Condensing the cloud: running CIEL on many-core

Malte Schwarzkopf  Derek G. Murray  Steven Hand
University of Cambridge
Condensing the cloud: running CIEL on many-core

Malte Schwarzkopf    Derek G. Murray    Steven Hand

University of Cambridge
Condensing the cloud: running CIEL on many-core

Malte Schwarzkopf  Derek G. Murray  Steven Hand

University of Cambridge
Condensing the cloud: running **CIEL** on **many-core**

Malte Schwarzkopf  Derek G. Murray  Steven Hand

University of Cambridge
Task graph
Elastic resources
Elastic resources

Fault tolerance
Elastic resources

Fault tolerance

→ Sequential code
Elastic resources

Fault tolerance

Sequential code

Deterministic parallelism
Suitable for many-core?
Many-core workers in a cluster?
CIEL: dynamic task graphs

• Allow tasks to spawn more tasks
CIEL: dynamic task graphs

- Allow tasks to spawn more tasks
CIEL: dynamic task graphs

• Allow tasks to spawn more tasks
Test environments

4x AMD Opteron 6168 „Magny-Cours“
- dodeca-core
- 1.9 GHz (64-bit), 512 KB L2, 12 MB L3
- 64 GB DDR3 @ 1333 MHz

Intel Single-Chip Cloud (SCC)
- tetracontakaiocta-core
- 533 MHz (32-bit P54C), 16 KB L1, 256 KB L2
- 64 GB DDR3 @ 800 MHz

timespin $\mu$-benchmark

T

$n$ tasks

$S_1 \quad S_2 \quad \ldots \quad S_n$
timespin μ-benchmark

\[ T \rightarrow S_1 \rightarrow S_2 \rightarrow \ldots \rightarrow S_n \rightarrow T' \]

\( n \) tasks
timespin $\mu$-benchmark

\[ T \xrightarrow{M} T' \]

\[ n \text{ tasks} \]

Relative overhead

\[ r = \frac{M}{t} \]
timespin on unmodified CIEL

rel. overhead

less is better

41.6x

5.1x

1.3x

1.04x
timespin, using lighttpd

- Rel. overhead
  - 1.1x
  - 1.59x
  - 2.03x

Number of cores: 47, 44, 41, 38, 35, 32, 29, 26, 23, 20, 17, 14, 11, 8, 5, 2

Seconds: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Less is better
timespin, using lighttpd

- 2.03x
- 1.59x
- 1.1x
- 1.05x

rel. overhead

less is better

number of cores

seconds
timespin, using lighttpd

relative overhead

less is better

1.1x

2.03x

1.59x

1.05x

number of cores

seconds
timespin, shared object store

relative overhead

seconds

number of cores

1.6x

1.3x

1.06x

1.03x

less is better
User code

master process

worker process

47 worker threads

Object store

Hardware
timespin, using multi-worker

less is better

1.27x

4.5x

1.51x

1.05x
Challenges and Opportunities

Contention vs. sharing
Challenges and Opportunities

Contention vs. sharing

I/O multiplexing
Challenges and Opportunities

Contention vs. sharing

I/O multiplexing

Fault tolerance
Conclusions and summary

• Investigated performance of the CIEL many-core

• Works unmodified, but fine-grained tasks suffer

• Started to address various challenges

• **Next**: multi-scale version for hybrid clusters

http://www.cl.cam.ac.uk/netos/ciel/