Multi-scale computing over heterogeneous resources

Malte Schwarzkopf
Steven Hand

Resource heterogeneity

... within a compute cluster:

Different hardware generations
Deliberate diversification
Per-task variance

... between general computing devices:

But: currently, resources are treated as homogeneous.
- Assume all VMs/machines have equal performance on a task.
- Pessimal performance: slowest resource dominates.
- No support for combination of weak and powerful devices.

How can we support distributed computations over heterogeneous resources?

Scales of parallelism

Different tools to deal with parallelism at different scales:

Increasing heterogeneity

Low-level message passing
OS-level threading
Coarse-grained message passing
Distributed task parallelism

But: we cannot run a general computation across these scales.
- Need to write specific code for each scale.
- Parallelism at one scale may limit parallelism at others.

How can we unify these different scales in a generalized model of parallelism?

The solution

Task-parallel computing over resource ensembles.

- Collections of resources:
  - Internally homogeneous.
  - Same scale of parallelism.
- Can nest \((A \subseteq B \subseteq C)\) and peer \((C-D)\) to form hierarchy.
- Have local coordinator who manages jobs & tasks within.

Supporting heterogeneity via:

- Self-description
- General profiling
- Job-specific profiling

Jobs start in an ensemble, but tasks can traverse boundaries.

Spawn rate exceeds completion rate:
- Job runtime decreases
- Resource utilization increases
- Ensemble-local scheduling implies locality

But:
- General profiling
- Job-specific profiling

Digraph or Hierarchy

Example

- Ensemble A
- Ensemble B
- Ensemble C
- Ensemble D

Open questions and challenges

Choosing a task granularity
Compiler pass (LLVM) to generate tasks at multiple scales

Quantify benefit of boundary transitions
Performance modelling, extrapolation from historical measurements

Avoid fan-in contention at ensembles
Gossip current spawn/completion rates and queue lengths

Expected results

- Job runtime decreases
- Resource utilization increases

But: currently, resources are treated as homogeneous.
- Assume all VMs/machines have equal performance on a task.
- Pessimal performance: slowest resource dominates.
- No support for combination of weak and powerful devices.

http://www.cl.cam.ac.uk/~ms705/