Parallel Software Technologies Laboratory PSTL

VolpexMPI: robust execution of MPI applications through process replication

Edgar Gabriel and Jaspal Subhlok

Department of Computer Science,
University of Houston

CS@UH

Parallel Software Technologies Laboratory PSTL

Contributors

- Collaborators:
  - David Anderson (UC Berkeley),
  - Margaret Cheung (UH Physics),
  - Rong Zheng (UH Computer Science)
- Students:
  - VolpexMPI:
    Rakhi Anand, Troy Leblanc
  - Volpex Dataspace:
    Girish Nanadagudi, Eshwar Rohit, Hien Nguyen

CS@UH

Outline

- Introduction and Motivation
- VOLPEX project overview
- VolpexMPI
  - design and concept
  - performance results on a homogeneous cluster
  - target selection problem
  - performance results on a 'heterogeneous' cluster
- Overview of ongoing work

Volpex: Parallel Execution on Volatile Nodes

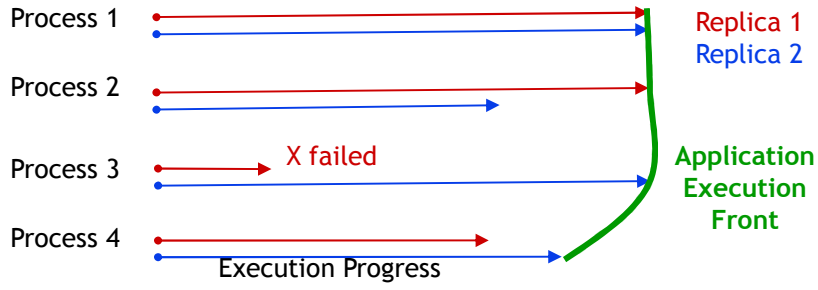
- Fault tolerance: why ?
  - Node failures on machines with thousands of processors
  - Node and communication failure in distributed environments
  - Very long running applications
  - Security relevant applications
- Volpex Project Goals:
  - Execution of communicating parallel programs on volatile ordinary desktops
  - Key problem: High failure rates AND coordinated execution

Major Challenges in VOLPEX

- Failure Management
  - Replicated processes
  - Independent process checkpoint/recovery
- Programming/Communication Model
  - Volpex Dataspace API
  - VolpexMPI
- Execution management
  - Selection of "good" nodes for execution
  - Integration with BOINC/Condor
  - Simulation to identify suitable codes (Dimemas)

The Volpex Approach

Redundancy and/or independent checkpoint/restarts
→ *multiple physical processes per logical process*



Volpex Goals:
- Application progress tied to the fastest process replica(s)
- Seamless progress despite failures
- Minimum overhead of redundancy

Dataspace Programming Model

- Independent processes communicate with one way, PUT/GETs with an abstract dataspace
  - Similar to Linda, Javaspaces, Tspaces etc.

`PUT (tag, data)` place data in dataspace indexed with tag
`READ (tag, data)` return data matching the tag
`GET (tag, data)` return and remove data matching tag

- Fault tolerance approach (checkpoint or replication) implies redundant processes/execution
  - ➔a logical PUT/GET may be executed many times
  - ➔a late replica may PUT a value that is out of date

VolpexMPI

- MPI library for execution of parallel application on volatile nodes
- Key features:
  - controlled redundancy: each MPI process can have multiple replicas
  - Receiver based direct communication between processes
  - Distributed sender logging
- Prototype implementation supports ~40 MPI functions
  - point-to-point operations (blocking and non-blocking)
  - collective operations
  - communicator management

Point-to-point communication

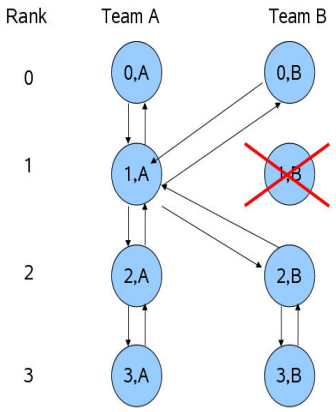
- Goal: efficient handling of multiple replicas for each MPI process
  - avoid sending each message to all replicas
- Concept:
  - receiver based communication model
    - sender buffers message locally
    - receiver contacts sender process requesting message
    - sequence numbers used for message matching in addition to the usual message envelope (tag, communicator, sender rank, recv rank)
  - no support for `MPI_ANY_SOURCE` as of today

Volpex MPI design

- Data transfer based on non-blocking sockets
  - supports timeout of messages and connection establishment
  - handling of failed processes
  - adding of new processes at runtime
- Sender buffer management:
  - circular buffer containing message envelopes and data
  - oldest log-entries are being overwritten
  - size of the circular buffer limits as of today ability to retrieve previous messages

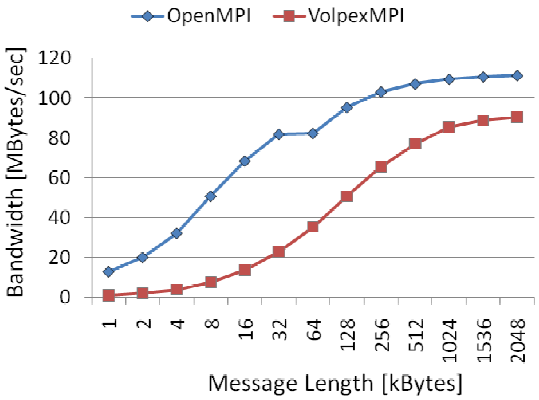
Managing Replicated MPI processes

- Team based approach:
  - Processes are spawned in teams
  - Only in case of failure, processes from different team is contacted
  - Optimal for homogeneous environments



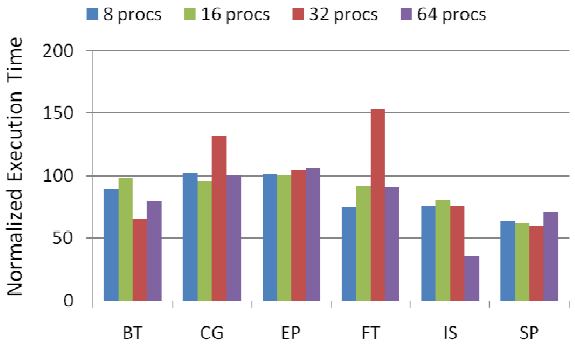
Bandwidth comparison

- 4 byte latency over Gigabit Ethernet:
  - Open MPI v1.4.1: ~50us
  - VolpexMPI: ~1.8ms



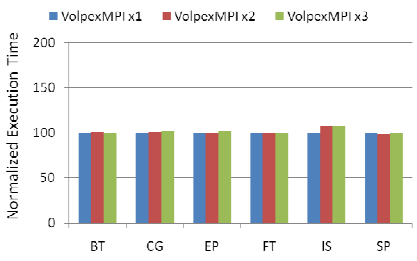
NAS Parallel Benchmarks

- Normalized execution times of VolpexMPI on a dedicated cluster over Gigabit Ethernet
- Open MPI v1.4.1 reference times are 100

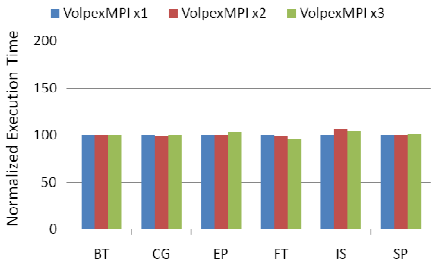


Influence of redundancy level

- Performance impact of executing one (x1), two (x2) and (x3) replicas of each process
- Normalized to the single redundancy VolpexMPI execution times



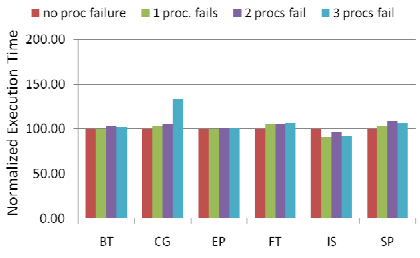
8 processes



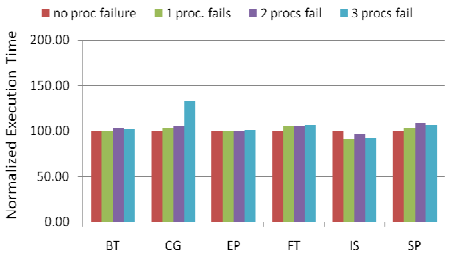
16 processes

Influence of process failures

- Double redundancy
- Failing processes from both teams
- Normalized to the double redundancy execution times



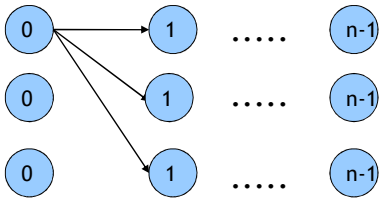
8 processes



16 processes

The Target Selection Problem revisited

- Identifying best set of replicas



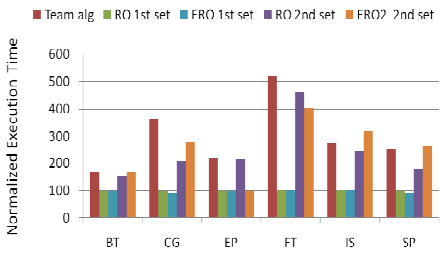
- Beneficial to connect to fastest replica
- Will make fast replica slow by making it handle more number of requests

Target Selection Algorithms

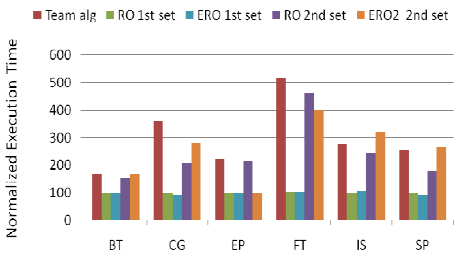
- RO: Response Order Algorithm
  - Request a message from all replicas of a given MPI rank
  - Target is selected based on response order of replicas
  - Regularly repeated during execution
- ERO: Extended Response Order Algorithm
  - Same preliminary steps as RO
  - Change to next (slower) target in the list if difference in newest sequence number for a particular message exceeds a given threshold

Target Selection Algorithms (II)

- Double redundancy tests on a heterogeneous configuration
  - fast nodes: Gigabit Ethernet, 2.2 GHz
  - slow nodes: Fast Ethernet, 1.0 GHz
- Initially, both teams contain processes on fast and slow nodes
- Each MPI rank has one fast and one slow process
- Normalized towards double redundancy numbers on GE



8 processes



16 processes

PARALLEL SOFTWARE TECHNOLOGIES LABORATORY PSTL

Beyond volunteer systems

- Experiments over InfiniBand in the planning
  - using RDMA Get operation would improve performance
  - requires changes in the message logging
- Beyond the MPI API
  - Functions to compare values of a variable across replicas
  - API allowing to perform operations on a single copy of a replica
    - e.g. result file written by a single replica of a process
  - API allowing to split execution of an operation across all replicas

CS@UH

PARALLEL SOFTWARE TECHNOLOGIES LABORATORY PSTL

Summary

- Volpex MPI allows for the seamless handling of multiple process replicas of MPI process
  - minimal or no performance penalty due to replication
  - seamless handling of process failures
  - different target selection algorithms for homogeneous and heterogeneous environmets
- Application have to carefully chosen for volunteer computing
  - communication/computation ratio
  - low degree of communication

CS@UH


Intel Core i7 975 Extreme Processor BX80601975 - 3.33GHz, LGA 1366, 6.4GT/s QPI, 8MB L3 Cache, Quad-Core, HyperThreading, Bloomfield, Retail
Item Number: I69-0975
Model: BX80601975
Availability: Order Today, Ships Today
List Price: $1,319.99
Instant Savings: - $220.02
Price: $1,099.97
Protect Your Investment
Choose From Extended Service Plans as Low as $207.98
Quantity 1
ADD TO CART
Add To My WISH LIST
More Intel Products
Very good.....until it melts.....
Reviewer: obsolete_power on Dec 06, 2009
Customer Rating: 1.8
Value 1.0
Features 3.0
Quality 1.0
2.0
I bought this CPU as an upgrade for my video editing suite and after a week of use, my computer just shut off and I started smelling burnt silicone. I inspected my computer and found that half the socket was molten. I do NOT recommend this CPU because of its obviously poor design. I already wrote a letter to Intel about this. Guess I should have used a CPU fan?
FAIL
failblog.org
CS@UH