PostgreSQL
In Brazilian Public Institutions

Flavio Gurgel
PG Day'15 Russia

About me

- Brazilian
- Now living in France
- Electronics Engineer
- DBA since 1995
- PostgreSQL user since 1999
- PostgreSQL DBA after all

Where I work

2009 - 2013
São Paulo
Brazil

Today
Paris
France

Timeline
Free Software in Brazil

- 2003 - Open Source law in the state of Paraná
- 2005 - Free Software Center at São Paulo's University – USP
- 2008 - ODF standards in the state of Paraná
- 2008 - ODF standards for the country
- 2008 - Country planning and Rulings for Free Software adoption and support contracts

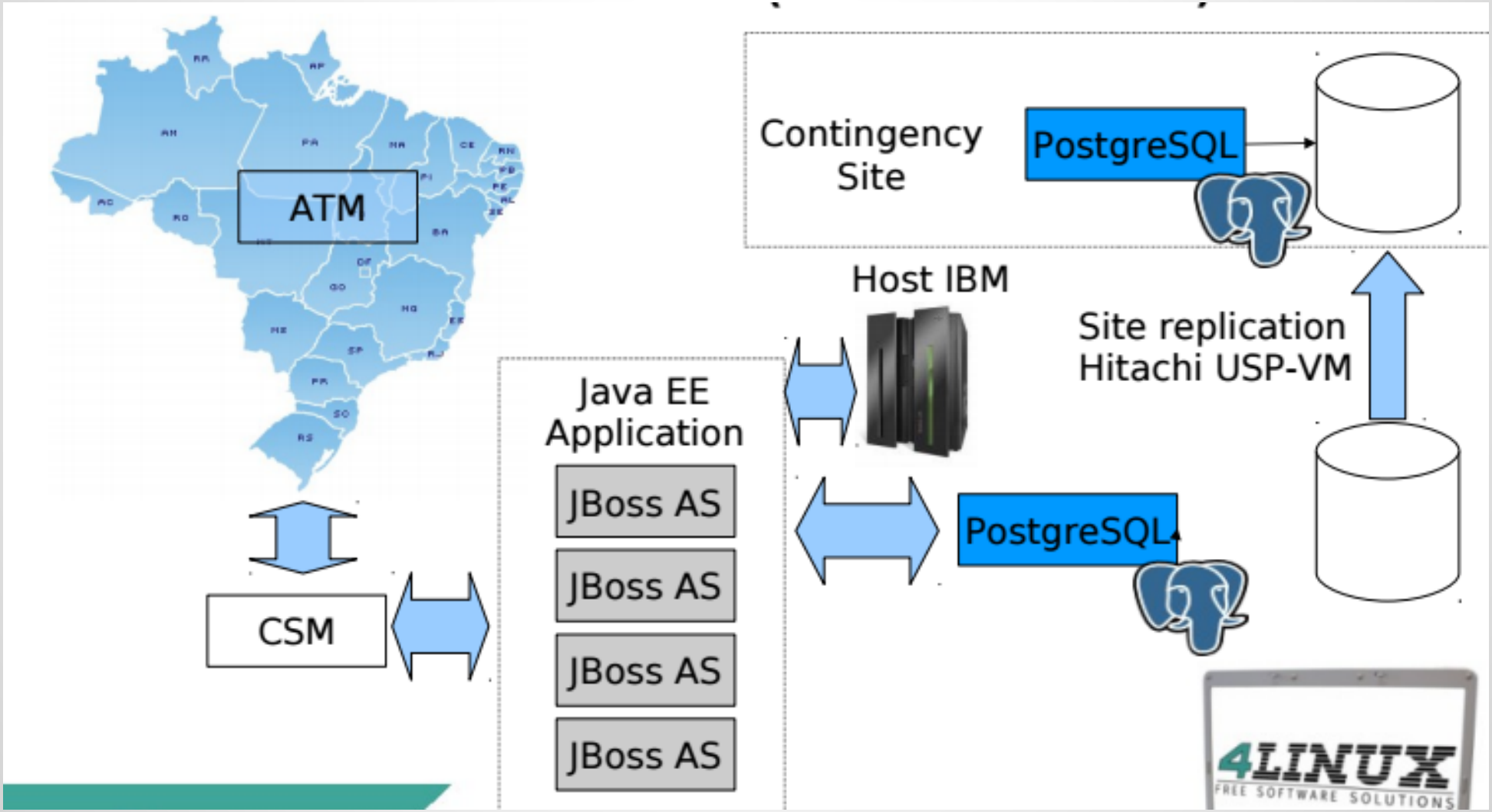
2009 - PostgreSQL arrives in a public bank

- 3 integrated systems
  - OLTP system to handle ATM requests
    - 750 TPS in peak hours
    - Average response about 1.5 seconds
    - ~250GB database
    - Highly available

2009 - PostgreSQL arrives in a public bank

- 3 integrated systems
  - OLAP system to monitor the ATMs
    - Ten's of requests/second
    - 1.5 TB database
  - OLTP messaging system
    - Small database
    - Mostly queue handling
    - Highly available

The OLTP System



First challenge
HW and OS were already chosen

- Non optimal storage system recycled from another project
- Non optimal final storage following "internal standards"
- Storage Tetris
- Solaris was (is) not open source
- ZFS was a new kid on the block at the time

Second challenge
PostgreSQL specifics

- People know the proprietary offer
- People trust proprietary offer
- People are afraid of changes
- DBAs are conservative
- PostgreSQL is mostly unknown
- High executives wanted something like RAC
  - shared disk multi-master approach
  - competition matters

Third challenge
Application

- Java application that implies Hibernate
- Ported from proprietary databases
- Some bad habits from developers
- The scale of a national bank is several times higher than any other

Fourth challenge
Address real challenges

- Amongst previous challenges
- PostgreSQL tuning
- Operating System tuning
- Hardware tuning
- Give answers to support tickets

A 6 month path to success
Availability

- Start with a multidiscipline support team
- Have a backup strategy in place
- Have a dump backup strategy
- Have a PITR backup strategy
- Store PITR backups in accessible disks
- Store dump backup in long term tape storage
- Replicate everything
- Did I say backup everything?
- Test your recoverability
- Test again in a regular fashion

A 6 month path to success
Stability

- Lots of PostgreSQL tuning
  - shared_buffers
  - work_mem
  - connections
  - checkpoints
  - bgwriter
  - sync method
  - buffers

A 6 month path to success Autovacuum

- NEVER TURN IT OFF
- Tune it well
- If it slows the system down, finetune again
- better be aggressive then never ending vacuums
- Transaction wraparound can freeze everything

A 6 month path to success Operating System

- SAN card queue
- Multipath
- When in Solaris
  - forcedirectio (UFS)
  - Limit ARC cache (ZFS)
  - have vendor support
- When in Linux
  - /etc/limits.conf
  - Use xfs if possible, otherwise ext4
  - Use a proven supported kernel version
  - You may need kernel performance experts

A 6 month path to success
Disks

- The base of good and consistent performance
- Have a tablespace strategy
- Separate pg_xlog
- Never ever use RAID 5
- Beware mixed auto tiering SAS/SSD arrays
- Know where the data is
- Discuss clear numbers with hardware vendor
- Attention to synchronous replication
- Watch fiber latency and bandwidth
- Watch closely array cache and processor usage

A 6 month path to success
Queries

- You may need to partition tables intelligently
- You may need to index your data in advanced ways
- You may need to destroy several unused indexes
- You WILL work with application developers
- Don’t let hibernate generate the schema
- Control the schema version
- You will have some night shifts

A 6 month path to success Maintenance

- Usually is not necessary to schedule reindex
- Sometimes is necessary to vacuum full a table
- Learn how to measure table and index bloat
- A purge policy has to be discuted
- Put the policy in place ASAP

Final Paradigms
Turning into reality

- Provide training sessions
- Know Postgres limits
- Shout out loud the strong advantages
- No DBMS is the same as the other
- Think PostgreSQL all the time
- Avoid comparisons and conflicts
- Selling points are hard to defeat

PostgreSQL
Strong points to spread

- PostgreSQL can keep very high TPS
- The MVCC model is rock solid
- COMMIT complicate transactions is fast
- Data was never lost in this bank project
- Extensive official documentation
- The open source model
- The GREAT community
- Support companies committed to the community
- Issues are solved quickly
- Cost is definitely lower

Needs risen at the project (remember - it was 8.3!)

- Some are addressed in recent versions
  - Scaling out
  - More transparent Caching
  - Maintenance routines low use of resources
  - SQL administration
  - System catalogs are a bit complicated
  - Graphical tools need improvement
  - Audit is hard to implement
  - No tool to show evident bottlenecks
  - Parallel query for analytics

Happy end

- PostgreSQL is running since then
- People in all corporate levels are happy
- PostgreSQL evolves fast
- Did I say that the community is great?

Coffee confessions:
psql is awesome! I wish I would never use that other proprietary tool again.

Thanks!

- fhagur@gmail.com
- @fhagur