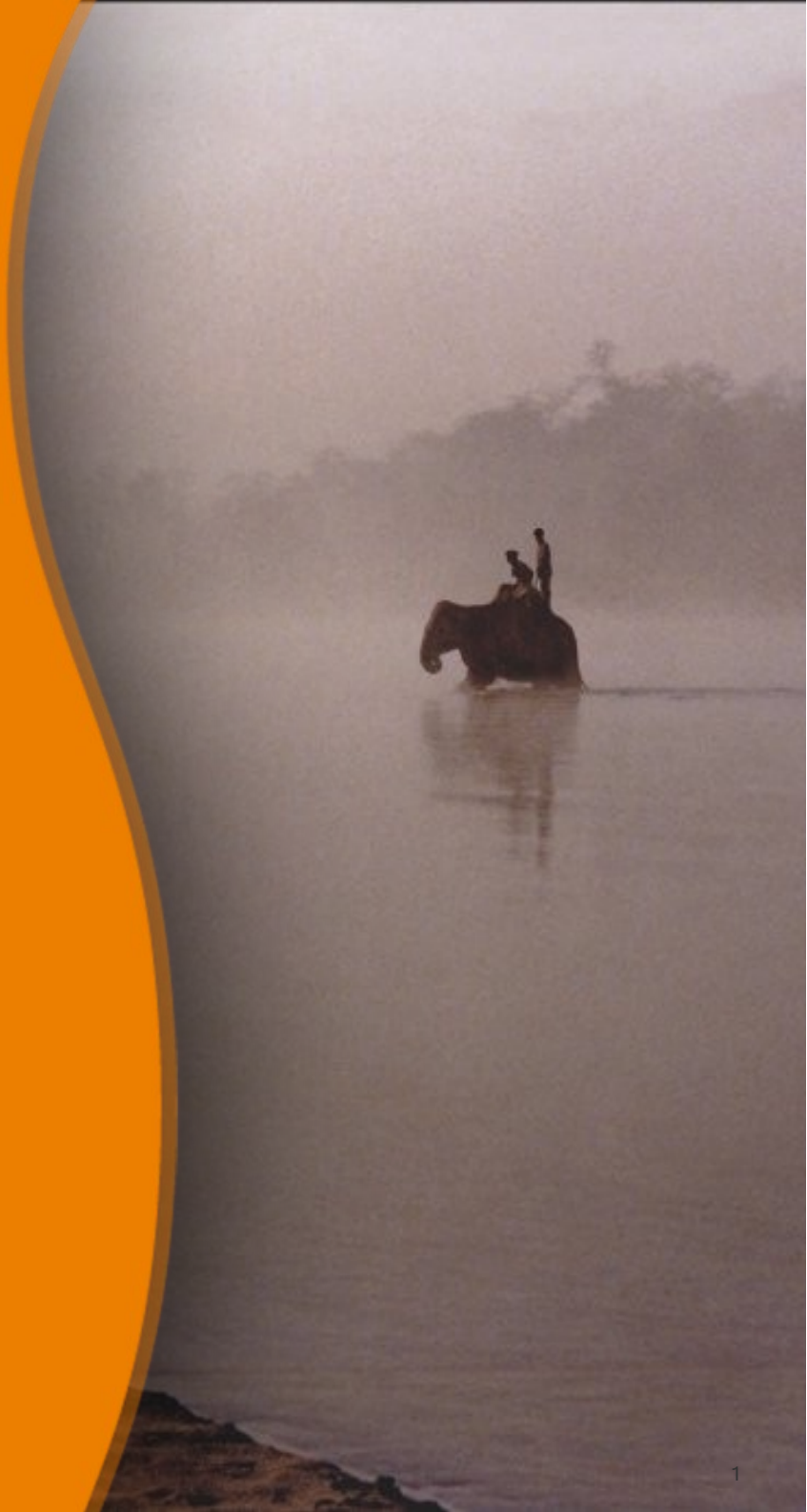


PostgreSQL tuning

Zdeněk Kotala
RPE
Sun Microsystems

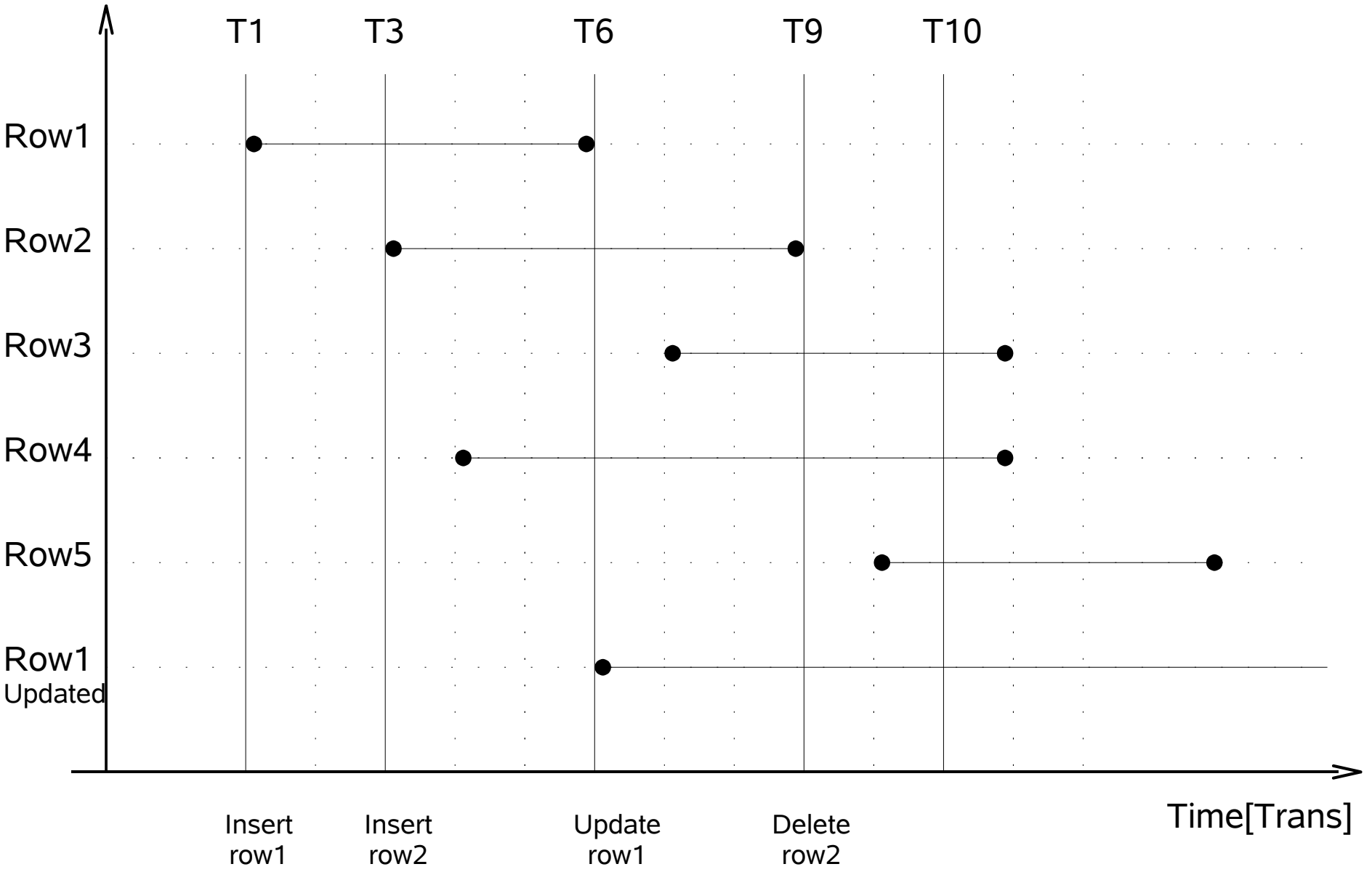
Agenda

- How postgresql works
- Configuration parameters
- Monitoring

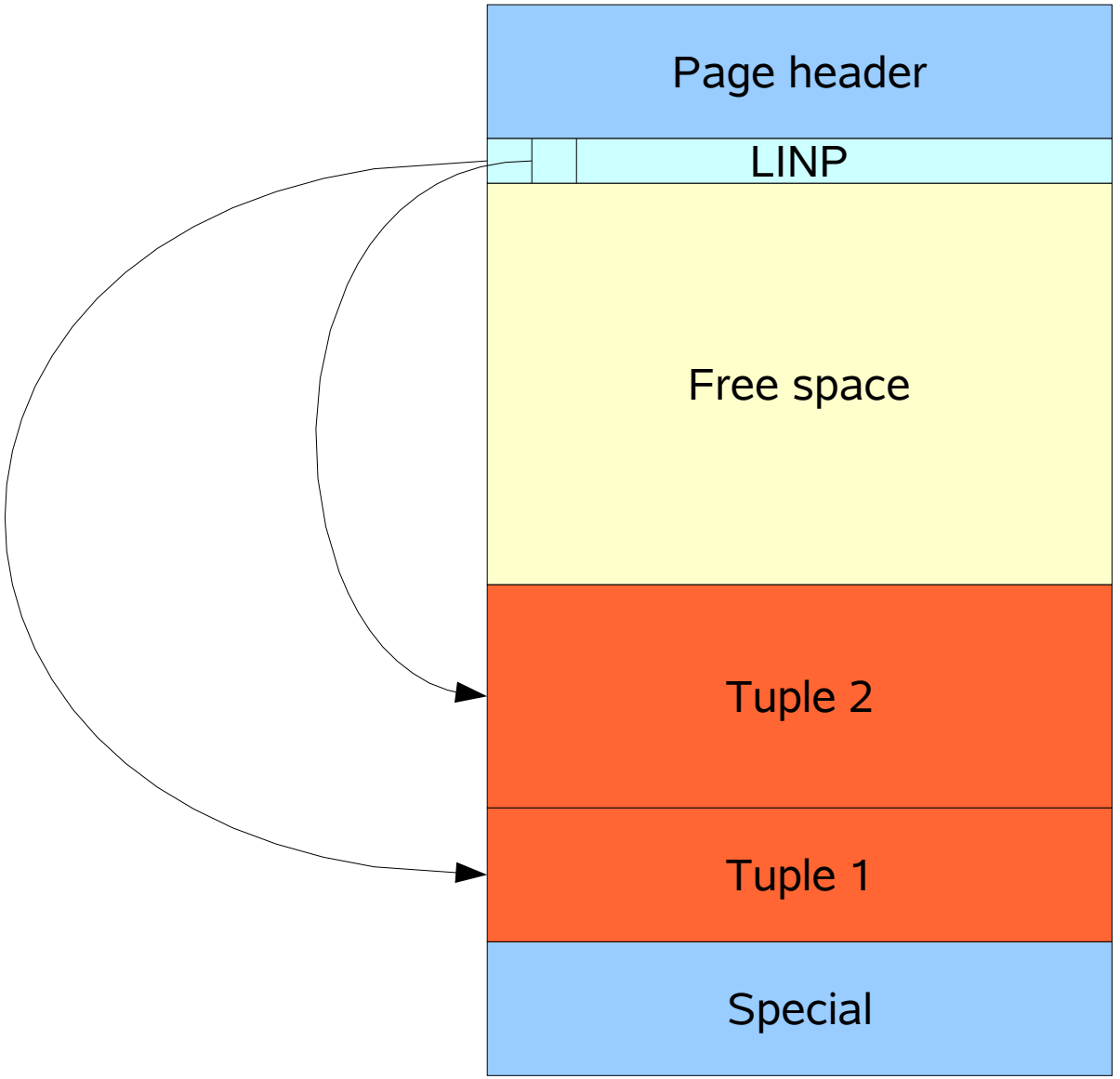
Tuning

- No general rule
- Tuning must be adjusted on
  - Application workload
  - Hardware configuration
- Bad application design invokes most problems
  - Application does not reflect RDBMS technology
  - Persistent layers are too abstract

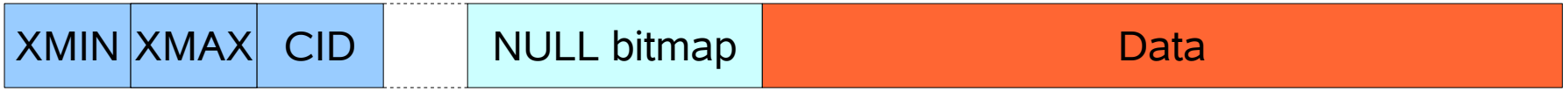
Multi Version Concurrency Control



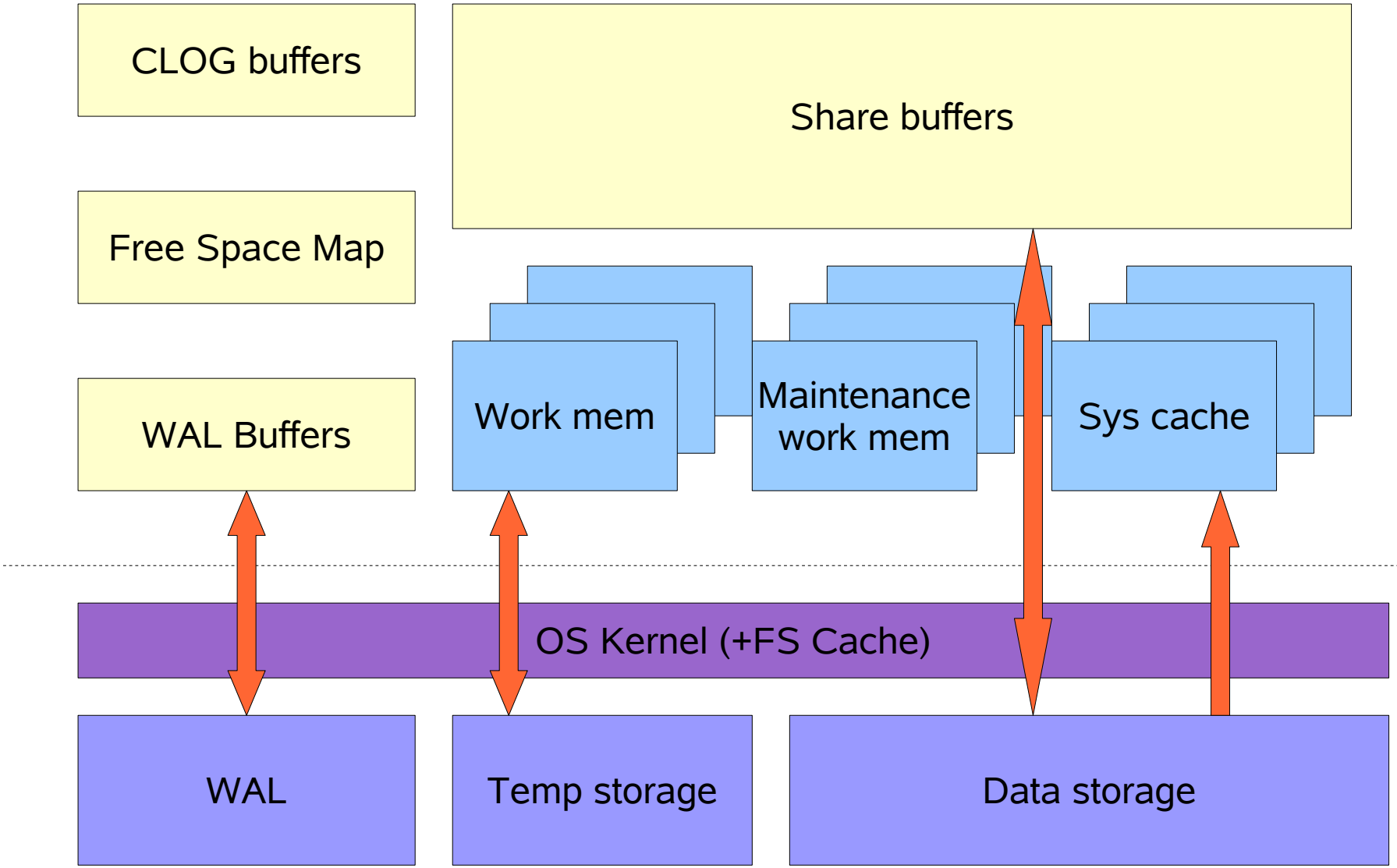
Page structure



Tuple structure

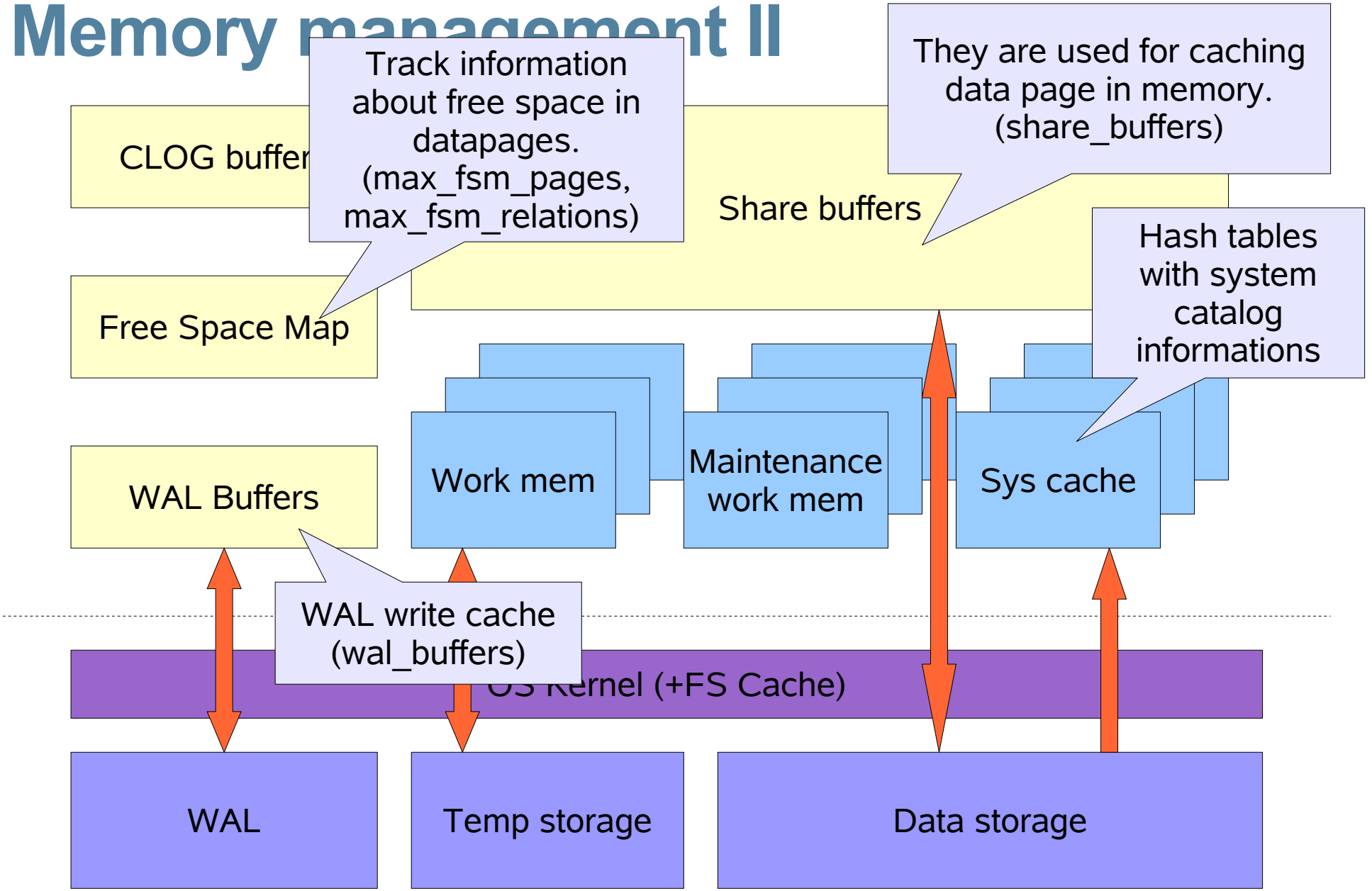


Memory management I

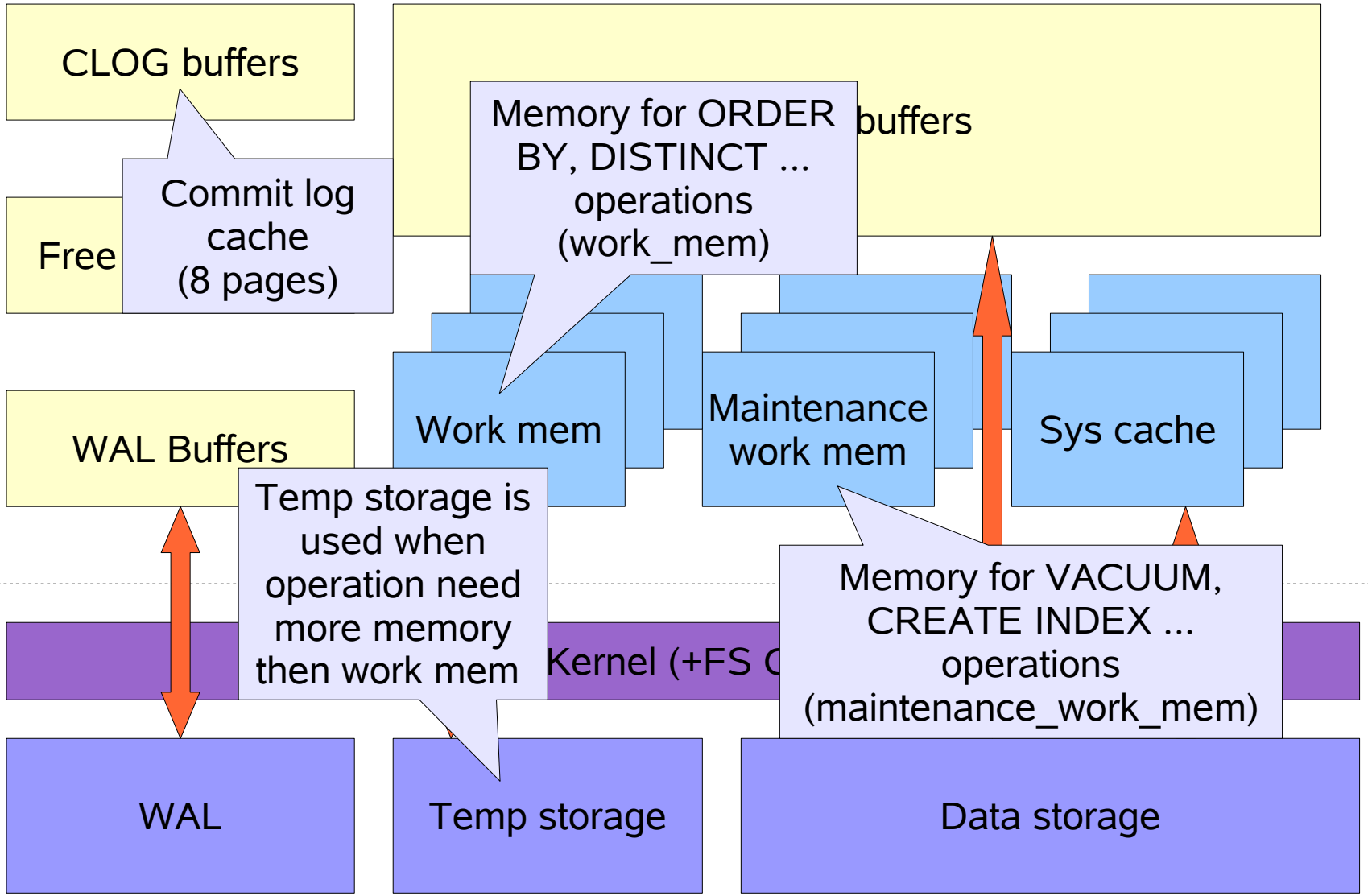


Notice: Clog, FSM file and others are not mentioned there.

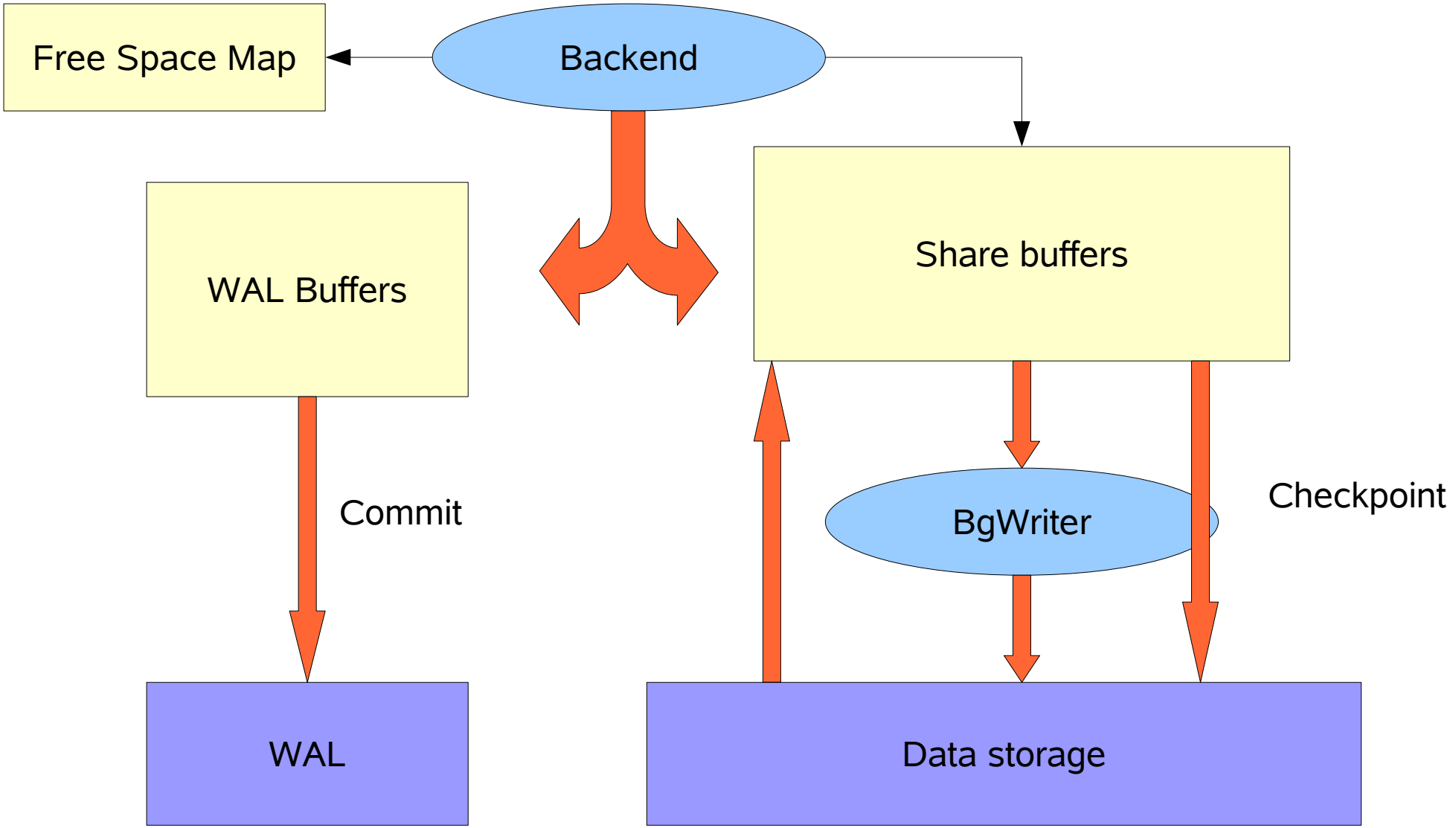
Memory management II



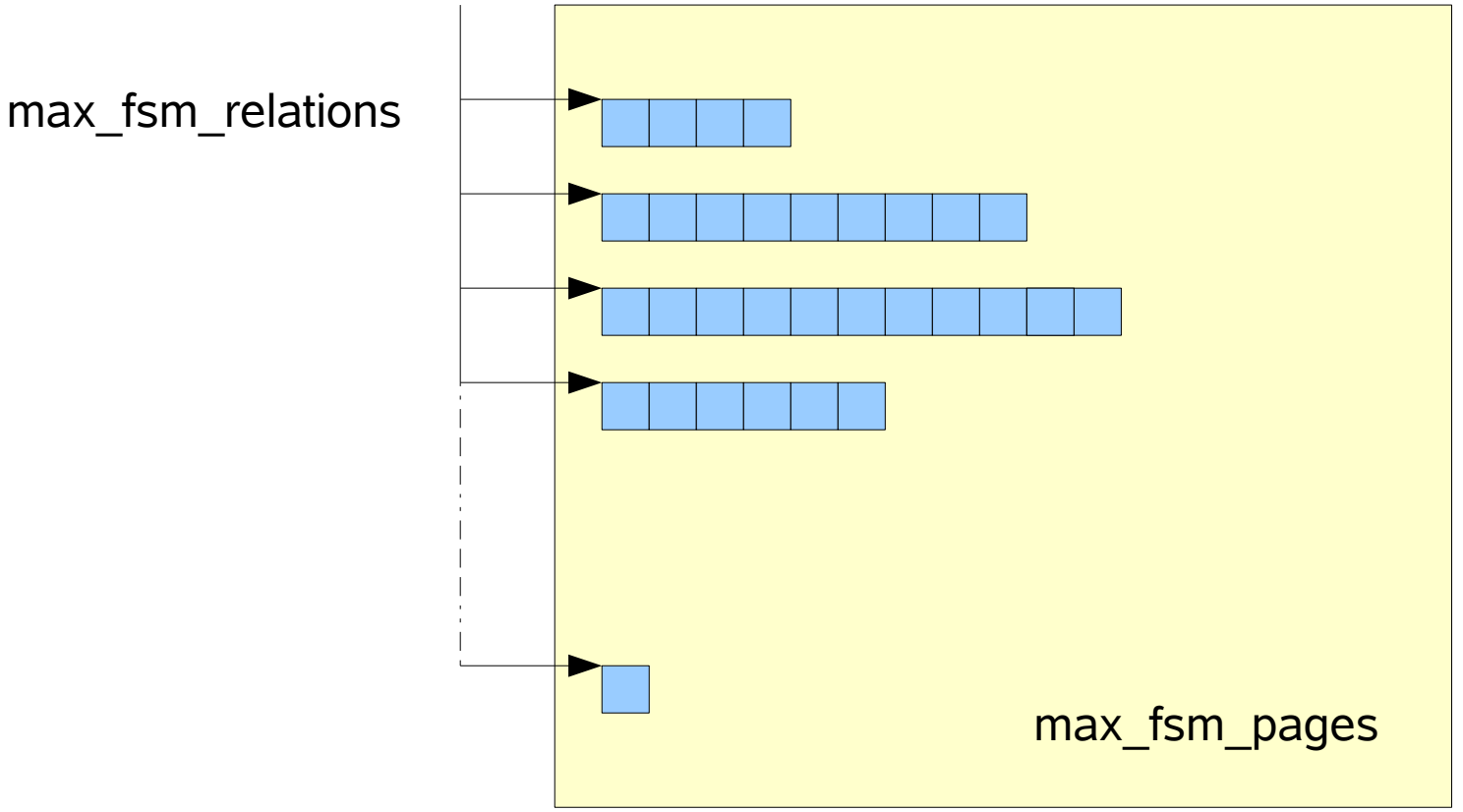
Memory management III



How INSERT works



Free Space Memory



Write Ahead Log (WAL)

- Store it on separate disk (forcedirectio UFS)
- Postgresql.conf
  - fsync – off during initial bulk load
  - wal_sync_method
    - open_datasync – UFS
    - fsync – ZFS
  - full_page_writes – could be off on ZFS
  - wal_buffers
  - wal_writer_delay – delay between WAL writer rounds
  - commit_delay – delay between commit and WAL buffer flush
  - commit_siblings

Checkpoints

- Synchronize Data storage and WAL
- Postgresql.conf
  - checkpoint_segments – number of checkpoint segments
  - checkpoint_timeout – maximal time between two checkpoints

Background Writer

- Reduce write waiting
- Speedup checkpoint operation
- Postgresql.conf
  - bgwriter_delay – delay between write rounds
  - bgwriter_lru_maxpages – maximal number of dirty page written in one round
  - bgwriter_lru_multiplier

Stats collector

- Several useful statistic
- Accessible as pg_stat_* views or pg_stat_*() function
- pg_stat_reset() reset all statistic
- Enabled by default in 8.3

D-Trace

- PostgreSQL 8.2 introduces 11 D-Trace probes
  - Transaction duration monitoring (transaction-start, transaction-commit, transaction-abort)
  - Locking monitoring (lwlock-acquire, lwlock-release ...)
- All global functions could be traced by PID provider

D-Trace example

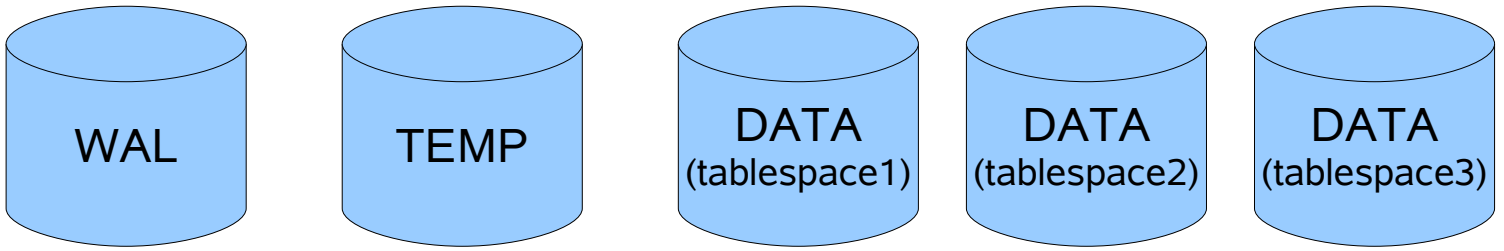
```
#!/usr/sbin/dtrace -qs
postgresql$1:::transaction-start
{
      @start["Start"] = count();
      self->ts  = timestamp;
}
postgresql$1:::transaction-abort
{
      @abort["Abort"] = count();
}
postgresql$1:::transaction-commit
/self->ts/
{
      @commit["Commit"] = count();
      @time["Total time (ns)"] = sum(timestamp - self->ts);
      self->ts=0;
}
```

Maintenance

- Vacuum
  - Cleans dead tuples
  - Updates FSM records
  - Tables, indexes defragmentation
  - Prevents transaction ID wraparound
- Analyze
  - Updates planner statistic

Hardware

- RAID 10 (mirror+striping)
- ECC memory
- Memory backup SCSI controller
- 8-12 CPUs are optimal (for version 8.2)
- Spread data to more spindles – store WAL, data, indexes separately



PostgreSQL tuning

Zdeněk Kotala
zdenek.kotala@sun.com