

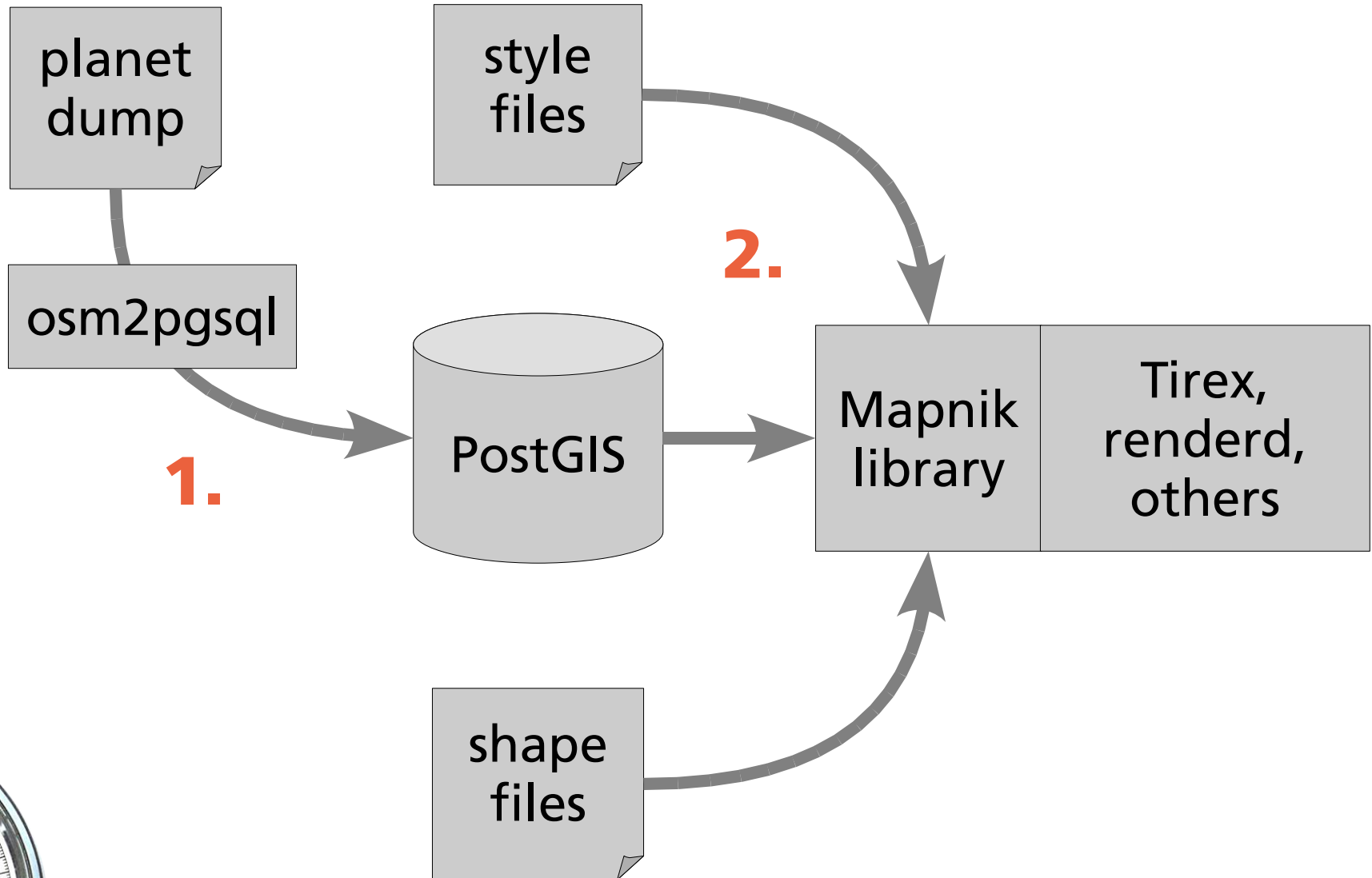
# Optimising the Mapnik Rendering Toolchain

Frederik Ramm  
Geofabrik GmbH

or: Things you could have found  
out yourself if only it didn't  
take so damn long to try them!



# The Rendering Toolchain



1.



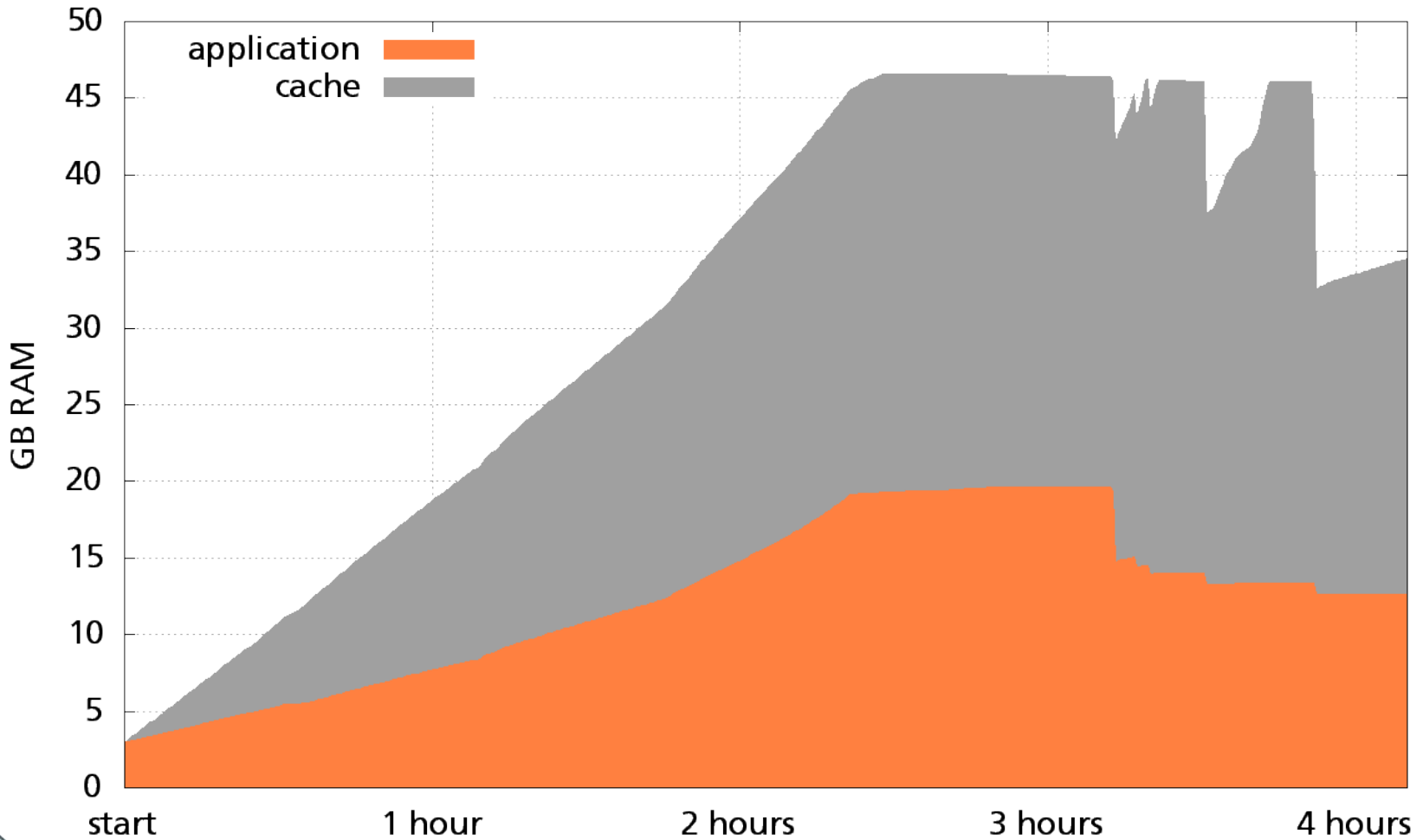
# Basic Setup

- Dual Quad-Core Xeon 2.4 GHz Machine, 48 GB RAM
- Ubuntu Linux
- Mapnik 0.7
- gzipped, full recent planet file
- different PostgreSQL/PostGIS variants (mostly 8.3)
- Areca RAID controller
- WD Raptor 10k RPM disks,  
standard Samsung SATA disks  
& a SuperTalent 128 GB SSD drive



# Full Planet Import

Memory usage during Full Planet Import

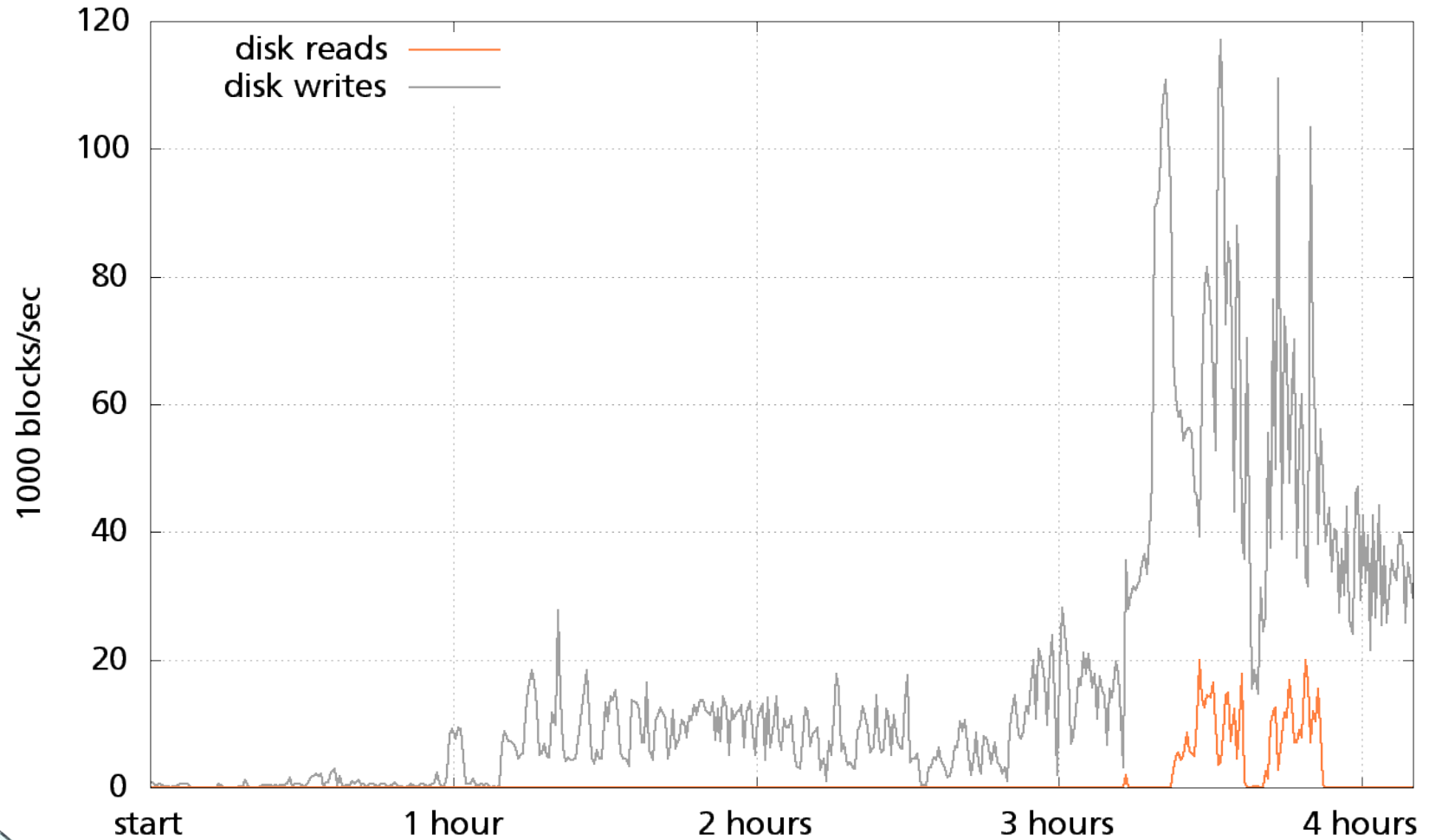


Standard SATA Disk, semi-tuned PostgreSQL 8.3



# Full Planet Import

Disk usage during Full Planet Import

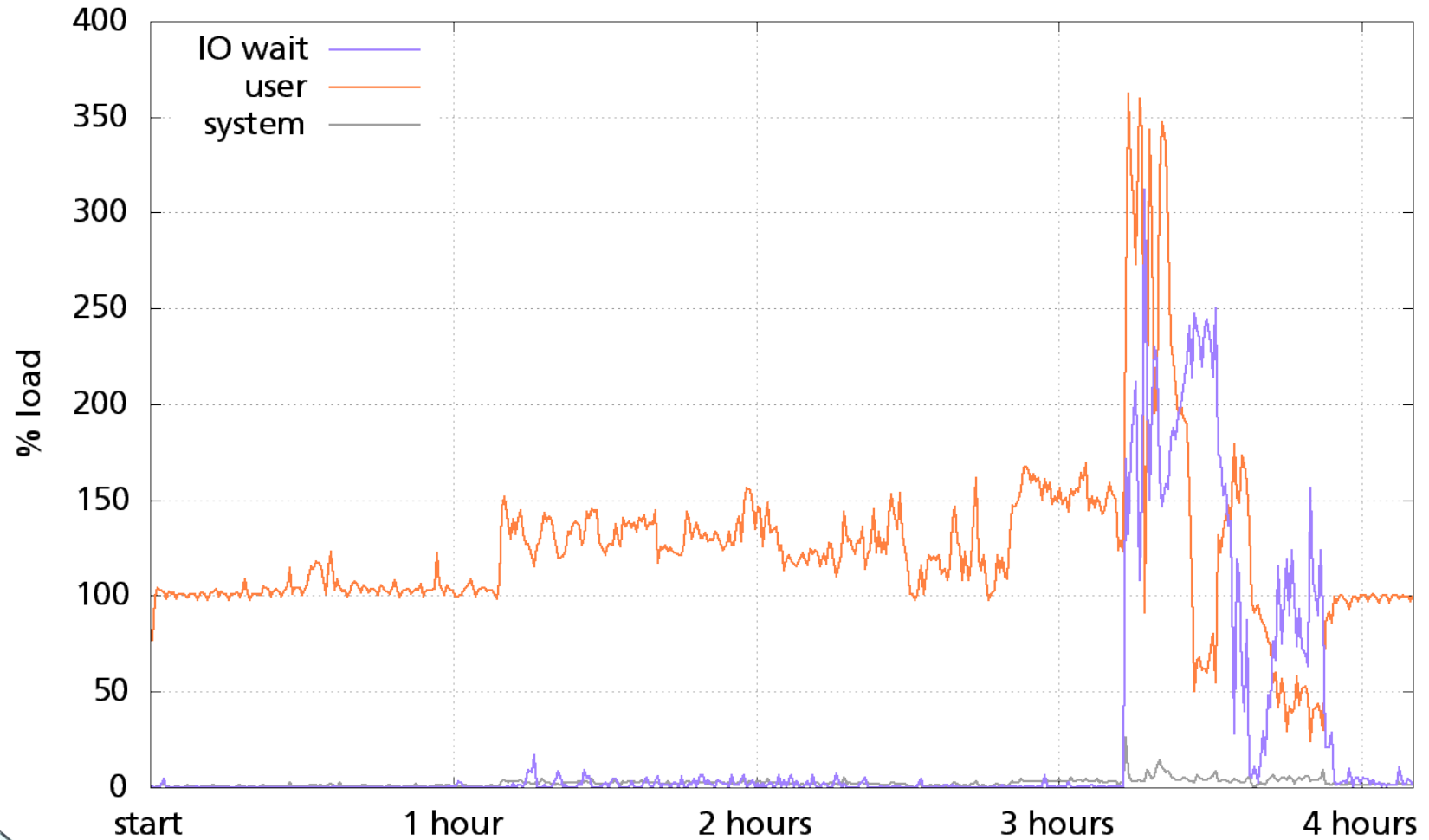


Standard SATA Disk, semi-tuned PostgreSQL 8.3



# Full Planet Import

CPU usage during Full Planet Import



Standard SATA Disk, semi-tuned PostgreSQL 8.3



# PostgreSQL Tuning for N00bs

/etc/postgres/8.3/main/postgresql.conf:

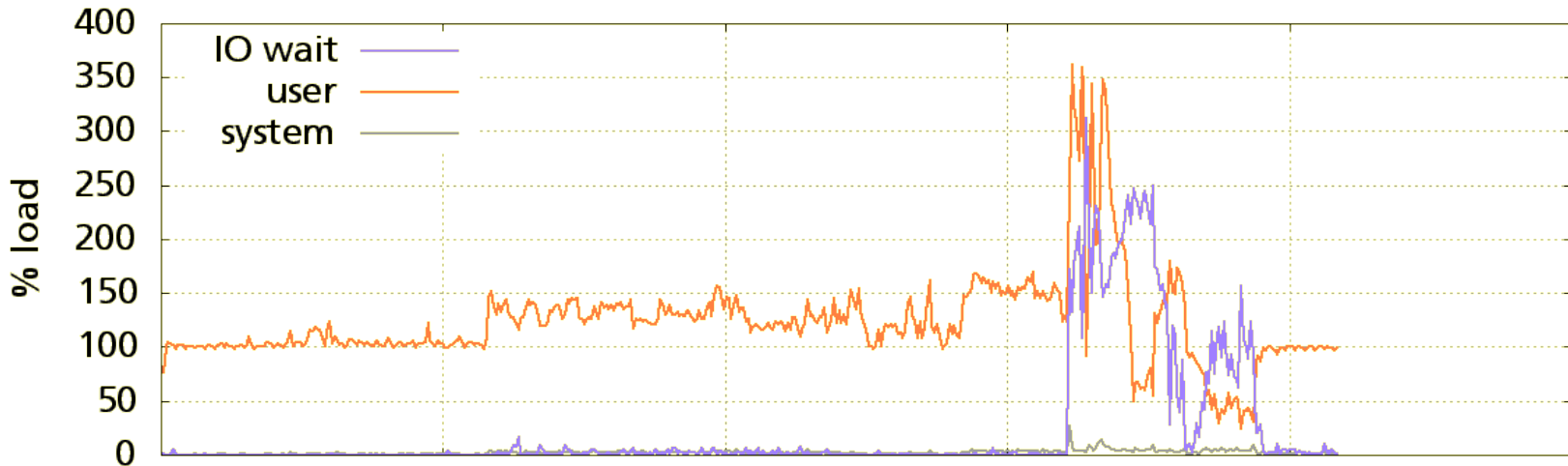
option	default	recommended
shared_buffers	24 MB	768 MB
work_mem	1 MB	512 MB
maintenance_work_mem	16 MB	512 MB
max_fsm_pages	153600	204800
fsync	on	off
autovacuum	on	?



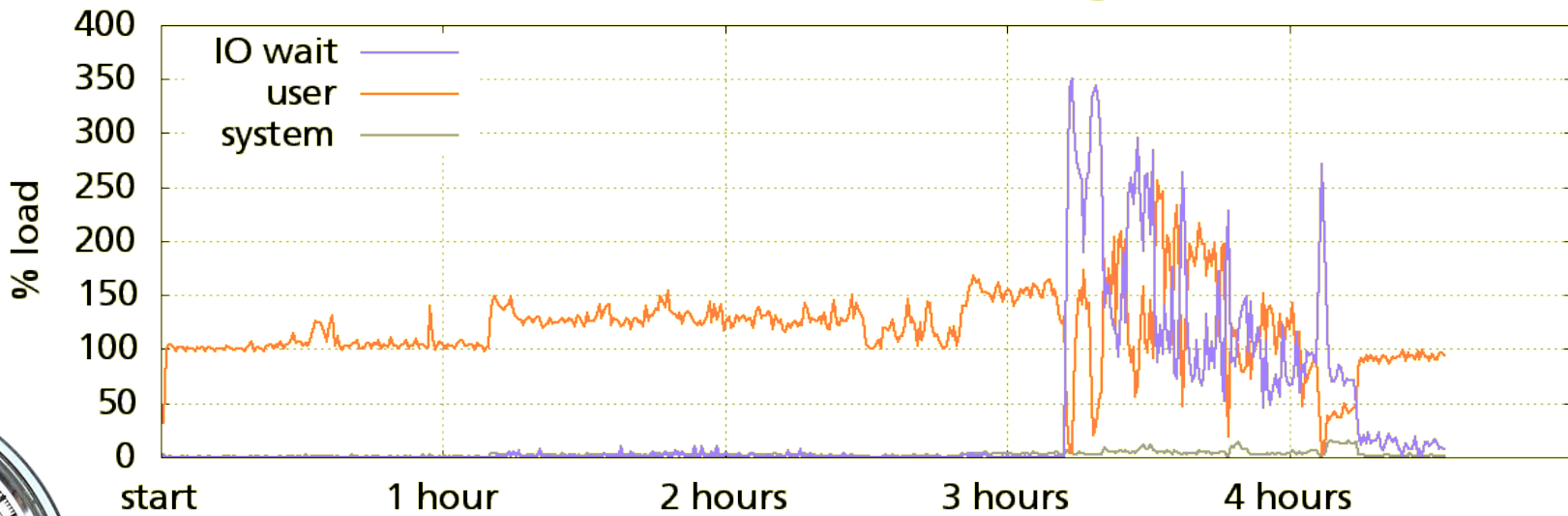


# Full Planet Import

CPU usage during Full Planet Import



Standard SATA Disk, semi-tuned PostgreSQL 8.3

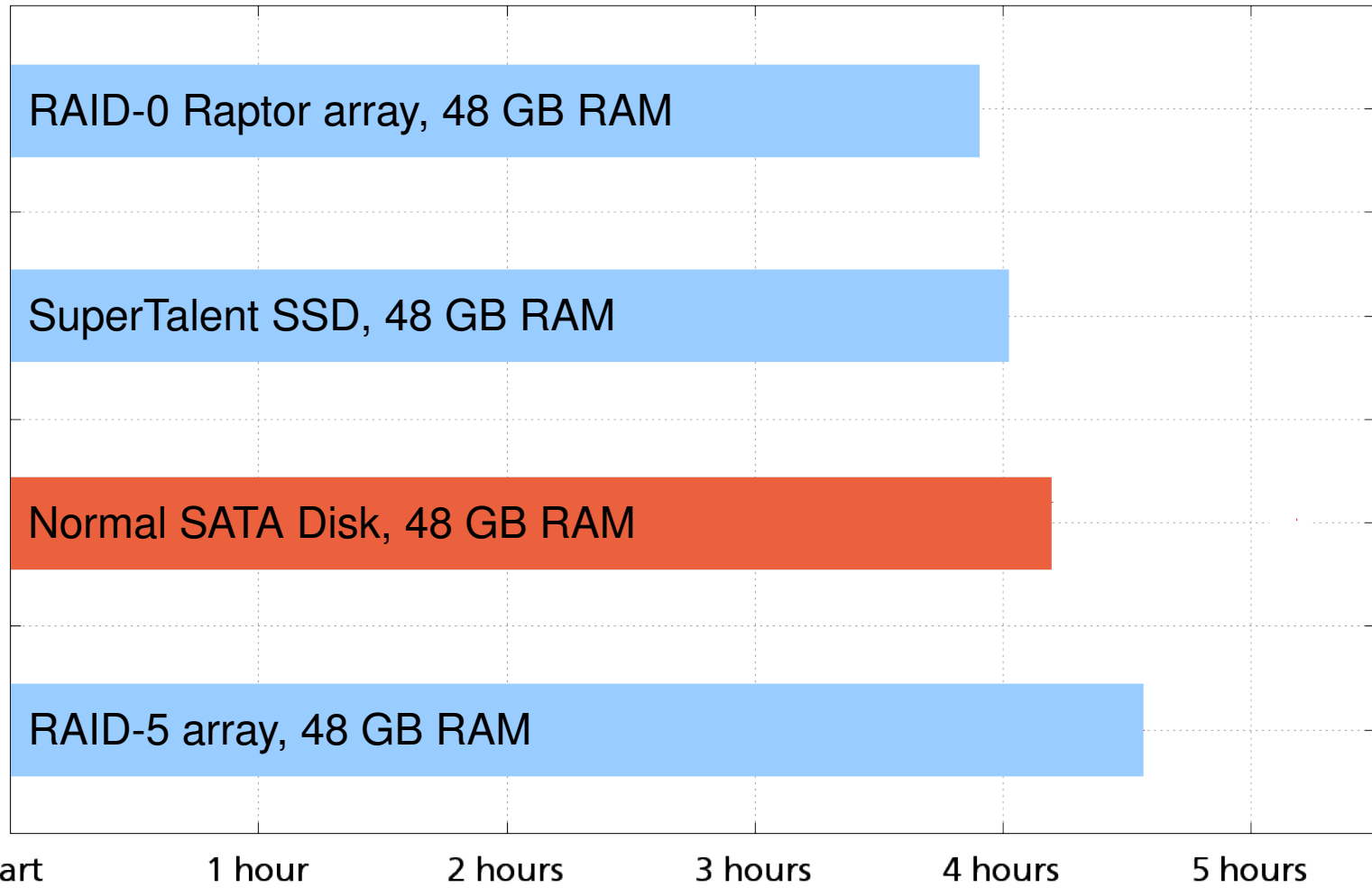


Standard SATA Disk, un-tuned PostgreSQL 8.3



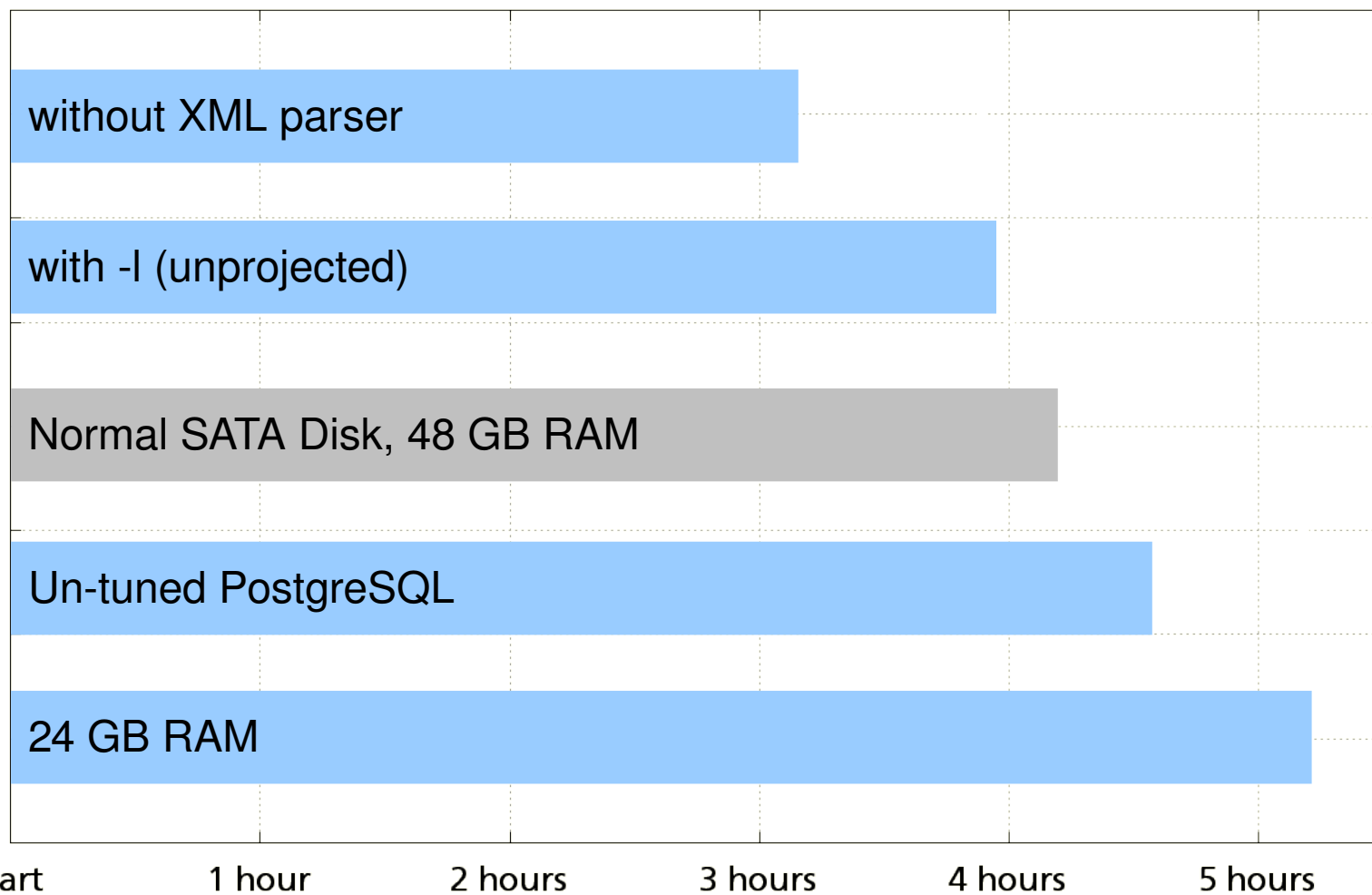
# Full Planet Import

Total Time for Full Planet Import



# Full Planet Import

Total Time for Full Planet Import



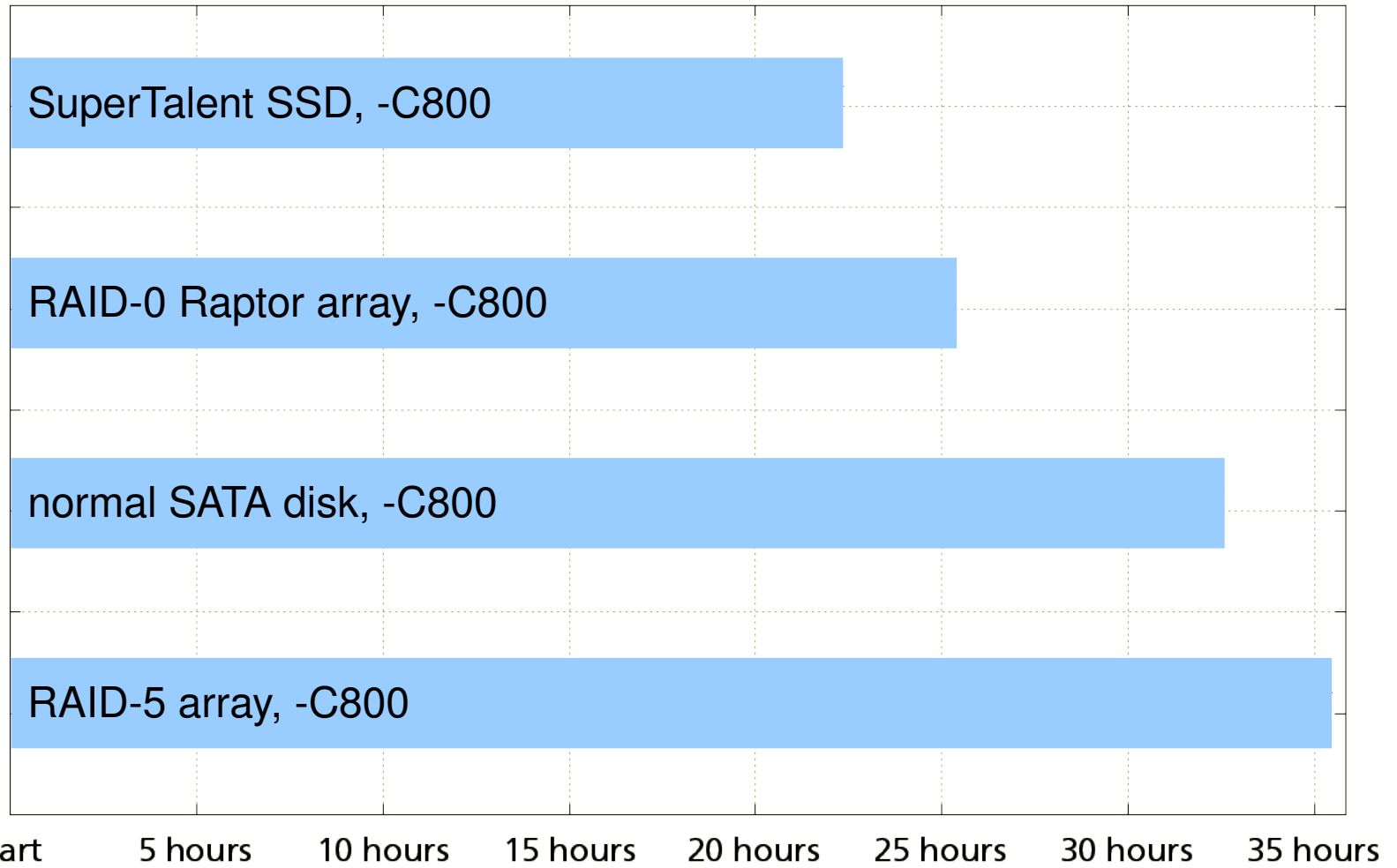
# Full Planet Import – Results

- fast disks are good
- but a lot of memory is even better
- RAID-0 is faster, RAID-5 slower than plain disk
- SSD doesn't make a big difference
- should be possible to shave off ~ 2 hours (50%)  
by making osm2pgsql's node/way processing  
multi-threaded and/or improve XML parsing



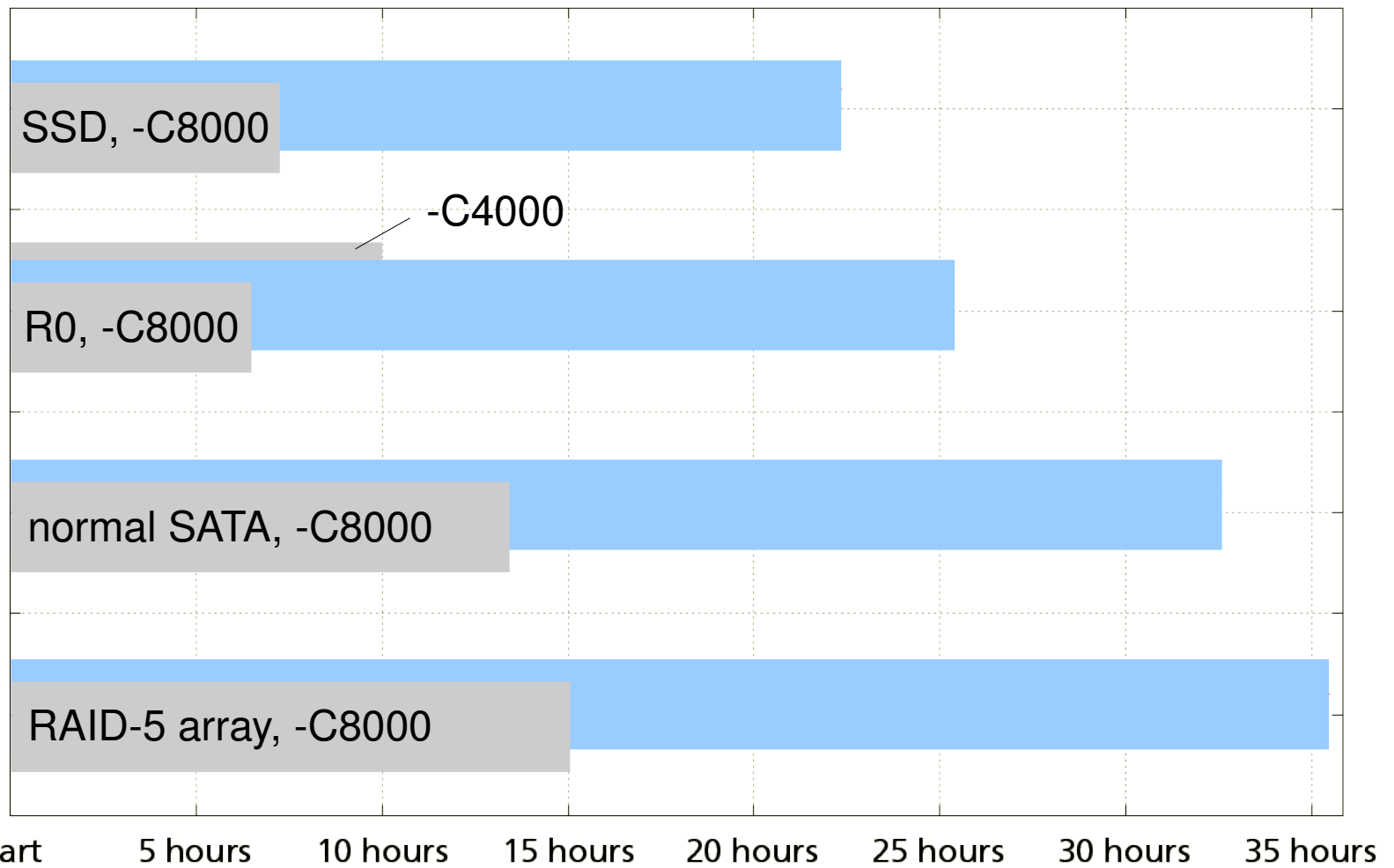
# Slim Planet Import

Time for Slim Planet Import



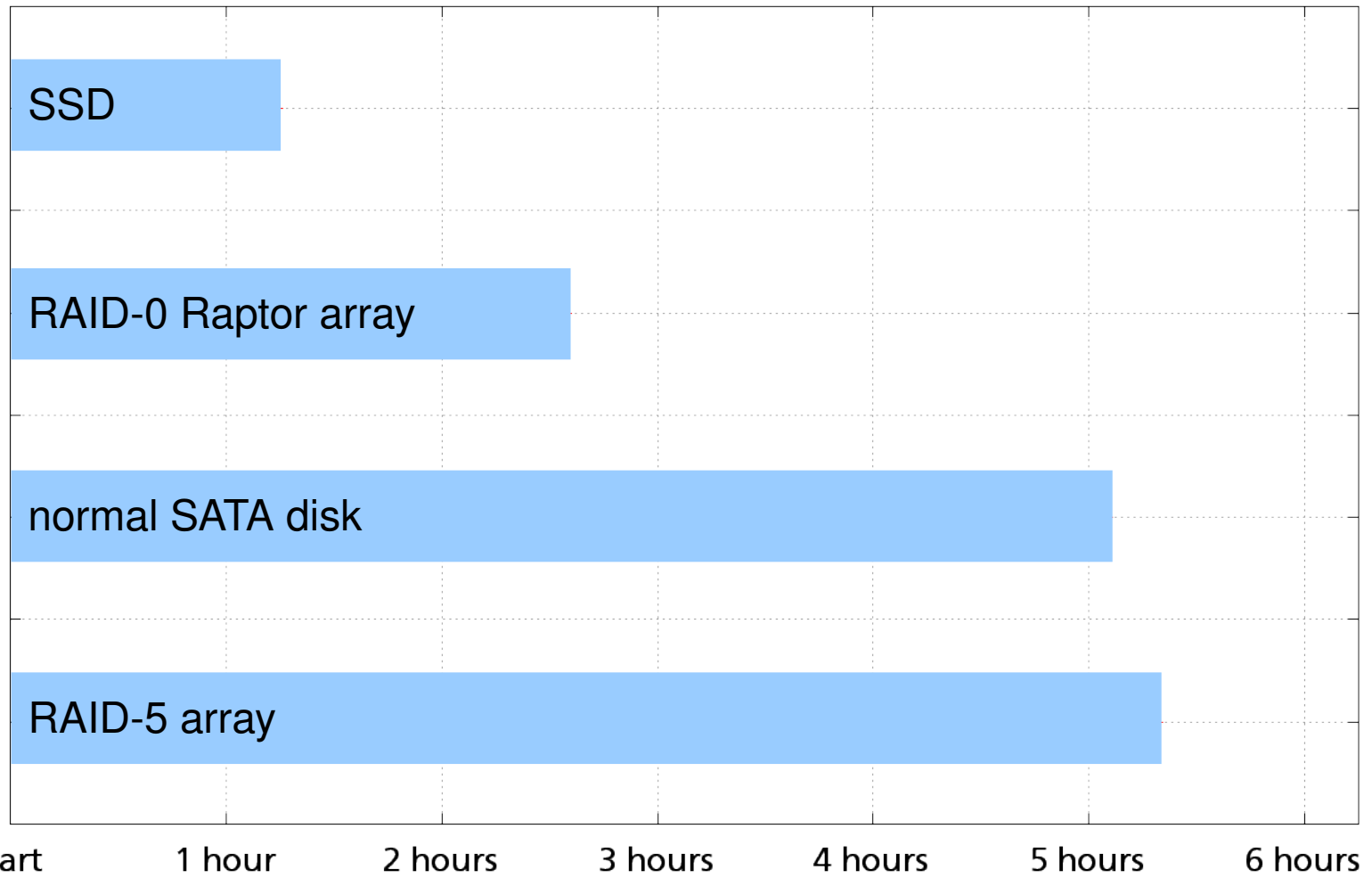
# Slim Planet Import

Time for Slim Planet Import



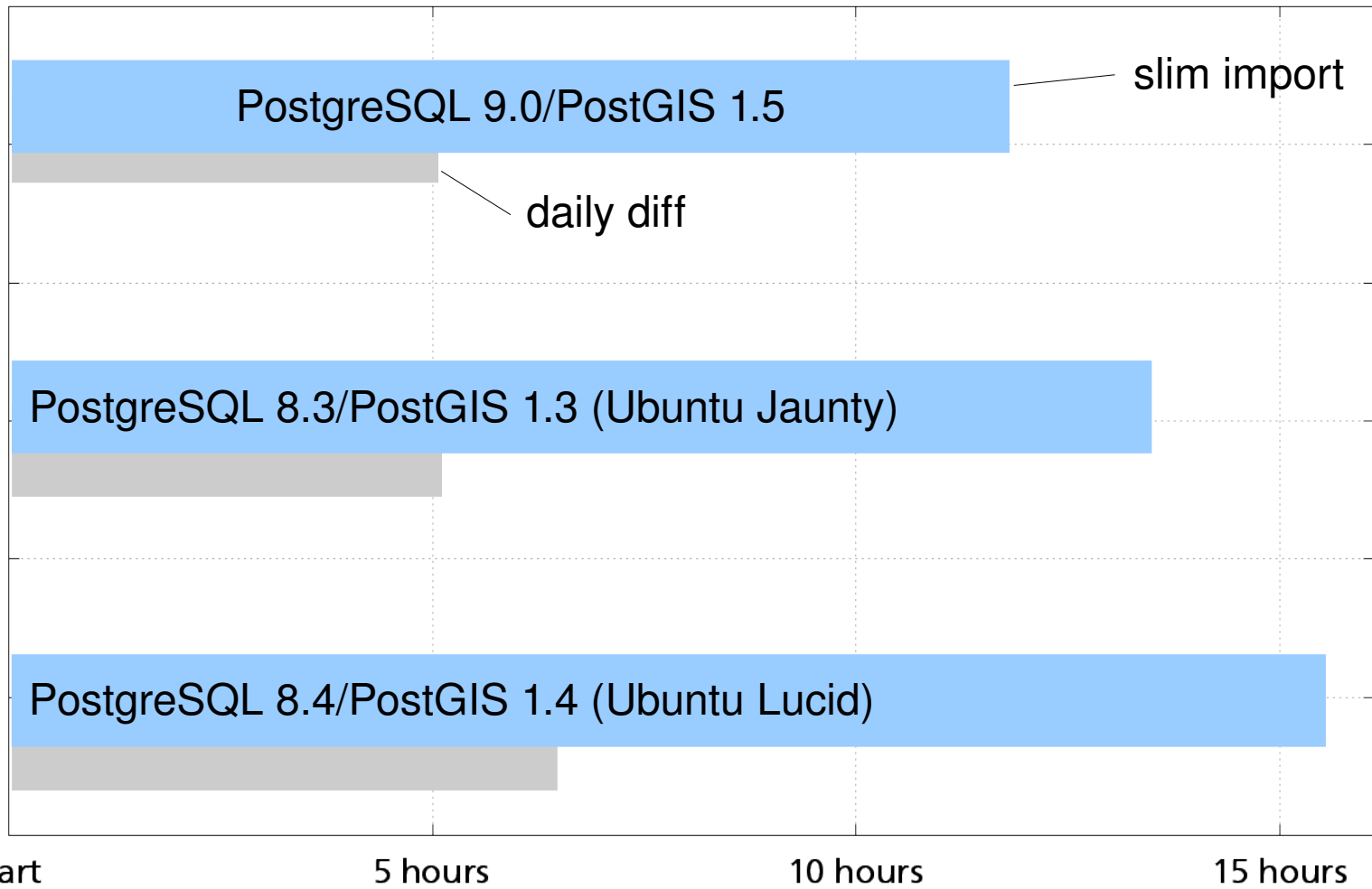
# Slim Mode Diff Application

Time for Applying Daily Diff



# PostgreSQL 8.4 and 9.0

Slim Import and Diff Application with -C8000 on normal SATA





# Slim Mode Import – Results

- fast disks are good
- but fast disks are worth little if -C is set too small – must be large enough to cache all nodes  
(highest node ID \* 8 bytes = ~ 6.5 GB currently)
- memory no factor on diff applications
- possible to do initial import on fat machine, then copy database
- do not use PostgreSQL 8.4



# 2.



# Render Test Cases

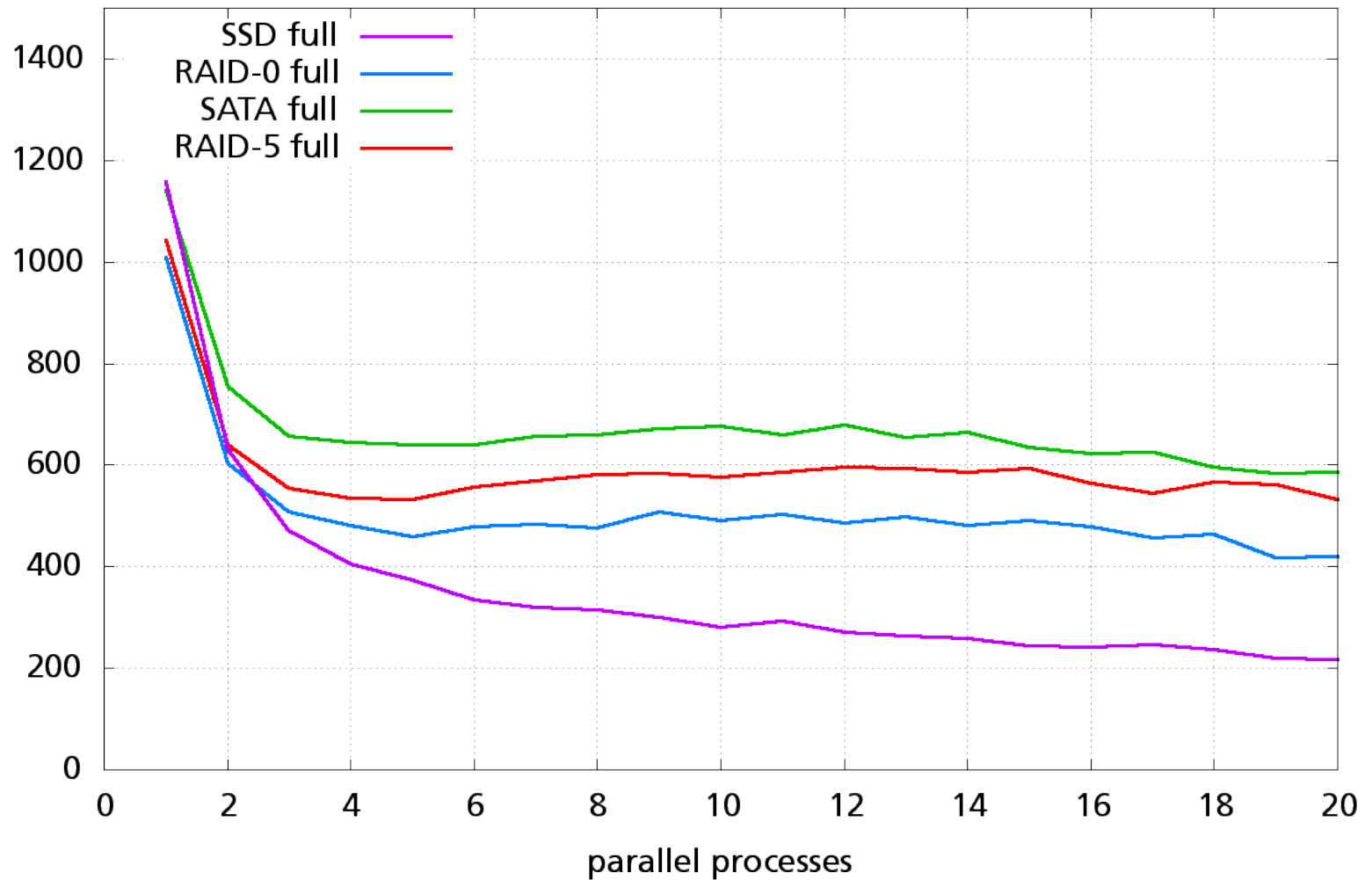
- “zoom-in” scenario
- “pan” scenarios on z12 and z16
- “random” tiles (770 meta tiles)

Results tended to be the same



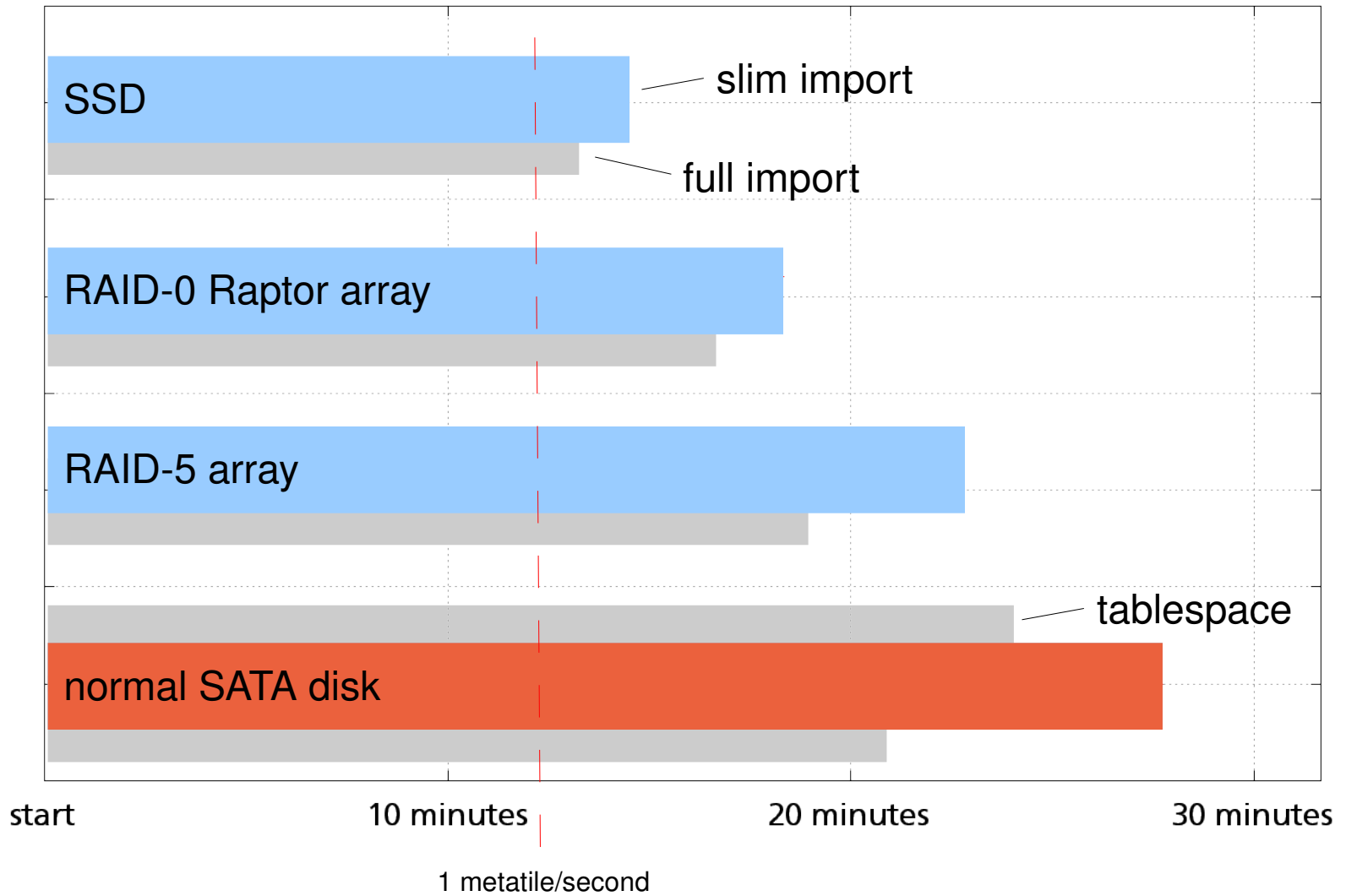
# Parallel Rendering Processes

Time to Render 50-Metatile Zoom-In Test Suite



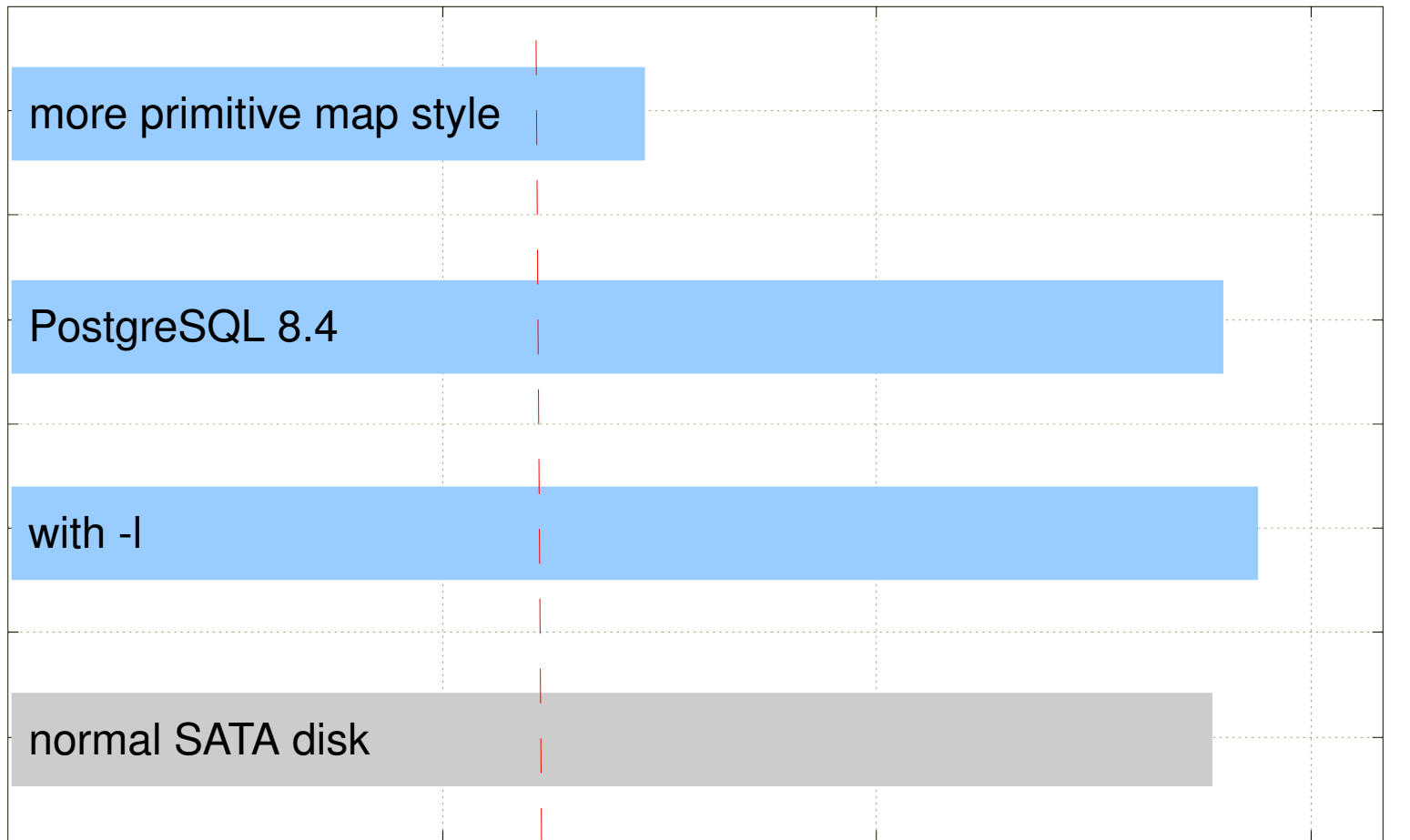
# Rendering Performance

Total Time for Random Render Test Case, 5 Parallel



# Rendering Performance

Total Time for Random Render Test Case, 5 Parallel



start

10 minutes

20 minutes

30 minutes

1 metatile/second



# Simplifying the Map Style

- use `analyze_postgis_log.pl` from [svn.openstreetmap.org/applications/utils/tirex/utils](http://svn.openstreetmap.org/applications/utils/tirex/utils)
- use simplified geometries for small zoom levels
- possibly: use bitmap layers for some types of areas (e.g. forests)



# Open Questions & Ideas

- different Mapnik versions
- special indexes in PostgreSQL
- different spindles for data, indexes etc
- planet import to RAM disk?
- vacuuming
- establish common test suite?





# Thank you

**Frederik Ramm**

**[frederik.ramm@geofabrik.de](mailto:frederik.ramm@geofabrik.de)**

