Parallel Grid Computing Challenges

eero.vainikko@ut.ee
University of Tartu, Institute of Computer Science
Distributed Systems research group

EXCS Kickoff
19 September 2008

In theory, there is NO difference between theory and practice.
But, in practice, there IS.

Jan L. A. van de Snepscheut

- Main driving forces of computational science
- Parallel computing - state of the art
- What is Grid computing?
- Grid computing main challanges
- How we are going to meet the challenges
- Friend-to-friend (F2F) computing framework
- F2F future
- DOUG and DOUG@F2F

1 Driving forces of computational science

1.1 Some predictions

Much-cited legend: In 1947 computer engineer Howard Aiken:

USA will need in the future at most 6 computers!

Why do we need parallel computing at all?

Gordon E. Moore's (founder of Intel) law:

(1965: the number of switches doubles every second year)

1975: - refinement of the above: **The number of switches on a CPU doubles every 18 months**

Until 2020 or 2030 we would reach in such a way to the atomic level or quantum computer

1.2 Grand challenges

***Grand Challenge problems* listed in Wikipedia today:**

- Applied fluid dynamics
- Biomedical imaging and biomechanics
- Fundamental computational sciences
- Meso- to macro-scale environmental modeling
- Molecular biology
- Nuclear power and weapons simulations
- Molecular design and process optimization
- Ecosystem simulations
- Cognition
- Strong artificial intelligence

2 Parallel and Grid computing - state of the art

2.1 What is Grid computing

Grid is a term for a system, where

- geographically remote
  – computers,
  – supercomuters
  – special devices (data storage, censors etc.)

are forming a common resource such a way that user of the system must not know

- where exactly his/her computations and data analysis tasks are being processed
- how exactly the data management is performed

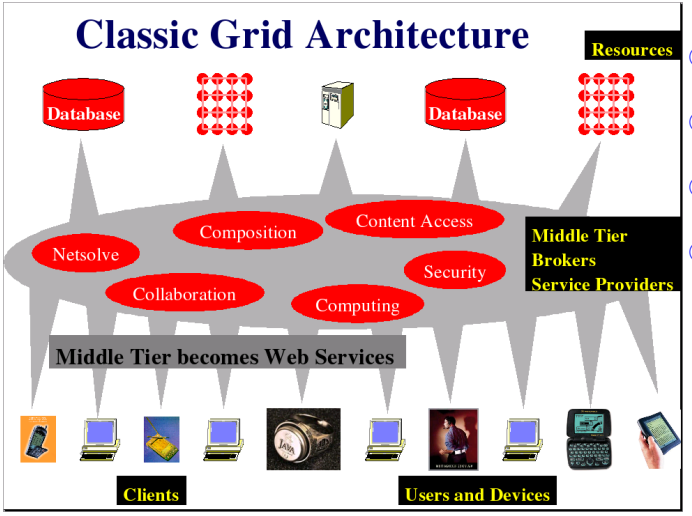
The main purpose of the Grid is to aggregate

- different computers,
- computer centers
- data storages
- special devices

in a way that their use is simple and comfortable also without any deeper knowledge in

- IT
- distributed- and supercomputing

Grid – the next big step after occurrence of the Internet



What was promised:

- As easy as power-grid
- Everywhere
- Accessible to everybody
- Access to any desired resource
  - CPU cycles
  - Storage
  - Special devices

2.2 State of the art

(What we have got so far)

- Batch processing
- - with global scale, though
  - Parallel jobs still available, within a single cluster
  - You really need to know the resource you are running on
- Grid = merely web-services in another wrapping!
- Storage – really unconfortable for user

Problems with existing models

(Not every problem in every middleware)

- Single point of failure
- Lack of scheduling
- Poor scalability
- No means for privacy
- No way for cycle stealing (scavenging)
- Firewall problem
- Requires very large installation
- No economy

Desktop Grids

Possible solution to the problems?

- BOINC www.boinc.org
  - Seti@home, Einstein@home, Proteins@home, Africa@home, SZTAKI@home, Folding@home
- Condor
- ALCHEMI
- JNGI
- **Minimum Intrusion Grid**

But still, for major parallel computational problems we need a Grid of computing nodes that can communicate with each other

- Automatic best available network/protocol choice
- Ability to traverse firewalls
- Setup as easy as *Skype*
- Interplay with social networking trust

Friend-to-friend Computing

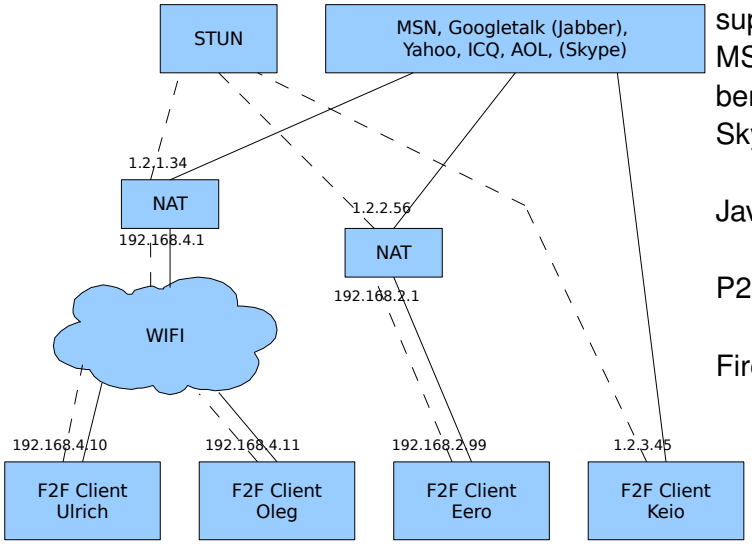
3 Parallel Grid Computing

3.1 Friend-to-friend Computing

Friend-to-friend Computing (F2F); Called also: Spontaneous Desktop Grid

- Developed at the Institute of Computer Science, University of Tartu (U.Norbisrath and E.Vainikko) f2f.ulno.net
- Computing environment based on spontaneous groups
- Instant Messaging friend groups used for triggering the Computing Grid Environment
- Merging ideas from
  - Peer-to-peer
  - High Performance Computing
  - Social networks in instant messaging
- Extremely easy concept for
  - authentication
  - authorization
    based on inherent trust between friends

3.2 F2F Architecture



IM protocols/solutions supported:
MSN, GoogleTalk (Jabber), Yahoo, ICQ, AOL, Skype

Java implementation

P2P-MPI

Firewall hole punching

Ulrich Norbisrath, Keio Kraaner, Eero Vainikko, and Oleg Batraĺev, Friend-to-Friend Computing Instant Messaging Based Spontaneous Desktop Grid. Accepted for The Third International Conference on Internet and Web Applications and Services (ICIW 2008), pp. 245-256, June 2008, Athens/Greece, IEEE Computer Society Press

3.3 F2F - next steps

- F2F kernel C-implementation - work in progress
  - Support for
    - other languages (Python, C, Fortran,...)
    - other devices (Mobile, PDA-s, Supercomputers)
- Interfacing with other Grid infrastructures
- Virtualisation
- ...F2F2F...
- Security
  - Authorisation
  - Encryption
  - Fault tolerance
  - Proof of non-vulnerability

- **Porting real applications to F2F**

4 DOUG

4.1 Domain Decomposition on Unstructured Grids

DOUG (*University of Bath, University of Tartu*) 1997 - 2008
I.G.Graham, M.Haggers, R. Scheichl, L.Stals, E.Vainikko, M.Tehver, K. Skaburskas, O. Batrašev

Parallel implementation based on:

- **MPI, UMFPACK , METIS, BLAS**
- Automatic parallelisation and load-balancing
- Systems of PDEs
- Highly heterogeneous 2D & 3D problems
- 2-level Additiivne Schwarz method
- 2-level partitioning using aggregation
- Automatic Coarse Grid generation
- Adaptive refinement of the coarse grid

4.2 DOUG Strategies

- Iterative solver based on Krylov subspace methods
  - **PCG**
  - **MINRES**
  - **BICGSTAB**
  - 2-layered **FPGMRES** with left or right preconditioning.
- Non-blocking communication where at all possible
  $\Rightarrow$ communication/computation overlap
- **Best algorithms possible for highly variable coefficients** - an ongoing research on parallel algorithm design

Porting DOUG to F2F

4.3 DOUG @ F2F

- Currently, DOUG written in Fortran95
- => research in the area of **re-engineering parallel applications / parallel application modernisation**
  - parsing (fortran) codes
  - parallel computation modelling
  - communication and data dependency derivation
  - automatic code generation

Some challenges (DOUG algorithms and F2F framework itself):

- Load balancing
- Fault tolerance
- Parallel efficiency

In conclusion: Wake up – the computing power is under your own fingertips

(...image commented out as I do not know whether there might be any copyright issue...)

Contents