

# Don't "Default"

## *5 Ways to Improve PostgreSQL Performance*

**Dave Cramer**  
**xTuple Canada**

**Friday, October 11, 2013**



# Don't Default

- ▶ 5 Easy ways to improve PostgreSQL performance
- ▶ Use Linux
- ▶ The best hardware you can afford (and then some)
- ▶ Tune PostgreSQL



# Use Linux

- ▶ Almost every application performs better on Linux
- ▶ Linux can be stripped down to the bare essentials not wasting CPU drawing a mouse cursor
- ▶ PostgreSQL designed for Linux, Windows was an afterthought
- ▶ Easier to tune



# Best hardware you can afford

- ▶ Your salary costs are considerably more than hardware costs
  - ▶ If employees have to wait for data workflow is interrupted
  - ▶ Pure time costs of waiting for data



# Best hardware you can afford

- ▶ Not all machines are equal. Most machines are not optimized for databases
  - ▶ Databases require disk speed, memory and then CPU; in that order
  - ▶ The computer marketing departments tend to market in the reverse order
  - ▶ Priority should be good RAID controller and disks, as much memory as you can afford, and then CPU.



# Best hardware you can afford

- ▶ How much is your data worth? Or how much opportunity will be lost if you lose your data?
- ▶ What would it cost worst case to be down for a day, or a week ?
  - ▶ RAID 1+0 disk array to protect against disk failure
  - ▶ Also better performance, more disks the better to spread the I/O over disks
  - ▶ Use more than one set of disks to offload writing of WAL logs



# Best hardware you can afford

- ▶ Battery backup or cache protection, this is a must! If your power fails you can lose ALL of your data without.



# Tune PostgreSQL

- ▶ What is important
  - ▶ Reading and writing to the disk
  - ▶ Databases save your data so they have to write to the disk
  - ▶ Searching for data means they have to read the disk, sometimes lots of it
  - ▶ Random access of memory is 120,000 times faster than random access of disk.





# Tuning knobs

- ▶ Big knobs to adjust
  - ▶ Memory
  - ▶ WAL size
  - ▶ Number of connections
  - ▶ Disk speed ratio



# Memory

## ▶ Shared Buffers

- ▶ PostgreSQL uses this for cache, rough setting 25% of available memory up to 8GB on linux. 1GB on Windows

## ▶ Work Memory

- ▶ Used for sorting, group by, etc
- ▶ If you don't have enough tape sort will be used
- ▶ Can be adjusted per user, connection, database
- ▶ Per connection



# Memory

- ▶ Effective cache
  - ▶ Tell postgresql how much memory is available
  - ▶ Roughly 75 % of available memory
  - ▶ Does not use memory, just a hint
- ▶ Maintenance work mem
  - ▶ Used by vacuum, create index

# WAL and checkpoints

- ▶ Two phase commit, causes writes to disk
  - ▶ Better to have on it's own disk to improve throughput

# Contact

**Dave Cramer**

**Sr. Software Architect**

**[DCramer@xTuple.com](mailto:DCramer@xTuple.com)**

