CHIST-ERA Projects Seminar
Day 2, Cross Topics
Heterogeneous Distributed Computing

Hugh Leather

Bern, April 29th, 2016
Heterogeneity and Distribution In Action

This presentation was written by these people:
Oh, no! No More Moore Anymore

Dennard Scaling

Power $\propto$ area
Dennard Scaling

Oh, no! No More Moore Anymore

Power $\propto$ area

Dennard Scaling
1974-2005
Oh, no! No More Moore Anymore

Moore’s Law

Transistor count doubles every 2 years
Moore’s Law

1965-2015

TSMC: Per transistor cost rises in 2015!

Oh, no! No More Moore Anymore
Heterogeneity to the rescue

Match the program to the hardware

Fat CPU
Heterogeneity to the rescue

Match the program to the hardware

Fat CPU

Small CPUs
Heterogeneity to the rescue

Match the program to the hardware

Fat CPU

GPU

Small CPUs
Heterogeneity to the rescue

Match the program to the hardware

- Fat CPU
- Small CPUs
- GPU
- FPGA
Heterogeneity to the rescue

But doesn’t fit on one machine

All different

Even different interconnect
The BIG PROBLEMS

- Too hard to program
  - CPUs, GPUs, FPGAs - complex interactions
  - Massive distribution - complex network
  - Must optimise at multiple scales
  - Only experts can play

Today
The BIG PROBLEMS

- Too much energy
- ~20MW each, 1.5% globally, growth exponential

Today

A Modern Data Centre

Data Centre Electricity Demands In the US (source: Energy Star)

Billion kiloWatt hour/year

2001  2005  2009  2013  2017

$3 billion
Heterogeneous Distributed Computing

- Machines internally heterogeneous
- Machines heterogeneous to each other
- Massive distributed networks
- Networks heterogeneous
- Very hard to program
- If we don’t get it right =>
  
  energy/performance disaster
The Projects

**HPDCJ**
Heterogenous Parallel Distributed Computing Computing in Java

**DIONASYS**
Declarative and Interoperable Overlay Networks, Applications to Systems of Systems

**DIVIDEND**
Distributed Heterogeneous Vertically Integrated Energy Efficient Data centres
# The Projects

<table>
<thead>
<tr>
<th></th>
<th>HPDCJ</th>
<th>DIVIDEND</th>
<th>DIONASYS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programming Model</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Dependability</strong></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Data Management</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Versatility</strong></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>**Optimisation</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td><strong>Techniques</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Distributed Techniques</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
High Performance Computing

- Parallel distributed computing in Java
  - PCJ library for parallel computing in Java
- Scalability up to 6000 cores
- CPU and GPGPU
- Fault Tolerance

- Easy for non expert programmers
  - New approach to teach students
System as a first-class component (holon)

- Generative programming
- Application in IoT
- System Composition

Self organising overlays
Prototypes Open Sourced
DIVIDEND

- Vertical integration
- Programming model
- Energy accounting
- Auto tuning
- More heterogeneity
- Fast networks

Prototypes Open Sourced
Already saving 22% energy
Write a program, then system automatically

- Chooses the right hardware
- Or creates new hardware
- Optimises everything
- It is easy for the programmer
In 5 Years we need

- Programming models for major domains
- DSLs to specialise to all devices (CPU, GPGPU, FPGA)
- Eliminate waste in computing
- SDN needs to be transparent the application
Roadmap

In 10 Years we need

- Universal languages for the masses
- Tool chains to co-design platforms and fabricate logic/network/memory blocks for services
- Programming without knowing what’s out there
CHIST-ERA’s Role

Provide new calls

- Parallel programming
- Energy optimisation
- Automatic hardware synthesis
Conclusion

● Energy/performance crisis looming

● Can’t program and optimise HDC

● We are making progress on this

● Need more calls on this