parXXL: Fine Grained Simulation Experiments on GdX

GdX meeting – June 28, 2006 – Paris

Jens Gustedt - Stéphane Vialle - Amelia De Vivo

In memory of Amelia De Vivo

**parXXL: Project overview**

Client machine: comfortable fine grained development environment

Distributed server: fine grained simulations on coarse grained architecture

Minimize \((T_{dev} + \sum T_{exec})\)

Run very large simul.

Speedup fine grained simulations.

Software suite parXXL:
- par::cellnet
- par::cell
- par::cntrl
- par::mem
- par::bench
- par::sys
- par::cpp

2 previous environments
- ParCeL-6
- SSCRAP

**parXXL: Programming model**

Fine grained parallelism

Coarse grained parallelism

- BSP relaxed super-steps
- Dynamic cell net evolution
- …

Cell net creation

Cell net computation

Cell comm.

Cell net evolution
**parXXL: Execution support**

SSCRAP

- PRO model
- Handle huge data
- High portability
- High performances

- par::cntrl
- par::mem
- par::sys
- par::sys
- POSIX standard
- MPI Threads
- 2 optimized runtimes

**parXXL: Cellular level (ParCeL-6)**

A cell:

- Cell Parameters
- Cell Variables
- Cell input channels
- Cell output channels
- Cell external IO mechanisms

//Init function
... 
... 

//Iter function
... 
... 

//Term function
... 

User code

Cyclic running

Dynamic cell net evolution

3 kinds of output channels
parXXL: Optimized cell nets

Basic cell net creations:

Main function

Ex: Cubic cell net

1 PE

1 PE

1 PE

Basic user code:
- Automatic load balancing
- Neighbor cells are not grouped

Optimized cell nets (par::cellnet)

Optimized user/par::cellnet code:
- Explicit load balancing
- Neighbor cells are grouped

parXXL: Process deployment

parXXL exec. support:
par::cntrl
par::mem
par::bench
par::sys
par::cpp

User code including par::cell operations

User code & optimized par::cellnet code

Cell net creation
Loop:
- Cell net computation
- Cell communications
- Cell net update
- Cell net delete

Client

Worker

Worker

Worker

Server

Internet
Relaxation perfs. on GdX machine

- **Speed up** fine grained simulations using more processors
- **Extensibility**: process larger problem on more processors with similar cost for each cell iteration

Scales until 420 $10^6$ Cells on 310 PEs

parXXL: Experience with GdX

**OAR experiences:**
- Authentication needs ssh & certificates:
  - not just OAR mechanisms,
  - but user ssh socket can not connect to the user machine
    not so comfortable!
- Nodes are not reliable on startup!
  forces use of interactive mode!

Great loss of time for benchmarks!
parXXL: Experience with GdX

Large scale experiences:

• Existing architecture is good (network is fast enough).
• parXXL computing and communication steps are scalable.
• MPI launching is sequential (1s per job!).
• Current parXXL cell net deployment is memory bound.

parXXL: Future GdX experiments

2006 objectives:

• Running $10^9$ cell simulations.
• Large scale pipeline algorithms (fluid dynamics) combining data and control parallelism.
• GdX “cluster” $\rightarrow$ GdX “Grid emulator” $\rightarrow$ Grid5000
  Use “wrekavoc” for heterogeneity emulation
  Use network emulation. Available?

2007 objectives:

• Experiments of complete physic simulations (LMOPS/LPMIA).
• Experiments on Grid5000.
parXXL: Conclusion & Perspectives

- A framework to implement fine grained applications on coarse grained architectures
- January 2006: scales up to 310 PEs (relaxation benchmark)
- Optimized runtimes for shm mainframes & for clusters

Future:
- Implement global IO functions & Hybrid cell communications
- Extend the cellular network library (par::cellnet)
- Experiment on Grid5000 (French national Grid)
- Application to optic & hot plasma simulations (ITER project)


parXXL: Fine Grained Simulation
Experiments on GdX

Questions?