Big Compute, Big Net & Big Data: How to be big

Luiz Monnerat

PETROBRAS

26/05/2014

Agenda

- Big Compute (HPC)
  - Commodity HW, free software, parallel processing, scalability, etc.
- Big Net (Internet)
- Big Data

What can we learn?

How to be big ????

- How to be even bigger?

> Big Compute

- High Performance Computing (HPC) or Supercomputing
- ... *powerful machines to solve very complex numerical problems* …….
- Complex numerical problems examples…
  - Seismic processing, climate, research, etc.
- Powerful machines examples
  - Tianhe-2 (China): The biggest supercomputer in the world
  - grifo04 (Petrobras): The biggest supercomputer in Latin America

> grifo04 (2010)

- 17 racks
- 544 nodes
- 1088 GPUs
- 40TB RAM

Biggest supercomputer in Latin America

> bwr1: 1300 CPUs (2004)

PETROBRAS

> How to be Big Compute.....

- Commodity Hardware & Free Software
  - Inside the Data Center!!!!

- Parallel Processing
  - Many computing units…….
- Heterogeneous/Accelerated Processing
  - Many more computing units…….
- Scalability
  - Ability to provide more performance as you add more resources

> bw1 (1999)

bw1: the second Petrobras Linux commodity cluster

example of early use of commodity hardware & software in DC

PETROBRAS

> How to be Big Compute.....

- Commodity Hardware & Free Software
  - Inside the Data Center!!!!
- Parallel Processing
  - Many computing units…….
- Heterogeneous/Accelerated Processing
  - Many more computing units…….
- Scalability
  - Ability to provide more performance as you add more resources

> Scalability

- For decades the HPC community has been working in several scalability issues…
  - Load balancing
  - Communication
  - Synchronization
  - etc.
- … but it has been quite tied to the Client/Server (& master/slave) models
  - Client/server architectures have intrinsic scalability issues

> Agenda

- Big Compute (HPC)
  - Commodity HW+free SW in the DC, parallel proc., scalability, etc.
- Big Net (Internet)
  - P2P
- Big Data
- How to be even bigger?

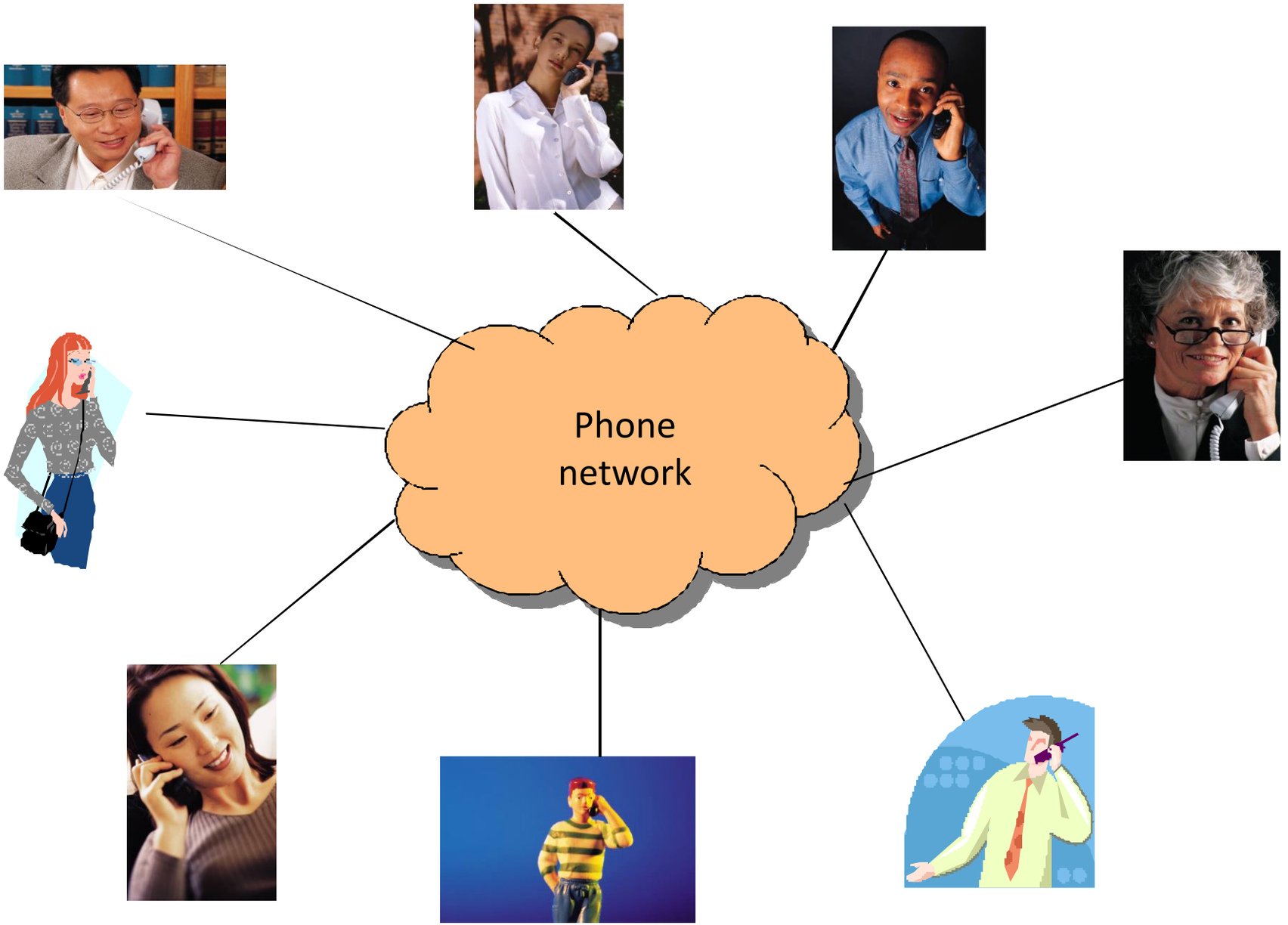
> Big Net

- Internet
  - World wide network of computers .... and things!
- Systems with millions of users and devices…
  - Client/Server model: Tens of thousands servers
- Systems with unpredictable demands…
  - Client/Server model: How to size the server farm?
- Systems with fluctuations in demand…
  - Client/Server model: Server farm must address peak demand.
- Client/server model:
  - Very very high cost
  - Difficult to size
  - Difficult to address fluctuations in demands (e.g., flash crowds)
  - Scalability is critical

> How to be Big Net.....

- P2P !!!!
  - peer-to-peer
  - Skype, Napster, KaZaA, eMule, BitTorrent, etc.
- Distributed systems architecture in which all participants perform the same functions (peers) and freely communicate among themselves
  - Each peer acts both as client and server
  - No dedicated server. All clients act also as servers.
- Highly and intrinsically scalable!!!!!!
  - If you add one more client, you are also adding one more server
  - If you add one million clients, you are also adding one millions servers
  - Allow the system to stay balanced as it grows
  - But distributing the server load may not be easy….
- Able to automatically adjust to fluctuations in demand.
- Much more scalable than client/server solutions

> P2P Scalability



> Agenda

- Big Compute (HPC)
  - Commodity HW+free software in the DC, parallel proc., scalability, etc.
- Big Net (Internet)
  - P2P
- Big Data
  - P2P in the DC
- How to be even bigger?

> Big Data

- *En vogue* .... several definitions....
  - Big volumes of unstructured data
  - Data Mining
  - Analytics
  - 3 V's : **Volume, Velocity**, Variety
  - 5 V's : **Volume, Velocity**, Variety, Value, Veracity
  - 6 V's, 7 V's ....
- The Volume and Velocity demands of Big Data environments challenge the traditional and client/server filesystems (including "scalable" parallel filesystems such as Lustre).

> How to be Big Data.....

- P2P inside de Data Center!
- P2P filesystem
- Google FS
- Hadoop FS (HDFS)
  - Based on Google FS
  - Hybrid P2P
- There are already other interesting P2P applications
  - e.g, SystemImager

> Pure and Hybrid P2P systems

- Several P2P systems use Client/Server solutions to store and manage metadata
  - Metadata is harder to manage in a P2P way
  - Metadata volume & performance demands are usually much smaller
  - Hybrid P2P systems
  - e.g., Napster, Hadoop FS, etc.

- Pure P2P systems
  - e.g., KAD/eMule, BitTorrent
  - More scalable
  - Typically use *Distributed Hash Tables* (DHTs) to store metadata

> Agenda

- Big Compute (HPC)
  - Commodity HW+free software in the DC, parallel proc., scalability, etc.
- Big Net (Internet)
  - P2P
- Big Data
  - P2P inside the DC
- How to be even bigger?
  - DHT : *Distributed Hash Tables*

> DHTs

- Distributed (& P2P) data structures
  - Like “normal” hash tables, DHTs are typically used to store metadata
  - The (meta)data is distributed among the peers according to a hash function (e.g., SHA1)

- DHTs are typically used as P2P directories
  - Metadata stored in a DHT is located through **lookup** requests
  - But locating metadata (solving lookups) in a highly distributed DHT is not an easy task..... and it is a key issue in DHTs.

> Solving Lookups in DHTs

- First DHTs solved lookups recursively
  - Introduced in the beginning of this century (e.g. Kademlia/Kad)
  - Each lookup generates several smaller lookups (or hops)
  - Multi-hop DHTs
- In a Multi-hop DHT, each lookup may require dozens of hops
  - Very high latency, but ok for many Internet applications
  - Not OK for Big Compute, Big Data and other Data Center apps
- More recently, Single-hop DHTs were introduced
  - Low latency P2P directories

> Single Hop DHTs

- OneHop (MIT, Gupta et al, 2004)
  - High levels of load imbalance
  - High bandwidth overheads
- 1h-Calot (IBM+Rochester, Tang et al, 2005)
  - Good load balance
  - Higher bandwidth overheads
- D1HT (Petrobras+UFRJ, Monnerat e Amorim, 2005)
  - *Distributed One Hop Hash Table*
  - Good load balance
  - Low bandwidth overheads

> D1HT

- Low latency:
  - As required by HPC & Big Data applications
- Low bandwidth overhead:
  - As required by Internet applications

D1HT: A general purpose DHT

Source code available for download (GPLv2)

Best HPC Brazilian PhD Thesis (2010)

PETROBRAS

> Conclusions

- Knowledge and experience acquired in HPC, Internet and Big Data environments are important in designing and building big systems
- Hybrid P2P systems are more scalable than Client/Server solutions, and allow low cost implementation of large scale systems
- Even bigger systems may require pure P2P implementations with the use of DHTs
- Most DHTs have high latency and/or high load imbalance and/or high bandwidth overhead
- D1HT is able to combine high scalability with low bandwidth overheads, good load balance and low latency, and so it can be used both in Internet and Data Center applications

Thank you!

http://br.linkedin.com/in/luizmonnerat

You may get D1HT!

http://www.cos.ufrj.br/~monnerat