Dynamically Adapting Communication Behavior of Parallel Applications

Lars Ailo Bongo

(larsab@cs.uit.no)

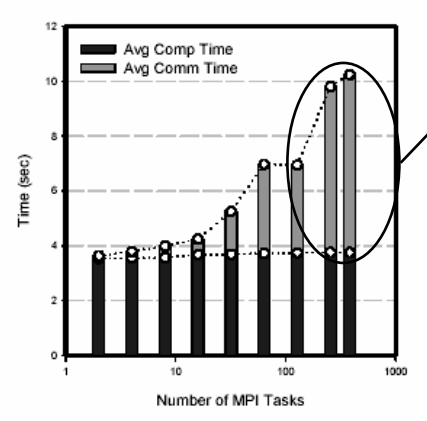
Work done with: Otto J. Anshus, John Markus Bjørndalen and Brian Vinter

http://www.cs.uit.no/forskning/DOS/hpdc/

NOTUR Emerging Technologies - Cluster

- Collaboration between:
  - NTNU
    - Department of Computer and Information Science (Project leader Anne C. Elster, Torbjørn Hallgren, and students)
    - Department of Mathematical Science (Einar Rønquist)
  - University of Tromsø
    - High Performance Distributed Computing group (Otto J. Anshus, John Markus Bjørndalen, Lars Ailo Bongo, Tore Larsen, Daniel Stødle and Brian Vinter).
    - Computing Center (High Performance Computing Group).

Scalability problem



Area of interest

Sweep3D: a solver for the 3-D, time-independent, particle transport equation on an orthogonal mesh.

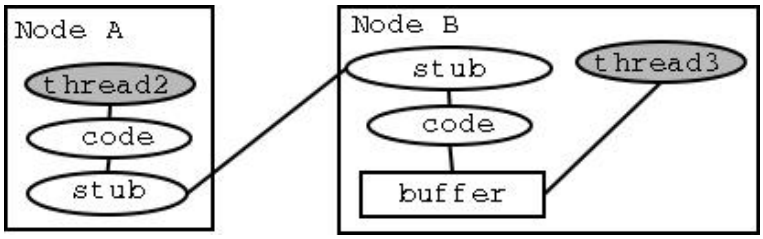
Figure from: Jeffrey S. Vetter and Andy Yoo. An Empirical Performance Evaluation of Scalable Scientific Applications. *Supercomputing 2002*.

Communication performance becomes increasingly important as applications scale up (in the figure weak scaling is used).

Our approach to improve communication performance

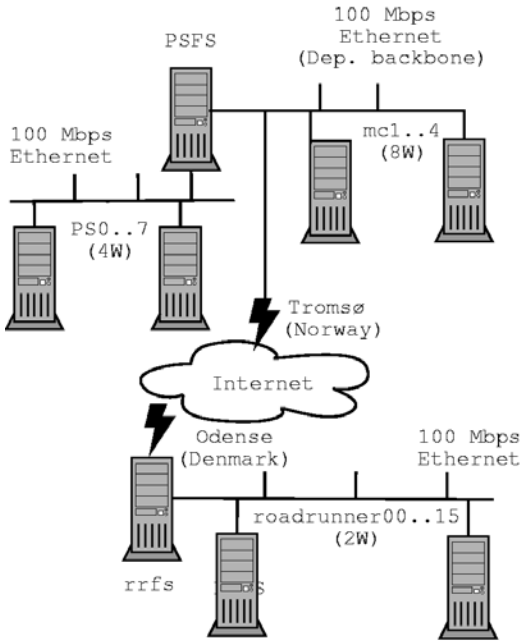
- Adapt applications communication structure and behavior.
- Requires:
  - A system for configuring and mapping the application to the given cluster topology and architecture (PATHS).
  - A system for collecting communication behavior data (EventSpace).
  - An approach for analyzing and finding communication behavior problems (EventSpace & work in progress).
  - An approach for reconfiguring the communication behavior (work in progress).

A simple *communication path* specified using PATHS

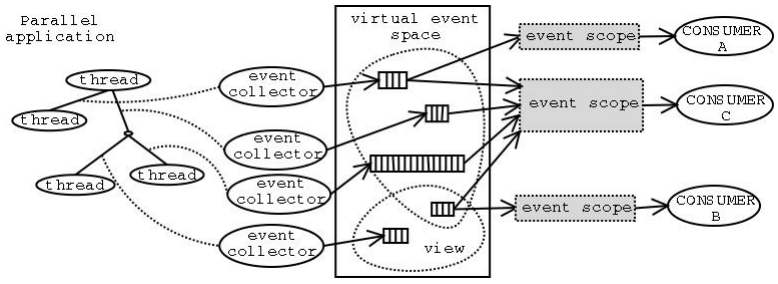


- Specify how threads communicate and synchronize.
- Specify what is *computed* where.
- Specify what is *stored* where.

Clusters used

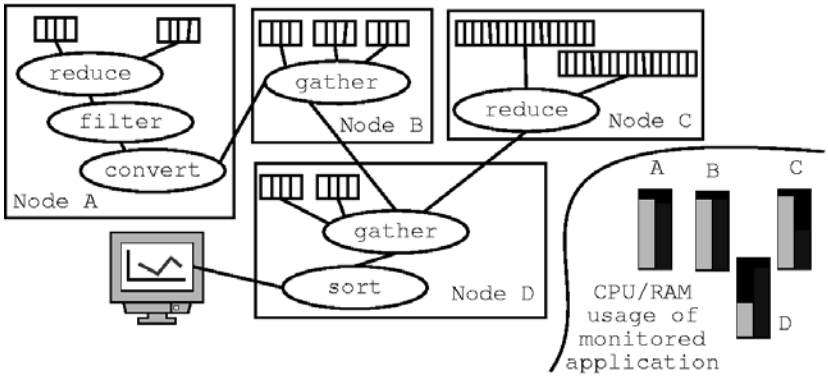


EventSpace architecture



A parallel applications is configured using PATHS. Multiple consumers observe collected data through event scopes providing different views of the communication behavior.

An event scope

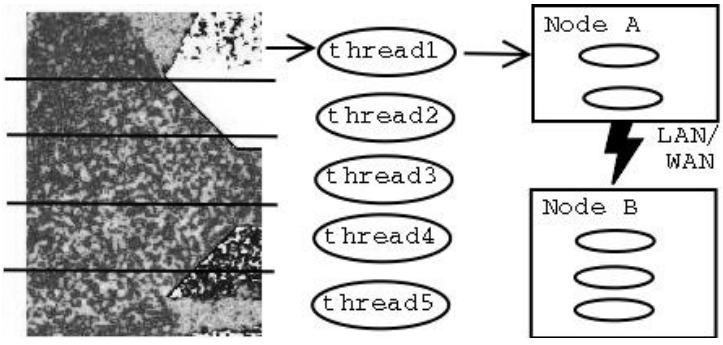


In the figure, the most computing intensive parts of the event scope path are mapped to the node with most available CPU. Also more data is collected on the node with most available memory.

EventSpace perturbation experiments

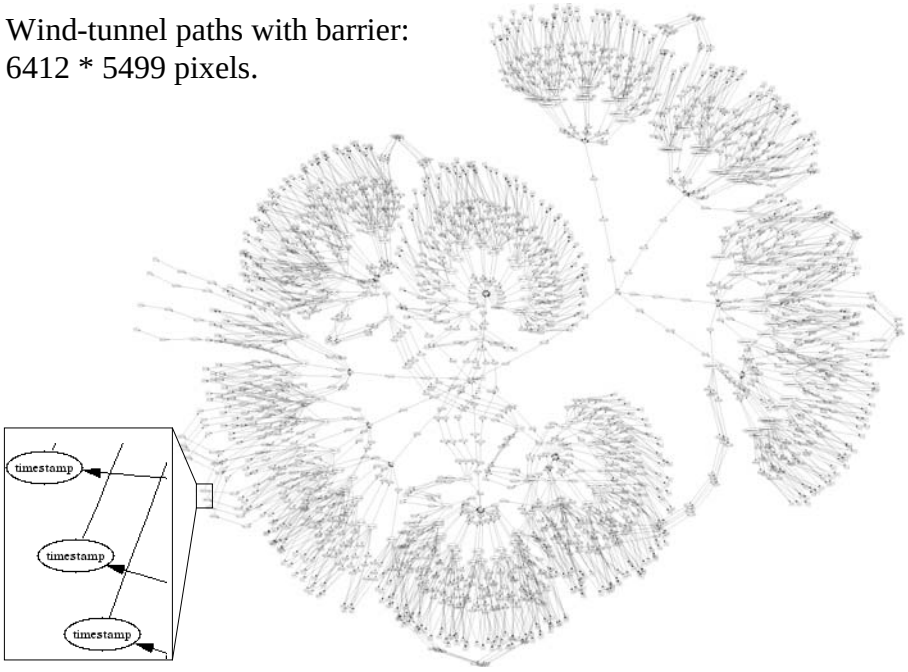
Application/ benchmark	**Event collectors** (slowdown)	**Event scopes** (slowdown)
Wind-tunnel	0.99 – 1.01	1.00 - 1.08
Ping-pong	1.05	1.09 (total 1.14)
Global reduction Benchmark	1.04 – 1.14	Work in progress

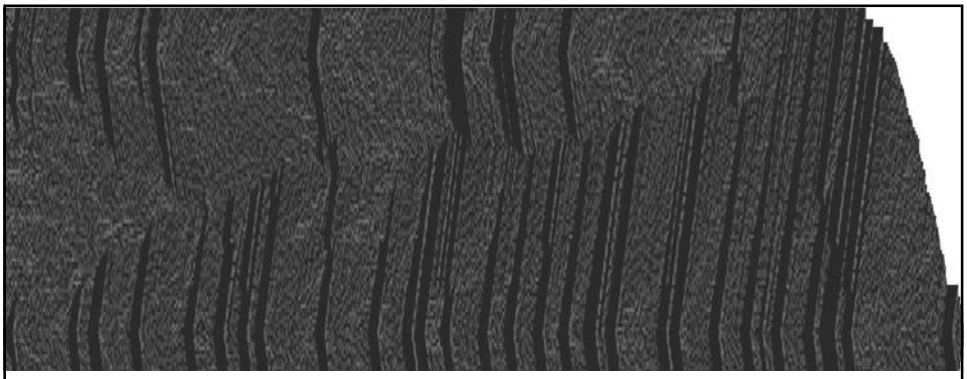
Case study: Wind-tunnel



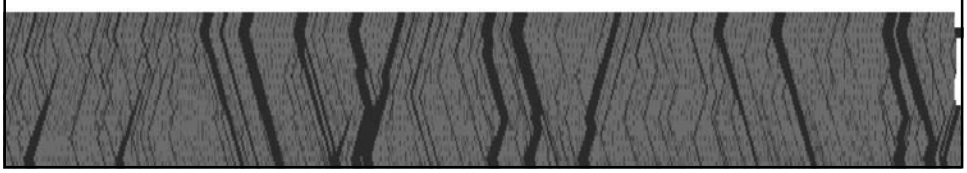
The wind-tunnel applications is implemented using data parallelism, and is configured and mapped to the available nodes.

Wind-tunnel paths with barrier:
6412 * 5499 pixels.

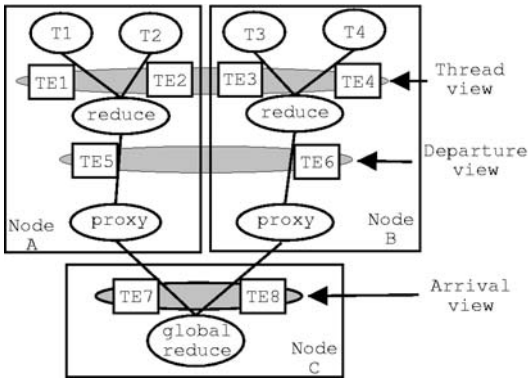




A *view* showing when a thread communicates (red) and computes (green). There is one horizontal line per thread.
Above: 4 threads per CPU. Below: 1 thread per CPU.



Collective communication performance analysis



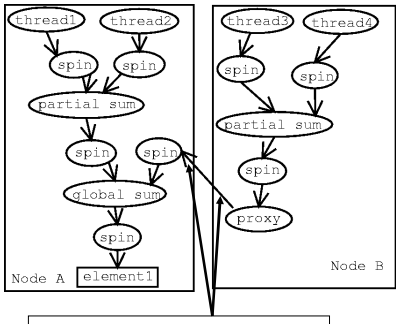
Different EventSpace views of a collective communication tree mapped to a cluster.

- Low-level analysis.
- Large amounts of data.
- Requires system wide analysis?
- Work in progress.

Event scope performance experiments (using wind-tunnel)

View	**View size**	**Observe rate**
Communication-computation	10080 bytes	8.4 Hz
Thread (col. op.)	5040 bytes	161 Hz
Arrival and departure (col. op.)	576 bytes	3.3 Hz (Python impl.)
Single wrapper	36 bytes	2000 Hz
Node	10260 bytes	4.2 Hz
Cluster	40824 bytes	2.2 Hz

Adding spin-wait to communication operations



Spin for coscheduling (not shown).

- Can we add spinning to communication paths to improve performance?
  - Achieve coscheduling?
  - Decrease synchronization conflicts?
  - 'Shake-up' global communication behavior leading to performance problems?
- Work in progress (presently not successful).

Future work: adaptable collective communication

- Can we build a system that:
  - Allows run-time reconfiguration of collective communication paths.
  - And, does not add a (large) performance penalty for collective operations.

• An article about the EventSpace system will appear at the *Euro-Par 2003* conference.
• An article about using EventSpace for performance analysis of collective operations is submitted to *Performance Tools 2003*.
• The EventSpace views were visualized using a display wall at the NORSIGD Seminar in Computer Graphics 2003 (Fornebu).