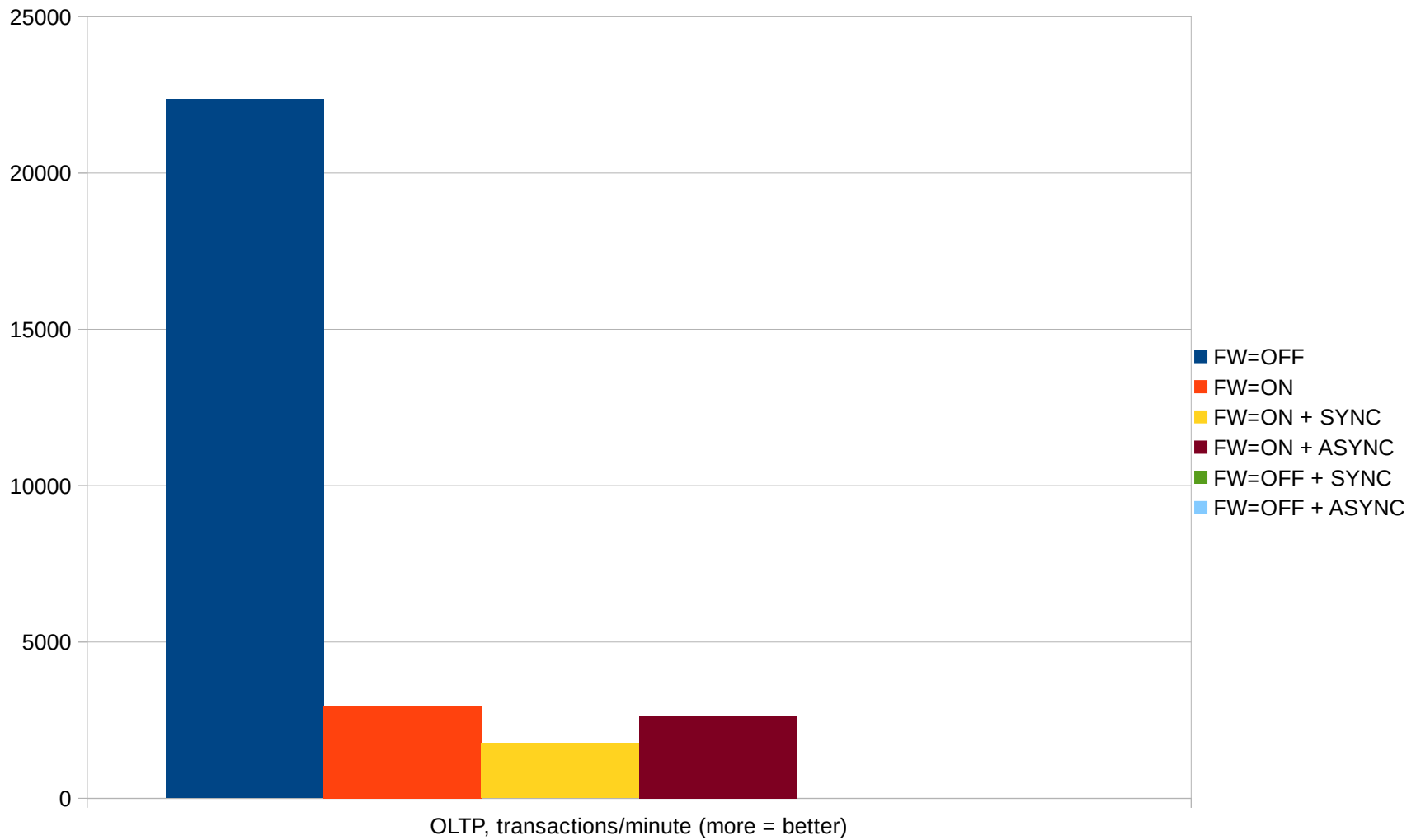
# Replication cost



# Replication cost

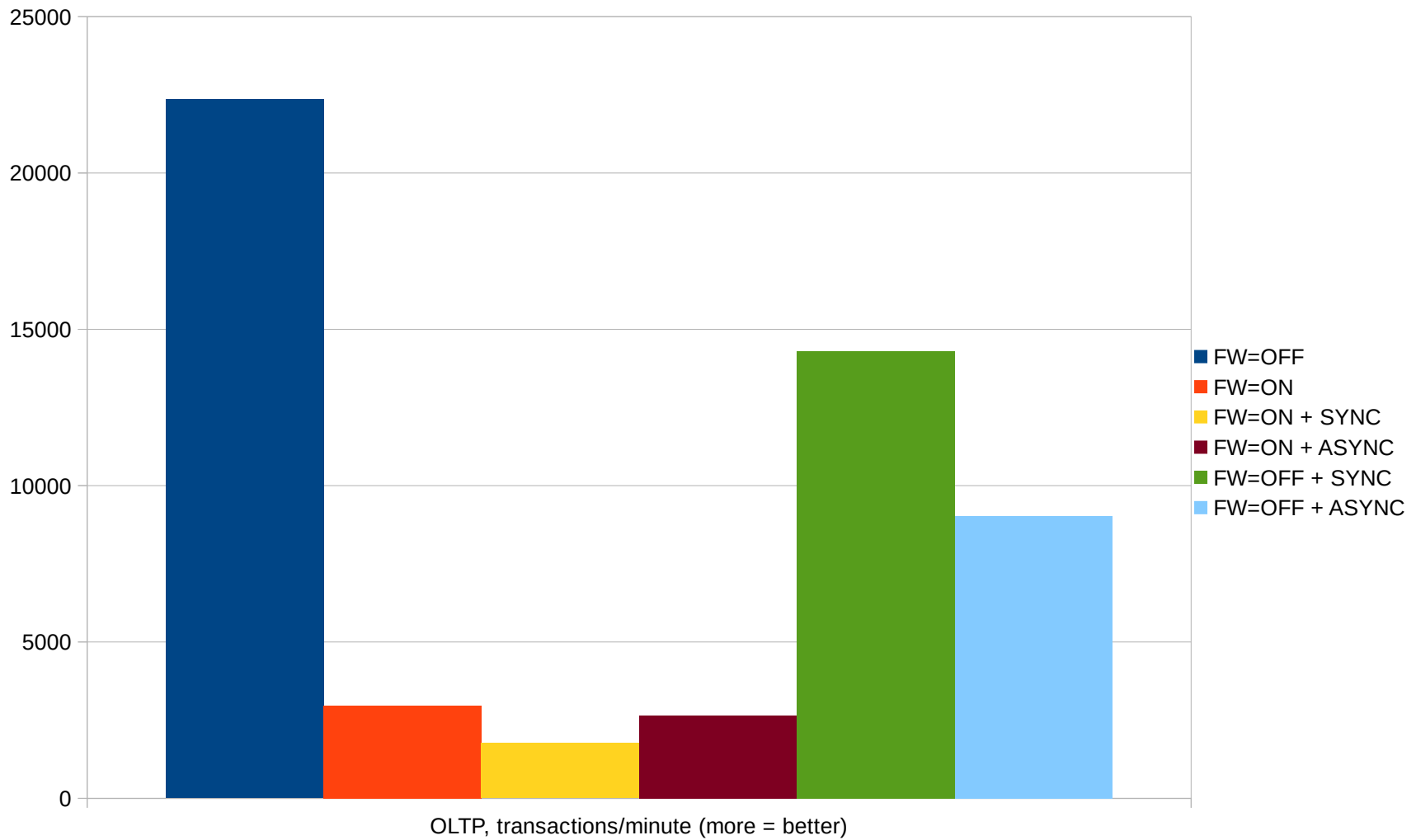


# Replication cost





# Replication cost



## Speeding up performance

---

### Old trick with shadows

- FW = ON on the shadow side is enough

## Speeding up performance

---

### Old trick with shadows

- FW = ON on the shadow side is enough

### The same trick with replication

- FW = OFF on the primary side
- FW = ON on the replica side
- Set up sync / async replication
- Double your disk space ;-)

## Read-only replica

---

### Applications

- Just a failover (very limited RO activity)
- Long-running reports
- Backup

## Read-write replica

---

### Applications

- Reports that modify data  
(emulated temporary tables)
- Merge changes from the other database  
(head office → branch or vice versa)

## Read-write replica

---

### Applications

- Reports that modify data (emulated temporary tables)
- Merge changes from the other database (head office → branch or vice versa)

### How to avoid conflicts

- Exclude «temporary» tables from replication set
- Make changes non-intersecting
- Ensure global ID scheme

## Mixed sync / async replication

---

### Multi-level recovery guarantee

- Sync replica(s) are for immediate recovery
- Async journal as a backup recovery option
- Minimized downtime

## Mixed sync / async replication

---

### Multi-level recovery guarantee

- Sync replica(s) are for immediate recovery
- Async journal as a backup recovery option
- Minimized downtime

### Other applications

- Use journal for PITR
- Use journal for audit



## Semi-sync replication

---

### Concept

- Relax synchronization requirements and thus improve performance
- Still be able to recover

## Semi-sync replication

---

### Concept

- Relax synchronization requirements and thus improve performance
- Still be able to recover

### Possible options

- Changes are received by the replica host
- Changes are received by the replica host and applied
- Changes are received by the replica host and applied and committed
- Quorum threshold

## Multi-source replication

---

### Concept

- Replica may receive changes from multiple primary databases
- Database UUID to separate sources
- Limit the replica to deal with particular sources

### Application

- Merge branches to the head office

## Multi-source replication

---

### Usage

- Set up multiple source directories
- Set up source filters for the replica
- Ensure non-intersecting operations or unique ID scheme

## Cascaded replication

---

### Concept

- Replica can also act as primary
- Applied changes are replicated further
- May also work for read-write replica
- Can be set up with a single entry in replication.conf

## Cascaded replication

---

### Concept

- Replica can also act as primary
- Applied changes are replicated further
- May also work for read-write replica
- Can be set up with a single entry in replication.conf

### Tricks

- What about loops?
- $A \rightarrow B \rightarrow C \rightarrow A$
- $A \rightarrow B \rightarrow A$

## Bi-directional replication

---

### Concept

- Not the same as «looped» cascaded replication
- Use read-write mode for replica
- Global record identification  
(UUID key, range-based key, composite key)
- Rules for conflict resolution

## Bi-directional replication

---

### Concept

- Not the same as «looped» cascaded replication
- Use read-write mode for replica
- Global record identification  
(UUID key, range-based key, composite key)
- Rules for conflict resolution

### Problems

- What to do with sequences
- Delayed conflicts and manual resolutions



## Simplifying migration

---

### Upgrading

- Use production FB / database as primary
- Set up async replication
- Backup and restore on new FB version
- Set up restored database as replica
- Wait until it catches up the primary
- Test the replica
- Once ready, promote replica to primary

## Simplifying migration

---

### Downgrading

- Use upgraded FB / database as primary, older FB / database as replica
- Set up async OR sync replication
- Switch to the older FB / database if something goes seriously wrong

## Creating custom replicators

---

### Concept

- Public CDC interface
- Plugin-based architecture
- Built-in replication is just a built-in plugin
- Public API to apply replication packets

## Creating custom replicators

---

### Concept

- Public CDC interface
- Plugin-based architecture
- Built-in replication is just a built-in plugin
- Public API to apply replication packets

### Applications

- Alternative journalling or transport options
- Publishing data to other products (e.g. OLAP)
- Row-level audit



Questions?