Distributed Objects for Parallel Programing

Adam Beguelin
adamb@cs.cmu.edu
School of Computer Science
Carnegie Mellon University
and
Pittsburgh Supercomputing Center

Dome: Distributed Object Migration Environment

- Ease of Programming
- Dynamic Load Balancing
- Architecture Independent Checkpoint/Restart

Project Members
- Adam Beguelin, adamb@cs.cmu.edu
- Erik Seligman, eriks@cs.cmu.edu
- Mike Starkey, starkey@cs.cmu.edu

Distributed Programing Methods

- Message passing
  - PVM, MPI, Express, Isis, ...
- Shared memory models
  - Midway, Linda, ..
- Parallel languages
  - HPF, Dino, pC++, ...

Motivation

What is difficult about writing distributed applications?

- Data distribution
- Load balancing
- Fault tolerance
- Visualization

The Dome Approach

- Distributed objects encapsulate parallelism
- Loadbalancing hidden within objects
- Fault tolerance built into objects
- Visualization integrated into object

Ease of Use

- SPMD Programming
  (Single Program Multiple Data)
- Library of Distributed Objects
- No explicit message passing

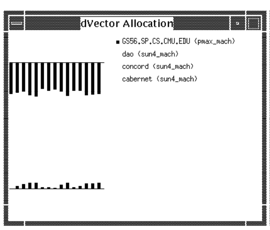
Dot Product Example

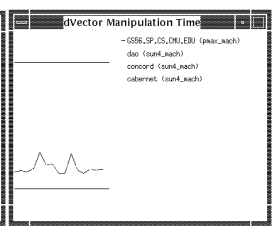
- Simple dot product using Dome dVectors

```
main()
{
    // Define the environment to have 5 tasks
    dome myenv("dprod", 5);

    // Define a few vectors
    dVector x(myenv, 1024);
    dVector y(myenv, 1024);
    dVector z(myenv, 1024);
    float sum;

    // These assignments happen in parallel
    x = 7;
    y = 13;

    // This multiply happens in parallel
    z = x * y;
    sum = z.gsum();
}
```


Dynamic Load Balancing at Runtime

- Load balancing dVectors





Failures Happen

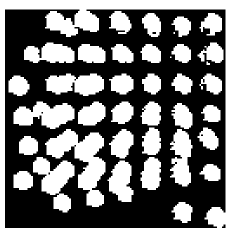
- Parallel programs use many machines
  - Chance of failure increases
- Parallel Programs are Long running
  - Chance of failure increases
- Heterogeneity important

Architecture Independent Fault Tolerance

- Checkpoints done at a high level
  - Dome objects
  - User defined state
- Checkpoints stored in external format
  - i.e. XDR
- Restarts done under user control
- Advantages
  - Small Checkpoints
  - Architecture Independence
- Disadvantages
  - Some work done by programmer
  - All tasks must restart

Molecular Dynamics Example

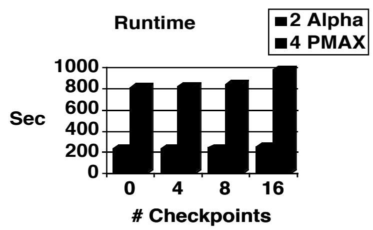
- Converted from CM Fortran
- Dome dVector Objects



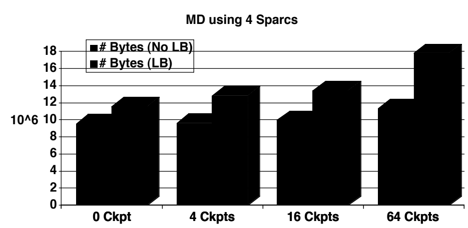
Preliminary timings

- Multiuser systems
- 256 Molecules, 10 iterations, Ethernet
- (10,000 iterations is minimal for real science)

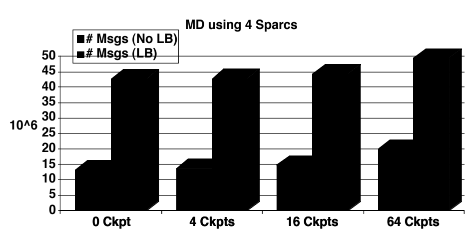
Time Cost of Checkpoints



Additional Data Transfer Cost



Additional Message Count Cost



Other Efforts

pC++

Language extensions for parallelism
Gannon et al., Supercomputing '93

DoPVM

Queueing Objects and Shared Objects
Hartley and Sunderam, Intl. Parallel Proc. Symposium 93

CC++

Chandy and Kesselman, Caltech

LAPACK++

C++ interface to LAPACK linear algebra package
Pozo, Dongarra, and Walker, OON-SKI 93

Future Work

- Sophisticated load balancing
- Distributed checkpoints
- More objects
- Parallel I/O
- Other language interfaces