BOINC Workshop 11

Hien Nguyen, Eshwar Rohit
University of Houston

Supervisors:
Dr. Jaspal Subhlok
University of Houston
Dr. David P. Anderson
SSL – U.C, Berkeley
Enable BOINC to efficiently support apps that require interprocess communication.

Goals:
- Easier programming for communicating applications
- Reduce execution time (not increase throughput)
Example Applications

REMD Protein Folding application
Each process runs a standard molecular simulation at different temperature

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
<th>P8</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEP-1</td>
<td>270</td>
<td>280</td>
<td>290</td>
<td>300</td>
<td>310</td>
<td>320</td>
<td>330</td>
<td>340</td>
</tr>
<tr>
<td>STEP-2</td>
<td>280</td>
<td>270</td>
<td>290</td>
<td>300</td>
<td>320</td>
<td>310</td>
<td>340</td>
<td>330</td>
</tr>
<tr>
<td>STEP-3</td>
<td>280</td>
<td>290</td>
<td>270</td>
<td>300</td>
<td>320</td>
<td>340</td>
<td>310</td>
<td>330</td>
</tr>
</tbody>
</table>
Example Applications

- Or many other applications:
  - Differential equation solvers (grid) (synchronous)
  - Game playing with alpha/beta pruning (asynchronous)
  - Search application.
  - ..... 

- Suitable applications: low to moderate amount and frequency of communication.
DIFFICULTIES

Job execution

Fast host

Slow host

Synchronization point

Worse as number of hosts increases or longer apps

Overall execution speed

Slow down overall execution speed

Hien Nguyen
University of Houston
1. Volpex Dataspace  
   • IPC for volunteer environment
2. Ensuring Efficiency  
   • Process management  
   • Host selection
3. Experiments And Evaluation
4. Future Work
Volpex Dataspace

- **Dataspace**: global shared space that processes can use for information exchange without a temporal or spatial coupling.

```
Volpex Dataspace Server

Put(ABC, 800)

Get(ABC,?) (800)
```
Volpex Dataspace – Fault Tolerance

Volpex DSS is unique in supporting redundant Put/Get operations

replicated

Put(ABC, 800)

Get(ABC, ?) (800)
Volpex MPI:

- An MPI library designed for executing parallel applications in volunteer environment.
- Direct communication between processes.
- Key Features
  - Controlled redundancy
  - Receiver based direct communication
  - Distributed sender based logging

ENSURING EFFICIENCY

Parallel program executes at the speed of the slowest process

- Process management
  - Simultaneous process starting
  - Failure and recovery
  - Replica management
  - Checkpoint/restart

- Host selection.

- Integrated with BOINC
Job execution scheme

BOINC Scheduler

Volpex Dataspace Server

Get work

Put data item

Get checkpoint

Put checkpoint

Get data item
Simultaneous process starting:

- All processes start computation together

- Volpex jobs have highest (infinite) priority: uninterruptible by other jobs.

- While waiting for all processes of a Volpex job to be ready: host can do other finite priority volunteer jobs.

- Use of boinc_temporary_exit()
Failure and recovery

Dead instance spotted by heartbeat mechanism: process instances regularly send heartbeat to Volpex DSS.

Slow instance detected when very old checkpoint commit attempted.

“Hot Spare” replaces the dead/slow process. Degree of replication maintained.
Hot spare policy:

- Fast replacement
- Hot spare group
- BOINC Scheduler
- Volpex Dataspase Server
Checkpointing:

- Process instance commits and uploads checkpoints to Volpex DSS (only stores latest checkpoint for each process).

- `Volpex_time_to_checkpoint()`  
  `Volpex_checkpoint(char* checkpoint)`

- Restarted process instance requests checkpoint from Volpex DSS.
Criteria for selecting volunteer hosts to assign to a Volpex job:

- CPU speed
- Memory capacity
- Disk space
- Upload bandwidth
- Predicted availability
Future availability prediction based on
- Last valued predictor: availability in the last hour
- Predictability: the number of availability changes in the past 2 weeks.

In essence: select hosts which change availability very rarely.

Method partly based on: Exploiting Non-Dedicated Resources for Cloud Computing Artur Andrzejak, Derrick Kondo, David P. Anderson. (NOMS10)
IMPLEMENTATION STATUS

- Volpex utilities: for scientists to submit, abort or query status of a Volpex job.

- Modified BOINC scheduler: includes host selection for Volpex job.

- Modified Volpex DSS: handles new type of requests, manages application execution.
EXPERIMENTS AND EVALUATION

Experiment Scheme:

- Application: Sieve of Eratosthenes, REMD Protein Folding
- Number of processes: 10 for Sieve and 16 for Folding
- Level of replication: 1, 2, 3
- Pool of around 80 nodes: On campus & global
- Host Selection Policies:
  - Random/Naïve
  - Minimum threshold on CPU, memory, disk space, upload bandwidth and predicted availability
Random Host Selection
10 Processes

Replication helps reduce variability in job’s execution time
For on-campus nodes, replication does not have remarkable effect.
Minimum threshold for host selection helps stabilize the execution time even w/o replication.
REMD for Protein Folding

Experiment

Time (minutes)

- Rep = 1, random selection
- Rep = 2, random selection
- Rep = 1, threshold based selection
- Rep = 2, threshold based selection
More experiments:
- Higher number of processes
- More applications
- Communication pattern (local/global, synch/asynch)
- Size and frequency of communication

More in depth study on:
- Host selection policy
- Optimal checkpoint interval

Granting credits to hosts:
- Granting credit to hot spares
- Granting credit to host that fails mid-way
If you have application?

We would be happy to help you in employing Volpex for your application.

Students / postdocs interested in working on Volpex welcome to contact us.

Our team contacts:

• Dr. Jaspal Subhlok: jaspal@uh.edu
• Dr. David Anderson: davea@ssl.berkeley.edu
• Dr. Edgar Gabriel: gabriel@cs.uh.edu
• Hien Nguyen: hien.nguyen.nx@gmail.com
• Eshwar Rohit: eshwar.rohit@gmail.com
• Rakhi Anand: rakhi@cs.uh.edu
• LaToya Green: latoya1987@msn.com

Our Website: http://volpex.cs.uh.edu
Thanks for listening!
IMPLEMENTATION STATUS

Scientist submits job specs to BOINC Scheduler.

Scheduler requests creation of a job and WU from Database.

Database dynamically creates result from WU.

Volpex Dataspace Server requests procID from Database.

Database returns procID to Volpex Dataspace Server.

Scheduler receives heartbeat and checkpoint from Volpex Dataspace Server.

ProcID of failed instance is retrieved and replaced in Hot Spare Group.