Enabling Synchronous Computation on Volunteer Computing Grids

Carlos Varela, cvarela@cs.rpi.edu
Rensselaer Polytechnic Institute
http://wcl.cs.rpi.edu/

<table>
<thead>
<tr>
<th>Faculty</th>
<th>Graduate Students</th>
</tr>
</thead>
<tbody>
<tr>
<td>Malik Magdon-Ismail, CS</td>
<td>Nathan Cole, Travis Desell</td>
</tr>
<tr>
<td>Heidi Newberg, AstroPhys</td>
<td>Kaoutar El Maghraoui, Ph.D., 2007</td>
</tr>
<tr>
<td>Bolek Szymanski, CS</td>
<td>Wei-Jen Wang, Ph.D., 2006</td>
</tr>
</tbody>
</table>

INRIA BOINC Workshop
Grenoble, France
September 11, 2008
Research Challenges/Directions

• **Applications:**
  Astroinformatics, particle physics, bioinformatics, virtual surgical planning, fluid dynamics, aeronautical design, climate modeling, etc.

• **Middleware** bridging the gap:
  Decentralized coordination; scheduling with inaccurate, partial knowledge about distributed resources; fault-tolerance; security; programmability; high performance
  ✓ Programming/coordination abstractions/models/languages.
  ✓ Decentralized resource management protocols.
  ✓ Integrated static and dynamic analyses techniques for performance prediction and evaluation.
  ✓ Heterogeneity-tolerant failure-oblivious algorithms.

• **Infrastructure:**
  Supercomputer centers, virtual grids, campus grids, grids in a box, planetary-scale networks
Milky Way Origin and Structure

- **Problem Statement:**
  What is the structure and origin of the Milky Way galaxy?
  How to analyze data from 10,000 square degrees of the north galactic cap collected in five optical filters over five years by the Sloan Digital Sky Survey? (over 10Tb in images)

- **Applications/Implications:**
  Astrophysics: origins and evolution of our galaxy.

- **Approach:**
  Experimental data analysis and simulation
  To use photometric and spectroscopic data for millions of stars to separate and describe components of the Milky Way

- **Software:**
  Generic Maximum Likelihood Evaluation (GMLE) framework.
  MilkyWay@Home BOINC project.

Ben Moore, Inst. Of Theo. Phys., Zurich

Kathryn V. Johnston, Wesleyan Univ.
Virtual Surgical Planning

- **Investigators:**
  K. Jansen, M. Shephard (RPI), C. Taylor, C. Zarins (Stanford)

- **Problem Statement:**
  How to develop a software framework to enable virtual surgical planning based on real patient data?

- **Applications/Implications:**
  Surgeons will be able to virtually evaluate vascular surgical options based on simulation rather than intuition alone.

- **Approach:**
  Scan of real patient is processed to extract solid model and inlet flow waveform.
  Model is discretized and flow equations solved.
  Multiple alterations to model are made within intuitive human-computer interface and evaluated similarly.

- **Software:**
  MEGA (SCOREC discretization toolkit)
  PHASTA (RPI flow solver).
Adaptive Partial Differential Equation Solvers

- **Investigators:**
  J. Flaherty, M. Shephard B. Szymanski, C. Varela (RPI)
  J. Teresco (Williams), E. Deelman (ISI-UCI)

- **Problem Statement:**
  How to dynamically adapt solutions to PDEs to account for underlying computing infrastructure?

- **Applications/Implications:**
  Materials fabrication, biomechanics, fluid dynamics, aeronautical design, ecology.

- **Approach:**
  Partition problem and dynamically map into computing infrastructure and balance load.
  Low communication overhead over low-latency connections.

- **Software:**
  Rensselaer Partition Model (RPM)
  Algorithm Oriented Mesh Database (AOMD).
  Dynamic Resource Utilization Model (DRUM)
Map of Rensselaer Grid Clusters
Maximum Likelihood Estimation on RPI Grid and BlueGene/L Supercomputer

- 2 Minute Evaluation
- MLE requires 10,000+ Evaluations
- 15+ Day Runtime

- ~100x Speedup
- 1.5 Day Runtime

- ~230x Speedup
- <1 Day Runtime

Computation Time (seconds)

- MPI/C
- SALSA/Java

Comparison of computation times for different scenarios:

- 1x1 OPT
- 32 - 4x2 OPT
- 40 - 4x1 OPT
- 32 - 4x2 OPT
- 40 - 4x1 OPT
- 128
- 256 - Virtual
- 512
- 1024 - Virtual
MilkyWay@Home: Volunteer Computing Grid

About MilkyWay@Home

MilkyWay@Home is a research project that uses Internet-connected computers to do research in modeling and determining the evolution of the Milky Way galaxy. You can participate by downloading and running a free program on your computer.

MilkyWay@Home is based at the Rensselaer Computer Science Department. This particular project is being developed to better understand the power of volunteer computer resources.

- EOS: Internet Operating System
- DGE: Distributed Genetic Likelihood Evolution
- Worldwide Computing Laboratory
- Science paper on Observed and Generic Likelihood Evolution

Join MilkyWay@Home

- This project uses BOINC. If you're already running BOINC, select Attach to Project. If not, download BOINC.
- When prompted, enter http://milkyway.cs.rpi.edu/milkyway/
- If you're running a command-line or pre-5.6 version of BOINC, create an account first.
- If you have any problems, get help here.

Returning participants

- New account - view stats, modify preferences
- Teams - create or join a team
- Contact
- Applications

Community

- Participant profiles
- Message boards

Project totals and leader boards

- Top participants
- Top computers
- Top teams
- Other statistics

More Information

- Server Status

Powered by BOINC

User of the day

UPT - Terry

It's a Bong addict and crunch 24/7 on quite a few projects and I'm a proud member of the UK Bong Team please come join us.

News

- March 24, 2008 Increased Granted Credit
- Due to many complaints that the reduction in credit was a bit too severe, we've increased the generated credit a bit. It should be around 4 credit per KI now. Please feel free to keep complaining about credit on the forums and we'll eventually get it right. :)

- March 22, 2008 Reduced Granted Credit
- Due to the fact that the application has been highly optimized in the last few versions, most people agree that the granted credit is way too high for the project by a factor of 3. We've scaled credit to keep it in line with other projects, so if you notice reduced credit this is why.

- March 20, 2008 Assembler: Updated and New Work Available
- I've updated the assembler to deal with some of the performance issues we've been having with it. You should see more work availability and hopefully we won't have results piling up that aren't assimilated. I've started a bunch of new searches and hopefully the work will keep flowing faster than before.

- March 19, 2008 Forums Down for Upgrade
- I just upgraded the BOINC server software. The database is a little shaky right now as we're trying to upgrade to PHP5 to facilitate it better. Therefore the forums will be down until we can get PHP5 up, hopefully it won't take more than a day. However, the assimilator problem seems to be fixed with this release and hopefully so will the freezes.

- March 19, 2008 Work Generation Downtime
- We're running a purge on the database to try and get things sped up. During this time we won't be generating any new work, but this should be completed sometime tonight and there will be new work after that.

...more

News is available as an RSS feed GML
Generic Maximum Likelihood Estimation

Specify Data and Models

Analyse New Results

GMLE
Determines search types and execution environments, performs distribution and optimization

Asynchronous
Asynchronous Genetic Search
Asynchronous Particle Swarm
Asynchronous Differential Evolution
Asynchronous Newton Method

The Internet

Partially Asynchronous
Asynchronous Simplex
Asynchronous Gradient Descent
Parallel Genetic Search
Parallel Particle Swarm
Newton Method

Grids

Synchronous
Gradient Descent
Simplex
Other Traditional Search Methods

Supercomputers

Synchronous Software Architecture

Scientific Models
- Data Initialisation
- Integral Function
- Integral Composition
- Likelihood Function
- Likelihood Composition

Search Routines
- Gradient Descent
- Genetic Search
- Simplex
- ...

Evaluation Request
Results

Evaluator Creation
Evaluator
Evaluator
Evaluator
Evaluator
Evaluator
Evaluator

Distribute Parameters
Combine Results

SALSA/Java (RPI Grid)
MPI/C (BlueGene)

Distributed Evaluation Framework
Asynchronous Search Strategies

• Use an asynchronous search methodology
  • No explicit dependencies
  • No iterations

• Continuously updated population
  • N individuals are generated randomly for the initial population
  • When a evaluator requests more work, stochastically apply operators to members of the population to create new parameter sets to evaluate
  • When results arrive, update the population

T. Desell, B. Szymanski and C. Varela,
“Asynchronous Genetic Search for Scientific Modeling on Large-Scale Heterogeneous Environments”, Heterogeneous Computing Workshop (IPDPS) 2008.

Asynchronous Software Architecture

Scientific Models
- Data Initialisation
- Integral Function
- Integral Composition
- Likelihood Function
- Likelihood Composition

Search Routines
- Evolutionary Methods
- Genetic Search
- Particle Swarm Optimisation
- ...

Evaluator Creation

Evaluator (1)

Evaluator (N)

BOINC (Internet)

SALSA/Java (RPI Grid)

Distributed Evaluation Framework

Initial Parameters

Optimised Parameters
Synchronous Volunteer Computing Challenges

- **Security**
  - Clients need to communicate with each other
  - A malicious client gaining control over other clients could cause a DDOS attack or damage other clients

- **Correctness**
  - One bad client can ruin the computations of other clients it is computing with

- **Volutility**
  - How can faults be handled without slowing or invalidating collective computation?

- **Heterogeneity**
  - User clients can be using any operating system and architecture
  - Latencies between clients range from local to global times, how can computation be distributed efficiently?
## World Migrating Agent Example

<table>
<thead>
<tr>
<th>Host</th>
<th>Location</th>
<th>OS/JVM</th>
<th>Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>yangtze.cs.uiuc.edu</td>
<td>Urbana IL, USA</td>
<td>Solaris 2.5.1 JDK 1.1.6</td>
<td>Ultra 2</td>
</tr>
<tr>
<td>vulcain.ecoledoc.lip6.fr</td>
<td>Paris, France</td>
<td>Linux 2.2.5 JDK 1.2pre2</td>
<td>Pentium II 350Mhz</td>
</tr>
<tr>
<td>solar.isr.co.jp</td>
<td>Tokyo, Japan</td>
<td>Solaris 2.6 JDK 1.1.6</td>
<td>Sparc 20</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Local actor creation</td>
<td>386us</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local message sending</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAN message sending</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAN message sending</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAN minimal actor migration</td>
<td>150-160 ms</td>
<td>240-250 ms</td>
<td>25-30 s</td>
</tr>
<tr>
<td>LAN 100Kb actor migration</td>
<td>3-7 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAN minimal actor migration</td>
<td>2-3 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAN 100Kb actor migration</td>
<td>30-60 ms</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Middleware/IOS

- **Middleware**
  - A software layer between distributed applications and operating systems.
  - Alleviates application programmers from directly dealing with distribution issues
    - Heterogeneous hardware/O.S.s
    - Load balancing
    - Fault-tolerance
    - Security
    - Quality of service

- **Internet Operating System (IOS)**
  - A decentralized middleware framework for adaptive, scalable execution
  - Modular architecture to evaluate different distribution and reconfiguration strategies

Application Topology-Sensitive Work-Stealing (ATS)

- A specialization of work stealing to collocate distributed application components that communicate frequently

- Process/actor migration to minimize high latency communication, based on
  - Location of acquaintances
  - Profiled communication history

- Tries to minimize the frequency of remote communication improving overall system throughput

Sparse Application Topology

Throughput (Messages Processed)

Time (seconds)

ARS Throughput
RS Throughput
Round Robin
Network Topology-Sensitive Work-Stealing (NTS)

- An extension of ATS to take the network topology and performance into consideration
- Periodically profiles end-to-end network performance among peer nodes
  - Latency
  - Bandwidth
- Tries to minimize the cost of remote communication improving overall system throughput
  - Tightly coupled actors stay within reasonably low latencies/high bandwidths
  - Loosely coupled actors can flow more freely

The Internet Operating System Middleware

Original Application

Reconfigurable Capabilities
- Migration
- Split & Merge
- Checkpointing

Reconfigurable Application

Application profiling

IOS API

Profiling Module
- Performance profiles
- Interfaces to resources profilers

Decision Module
- Evaluates the expected gain of a potential reconfiguration

Protocol Module
- Sends and receives work steal requests

IOS Agent

Initiate a work steal request when there are enough local resources

9/17/08
Case Study: Parallel Decomposition of Heat Distribution

Legend:
- Ghost Cells
- Data Cells
- Boundary Cells
- Ghost Cell Exchange
- 4-pt update stencil

Image Courtesy of Elaine Wenderholm
Impact of Process Granularity

Experiments on a dual-processor node (SUN Blade 1000)
Example: Split and Merge Operations

Parallel Decomposition

Legend
- Ghost Cells
- Data Cells
- Boundary Cells
- Ghost Cell Exchange
- 4-pt update stencil

Original Data Space

Split

Merge
Malleable Applications

- **Small process granularity**
  - Generally enables better OS memory hierarchy utilization (data fits in L2 cache)
  - Incurs in higher context switching overhead

- **Large process granularity**
  - Requires less scheduling and context switching
  - May not use memory hierarchy efficiently (cache misses)

- **Split and merge** functionality enables application processes to dynamically change granularity
  - to improve **performance**
  - to **scale** to larger run-time environments


Security / Correctness--- Potential Directions

• **IOS uses the Java platform**
  - Can leverage JVM sandbox, bytecode authentication

• **Additional security measures**
  - Clients only accept communication from signed/trusted neighbors
  - Secure communication between applications (public/private key encryption)

• **Profiling**
  - [Client and Server Side] Detect malicious anomalous clients through stochastic result verification and redundancy
  - [Client Side] Keep a “trust” measure of p2p neighbors and share this knowledge, i.e., a web of trust.
Volatility --- Potential Directions

- **Stochastic Replication**
  - Used for verification/correctness/trust management
  - Also can be used for fault tolerance
    - Replication must be done within neighbors sharing the same shared computation

- **Algorithmic Fault-tolerance**
  - Heterogeneity-tolerant failure-oblivious asynchronous algorithms
  - Randomized algorithms
  - Peer group checkpointing strategies

- **Fault-tolerant Programming Models**
  - Lightweight process/data checkpointing/replication

Heterogeneity --- Potential Directions

- Java based clients are designed to run on any OS/hardware platform
  - Is performance a significant concern? (e.g., for GPU/hybrid architectures)

- Synchronized computation requires similar compute times among workers
  - IOS can profile and perform dynamic reconfiguration/load balancing
  - Profiled knowledge of clients (e.g., multi-core architectures) can allow for heterogeneity-aware work scheduling and distribution
  - Group clients in similar geographical areas (low inter-client latencies) into virtual clusters to minimize communication overhead
Discussion

• Security of clients and servers is extremely important in volunteer grid computing – different middleware strategies can enable secure distributed computation.

• Adaptive middleware can use profiling, replication and reconfiguration (migration, malleability) to enable synchronous computing on dynamic volunteer computing grids.
Merci!

Software and publications freely available at: http://wcl.cs.rpi.edu/

Please join MilkyWay@Home at: http://milkyway.cs.rpi.edu/