Laboratório de Sistemas Distribuídos

A Large Scale Fault-Tolerant Grid Information Service

Francisco Brasileiro, Lauro Beltrão
Alisson Andrade, Walfredo Cirne
{fubica,lauro,alisson,walfredo}@dsc.ufcg.edu.br

Universidade Federal de Campina Grande, Brazil

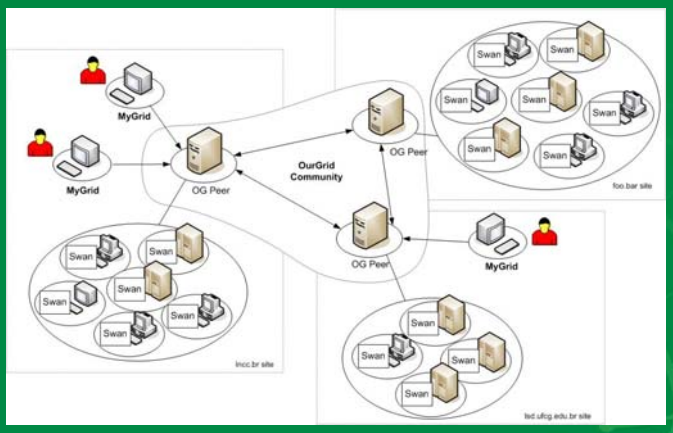
Sujoy Basu, Sujata Banerjee
{Sujoy.Basu,Sujata.Benarjee}@hp.com

Hewlett-Packard Laboratories, USA

MGC'2006, Melbourne, November 27th 2006

Motivation

– OurGrid is a P2P grid middleware developed mainly at LSD/UFCG that envisions to gather hundreds of thousands of resources in tens of thousands of peers (see http://www.ourgrid.org/)



Motivation

- How to discover "entities" with particular attributes in a large P2P grid system?
  - A scheduler might want to locate suitable resources
    - OS=linux && RAM ≥ 1G && clock > 4GHz && load < 0.5
  - A user may want to locate a dataset that contains particular data items
    - rainfall && -37º52' < long < -37º46' && 144º54' < lat < 145º03'
  - …

Motivation

- To support this service a large scale Grid Information Service must
  - Scale to large amounts of meta-data and to high operation load
  - Be able to execute rich queries that encompass not only multiple attributes, but also range operators
  - Tolerate faults

Centralized/static hierarchical GISs

- Support rich queries
- Fault tolerance may be implemented via the use of redundant servers
- Scale poorly (at a reasonable cost)

P2P GISs

- DHT-based
  - Scale well
  - Highly reliable
  - Do not support rich queries in a natural way
- Distributed tree-based
  - Scale well
  - Support rich queries
  - Do not tolerate faults

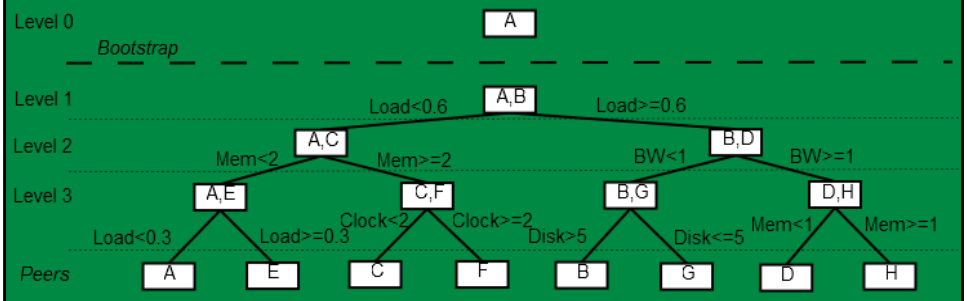
Our Approach

- Extend an existent kd-tree based GIS with fault tolerance mechanisms
- Do that without impacting scalability and the ability to support rich queries

Outline of the rest of this presentation

- Explain the functioning of NodeWiz
- Analyze the impact of faults in NodeWiz
- Explain the fault tolerance mechanism proposed
- Show the results of an experiment run with the fault-tolerant version of NodeWiz
- Present conclusions and directions for future research

NodeWiz's Rationale

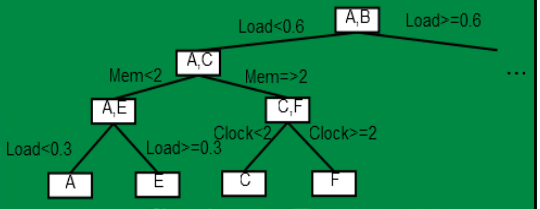


- Providers advert their service attributes
- Each NodeWiz node is responsible for a portion of the attribute subspace
- Clients query for services with particular attributes
- Adverts have a TTL and should be renewed
- Peers maintain routing tables to route operations

Performing an operation

- Query sent to A: load < 0.4 && Mem >= 2GB && Clock >= 2 Ghz

Peer A			
Level	Attr	Range	Route
0	Load	0.6 - INF	B
1	Mem	2 – INF	C
2	Load	0.3 – 0.6	E



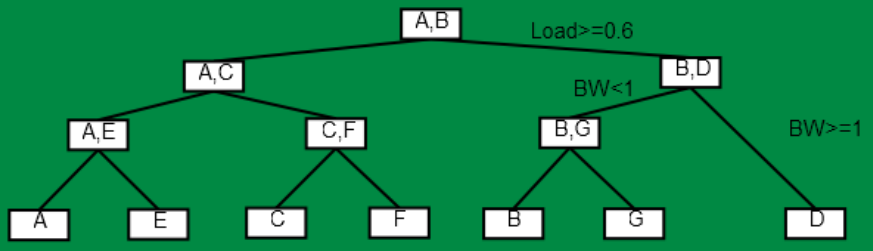
Peer C			
Level	Attr	Range	Route
0	Load	0.6 - INF	B
1	Mem	0 - 2	A
2	Clock	2 - INF	F

Peer F			
Level	Attr	Range	Route
0	Load	0.6 - INF	B
1	Mem	0 - 2	A
2	Clock	0 - 2	C

Joining a NodeWiz Network

- Identify the most overloaded peer
- Balance load with it
- Identify attribute subspace that can improve load balance

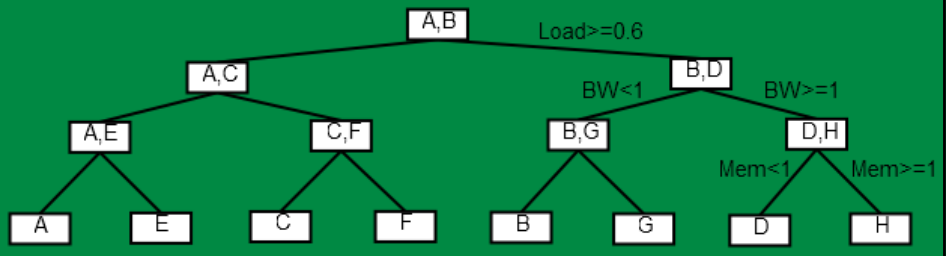
Joining a NodeWiz Network



- Say, node H wants to join the system
  - It asks one node it knows, which is the most overloaded node
  - It then joins the system at this node

Peer D			
Level	Attr	Range	Route
0	Load	0 - 0.6	A
1	BW	0 – 1	B

Joining a NodeWiz Network



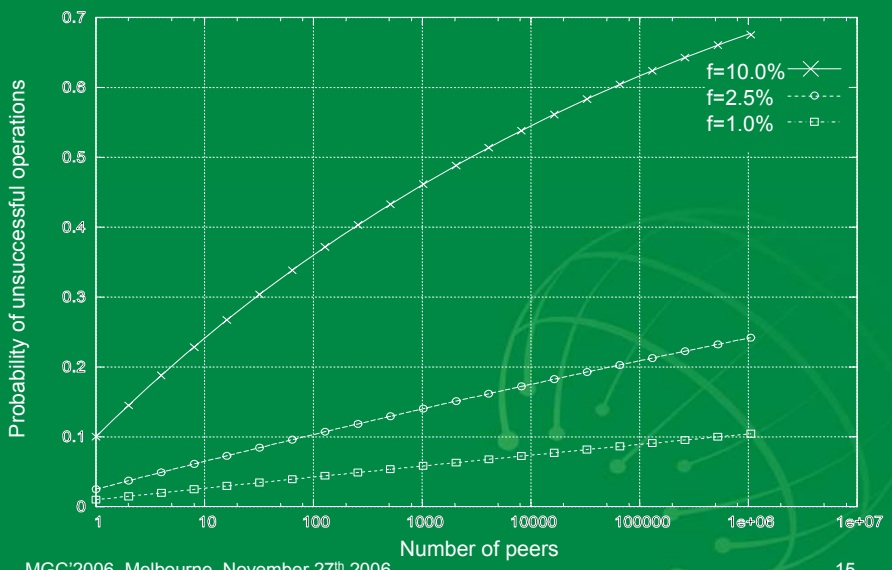
Peer D			
Level	Attr	Range	Route
0	Load	0 - 0.6	A
1	BW	0 – 1	B
2	Mem	1 – INF	H

Peer H			
Level	Attr	Range	Route
0	Load	0 - 0.6	A
1	BW	0 – 1	B
2	Mem	0 – 1	D

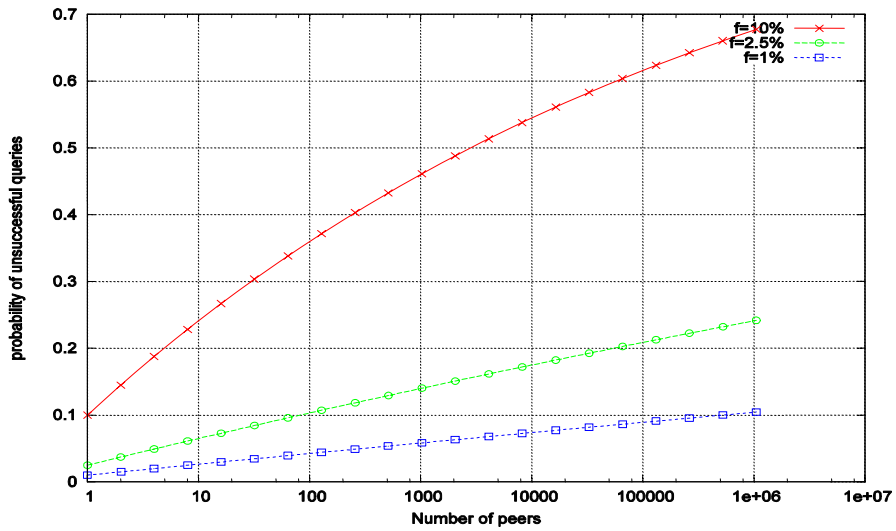
On the consequences of faults

- For the sake of analysis we assume
  - System is *well-formed*
    - all routing tables have the same size ($L=\log_2 N$)
  - Load is *well-balanced*
    - all peers receive the same load
    - All queries match a single peer
  - Probability of unsuccessful operation issued to P:
    - $POU(P) = \sum_{i=0}^{L} \frac{R_i(P)}{N} * [1 - (1 - f)^{i+1}]$

The impact of faults



The impact of faults



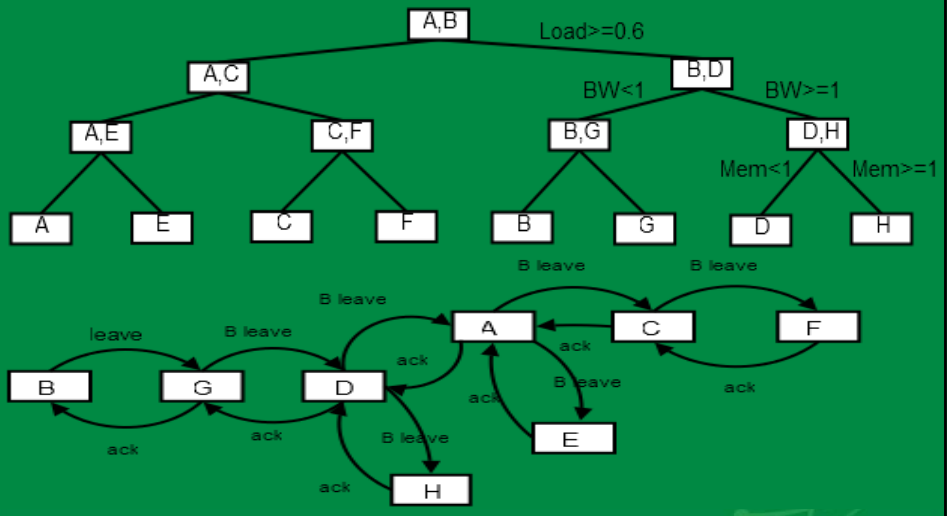
Voluntary leaves

- When a peer L leaves, its attribute subspace must be reclaimed by another peer
- L's replacement - R(L) - is the last peer in L's routing table
- L contacts R(L) informing about its departure
- R(L) stores L's state and propagates L's request for other peers that have L in their routing table

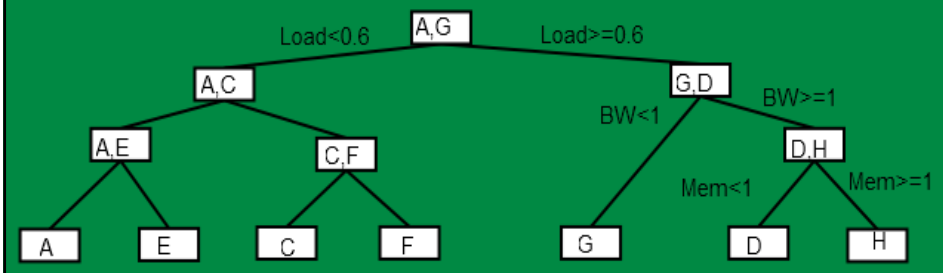
Voluntary leaves

- During leaving process, routing tables may become inconsistent
- L is kept in the system until it receives authorization to leave from R(L)
- L also forwards new adverts or queries to R(L)
- R(L) waits for acknowledges from other peers before authorizing L's departure

Voluntary leaves: B's departure



Voluntary leaves: B's departure

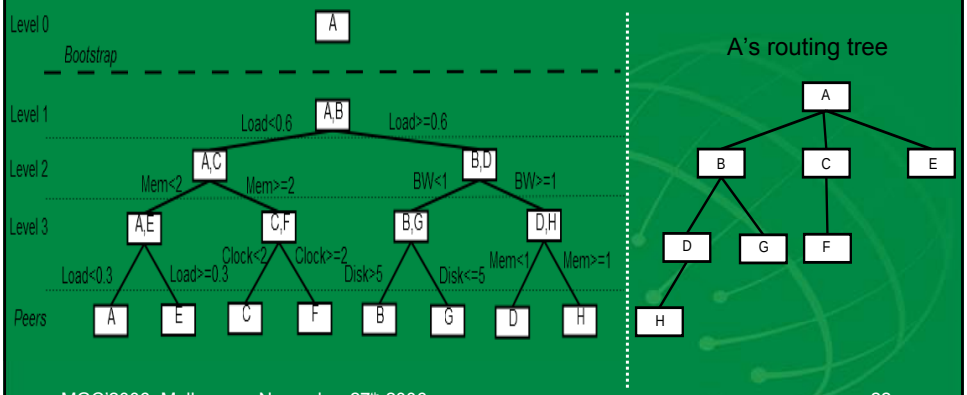


Dealing with involuntary leaves

- We are not concerned with the state of faulty peers
- Our approach is: R(L) must detect that L has failed, so R(L) monitors L
- When L fails, R(L) creates a virtual peer S(L) that assumes L's identity
- S(L) initiates a voluntary leave with R(L)

Dealing with simultaneous faults

- What if both L and R(L) fail at "the same time"?



Dealing with simultaneous faults

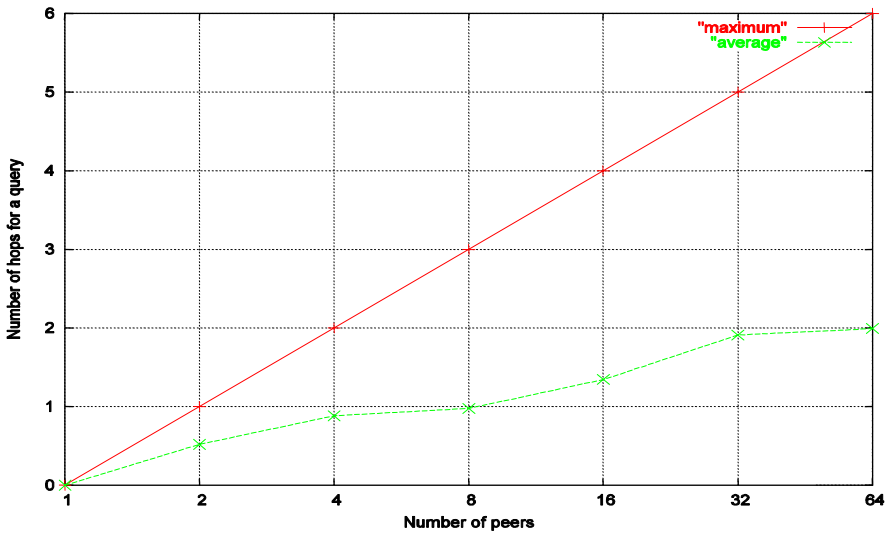
- A peer *P* monitors a set of peers where each set has a leader that may be replaced by *P*
- *P* monitors L sets, where L is the size of *P*'s routing table
  - Each peer in its routing table is a set leader
- Each set is composed by every peer that can replace the set leader before *P* replaces it
  - Example:

Peer A				
Level	Attr	Range	Route/Set leader	Set
0	Load	0.6 - INF	B	{D,G,H}
1	Mem	2 - INF	C	{F}
2	Load	0.3 – 0.6	E	{}

Experiments

- Performed in PlanetLab in a system with up to 64 peers
- Adverts came from PlanetLab data collected by Ganglia
- Experiments have a start up phase and a performance collection phase
  - In the start up phase peers join the system while adverts are stored (no TTL)
  - In the collection phase queries are sent randomly to peers in the system
  - Half of data is used in each phase

Experiments



Conclusions and future work

- Faults have a great impact on the QoS of tree-based GIS
- The fault tolerance mechanisms proposed solve the problem without impacting scalability
- OurGrid 4.0 will incorporate the fault-tolerant version of NodeWiz to perform resource discovery
- We intend to perform experiments to compare FTNodeWiz with DHT-based GISs in fault-prone scenarios

Questions?

- Contact information:
  - Francisco Brasileiro
    - fubica@dsc.ufcg.edu.br
  - LSD/UFCG
    - http://lsd.ufcg.edu.br/
  - OurGrid project
    - http://www.ourgrid.org/
  - For a real-time snapshot of the running system access http://status.ourgrid.org/