V-BOINC: The Virtualization of BOINC

Gary A. McGilvary
Malcolm Atkinson
Adam Barker
Ashley Lloyd
INTRODUCTION

Berkeley Open Infrastructure for Network Computing

Your PC

1. get instructions
2. download applications and input files
3. compute
4. upload output files
5. report results

Projects Servers
INTRODUCTION

BOINC Client

• Attach to project >> organize and monitor application execution >> return results

• BOINC Client Environment: boinccmd, BOINC Manager, Screensaver

```
boinccmd [--host hostname] [--passwd passwd] command

boinccmd --project_attach http://setiathome.berkeley.edu f2j923r9j23rmd2d2d
boinccmd --project http://setiathome.berkeley.edu allowmorework
boinccmd --project http://setiathome.berkeley.edu update
boinccmd --project http://setiathome.berkeley.edu nomorework
boinccmd --project http://setiathome.berkeley.edu reset
boinccmd --project http://setiathome.berkeley.edu detach
```
INTRODUCTION

BOINC Client

• Attach to project >> organize and monitor application execution >> return results
• BOINC Client Environment: boinccmd, BOINC Manager, Screensaver
INTRODUCTION

**BOINC Client**

- Attach to project >> organize and monitor application execution >> return results
- BOINC Client Environment: *boinc*cmd, BOINC Manager, Screensaver
INTRODUCTION

BOINC Server

• Distributes scientific applications and collect results (and more...)

• BOINC Server Environment: BOINC Server Source Code, MySQL, PHP and Apache

• Daemons:


http://garymcgilvary.co.uk
INTRODUCTION

• BOINC downfalls:

  - Applications must be ported to each machine architecture
  - Project server must be trusted/secure
  - Applications must contain checkpointing code
  - Applications with dependencies difficult to execute
V-BOINC

• V-BOINC is the virtualized version of BOINC
  - Run BOINC applications on virtual machines rather than on the host machine

• The V-BOINC Solution. Use virtualization to:
  - Avoid multi-platform application porting
  - Run applications in a sandbox environment
  - Provide system-level checkpointing
  - Allow applications with dependencies to run (easily) under BOINC
RELATE WORK

- Similar Research Projects:
  - BOINC vboxwrapper
    - does not deal with application dependencies
    - requires additional server modifications to configuration files
    - requires specific configuration details (e.g., folders must be named etc)
    - virtual machine snapshots are not considered
  - CERNVM
    - only operates with LHC experiments.
    - a volunteer user cannot select the project they wish to attach to
    - virtual machine snapshots are not considered
    - server implementation not publicly available

CernVM is a baseline Virtual Software Appliance for the participants of CERN LHC experiments. The Appliance represents complete, portable and easy to configure user environment for developing and running LHC data analysis locally and on the Grid, independently of Operating System software and hardware platform (Linux, Windows, MacOS). The goal is to remove a need for the installation of the experiment software and to minimize the number of platforms (compiler-OS combinations) on which experiment software needs to be supported and tested.
OUTLINE

1. Introduction
2. Design
3. Implementation
4. Evaluation
5. Conclusions/Future Work
V-BOINC COMPONENTS

Distributes Virtual Machines

- boinccmd (modified)
- V-BOINC Manager (GUI)

Regular BOINC Project Server

Application Dependency Disk (DepDisk)
V-BOINC ARCHITECTURE

0. Install V-BOINC Client

http://garymcgilvary.co.uk

Modified BOINC Client (no need to download regular BOINC)

Perform regular BOINC tasks and controlling VM and VM BOINC Client

Java GUI (.jar file)
V-BOINC ARCHITECTURE

To use V-BOINC, please enter the V-BOINC Server URL and Weak key as well as the BOINC Server URL and Weak key.

Press 'Attach to Project' within the Local BOINC Controls to attach to the V-BOINC project entered.

Once the virtual machine has downloaded and is running, V-BOINC will attach to your desired BOINC project entered in the BOINC Server URL field.

One may then control the Virtual Machine, Local BOINC and Virtual Machine BOINC clients using the respective controls.
V-BOINC ARCHITECTURE

1. Request VM/Download Dependencies

boinccmd --project_attach http://107.21.228.163/VBOINC 167dm32mc00x123dqwq2

wget -r http://setiathome.berkeley.edu/download/application_disks/

Assumption: BOINC project developers must make .vdi publicly available
2. Obtain VM, Executable and Disk

- Decompresses virtual machine
- Signals to V-BOINC client to take control
- Waits instruction to complete by until V-BOINC
- Returns result and log to server
V-BOINC ARCHITECTURE

- BOINC Core Client
- BOINC Task
- Application Dependencies
- Downloaded Virtual Machine
- VBOINC Middleware
  - VirtualBox API
  - Resource Monitor
  - Failure Detection
- (Modified) BOINC Core Client
- VBOINC Client GUI
V-BOINC ARCHITECTURE

3. Mount the DepDisk or Create/Attach Fresh Disk then Setup VM

DepDisk exists: attach

No DepDisk: create virtual disk locally
VM ARCHITECTURE

Why create and attach a fresh virtual disk if no DepDisk exists?

Original VM Architecture:

Modified Architecture:
(on V-BOINC Server, transferred to volunteer)

Replaced Architecture:
(on volunteer host)
3. Mount the DepDisk or Create/Attach Fresh Disk then Setup VM

DepDisk exists: attach

No DepDisk: create virtual disk locally

Configure Network

Set user-specified memory, CPU cap etc
V-BOINC ARCHITECTURE

4. Start VM

V-BOINC Client waits until VM running and has IP address

Takes Periodic Snapshots

Availability VM Poll

Connects VM BOINC Client to BOINC project
5/6/7. Get/Execute Job and Return Results

Attach to Regular BOINC Project

Normal BOINC process applies
Typical BOINC Configuration:

- **apps**
  - SETI

- **download**
  - DATA
  - DATA

Scientific Executable executes scientific computation over data

V-BOINC Configuration:

- **apps**
  - VM

- **download**
  - tar
  - vbox

VM Executable decompresses and waits for VM to complete
CLIENT MODIFICATIONS

Requires large amount of source code modifications/additions!

Check for DepDisks
Register and control VM (start, stop, etc)
Create and mount DepDisks
Configure VM (format disk, install BOINC, etc)
Control VM BOINC Client
Receive input from user/interact with GUI
Take periodic snapshots
Check VM availability periodically

..........
# WHY VIRTUALBOX?

<table>
<thead>
<tr>
<th>Requirement</th>
<th>QEMU/KVM</th>
<th>VirtualBox</th>
<th>VMWare Player</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique IP Address Allocation</td>
<td>✓¹</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Headless VM</td>
<td>✓</td>
<td>✓</td>
<td>✓¹</td>
</tr>
<tr>
<td>Image Size &lt; 235MB (compressed)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Boot Time &lt; 20s</td>
<td>✓²</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Basic VM Control</td>
<td>✓</td>
<td>✓</td>
<td>✓¹</td>
</tr>
<tr>
<td>Remote Command Execution</td>
<td>✗</td>
<td>✓</td>
<td>✓¹</td>
</tr>
<tr>
<td>Checkpointing</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Portability (Mac &amp; Linux)</td>
<td>✓³</td>
<td>✓</td>
<td>X</td>
</tr>
</tbody>
</table>

1. additional configuration and/or installation required on host
2. only when used with KVM enabled
3. KVM component not available on Mac OS X

Tested on VirtualBox 4.1.8
EVALUATION

BOINC vs V-BOINC

- **Benchmarks**: Primes, Create5GB, CPU, Memory, I/O and Disk

![Graph comparing execution times for different benchmarks between Host, BOINC, VM, and VBOINC]
EVALUATION

**BOINC vs V-BOINC**

- **Use Case Application**: Simple Parallel R INTerface (SPRINT)
  - provides parallel functions of
  - based on MPI
  
  http://r-sprint.org

- **pcor**: parallel correlation (memory/compute-intensive)
  - \( p\text{cor}(t (x, y = x)) \)

- [Diagram of 321 samples and 11000 genes]
EVALUATION

BOINC vs V-BOINC

- **Use Case Application**: Simple Parallel R INTerface (SPRINT)
EVALUATION

The Effect of Checkpointing

- Storage space valuable on volunteer host

<table>
<thead>
<tr>
<th>Disk and memory usage</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk: use at most</td>
<td>10</td>
<td>GB</td>
</tr>
<tr>
<td>Disk: leave free at least</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Values smaller than 0.001 are ignored</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk: use at most</td>
<td></td>
<td>% of total</td>
</tr>
<tr>
<td>Tasks checkpoint to disk at most every</td>
<td>60</td>
<td>seconds</td>
</tr>
<tr>
<td>Swap space: use at most</td>
<td>40</td>
<td>% of total</td>
</tr>
<tr>
<td>Memory: when computer is in use, use at most</td>
<td>60</td>
<td>% of total</td>
</tr>
<tr>
<td>Memory: when computer is not in use, use at most</td>
<td>85</td>
<td>% of total</td>
</tr>
</tbody>
</table>

Snapshot consists of:
- copy of virtual machine settings
- current state of virtual disks (.vdi)
- current state of memory (.sav)

Periodic Snapshots: 1 per minute for 10mins per benchmark
## EVALUATION

### The Effect of Checkpointing

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Snapshot Time (s)</th>
<th>Memory Size (MB)</th>
<th>DepDisk Snapshot Size (KB)(D:36)</th>
<th>VM Snapshot Size (KB)(D:8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1.1779</td>
<td>86.9</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>Memory</td>
<td>1.7142</td>
<td>56.76</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>I/O</td>
<td>0.9425</td>
<td>43.57</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>Disk</td>
<td>24.6023</td>
<td>1126.4</td>
<td>54374.4</td>
<td>8</td>
</tr>
<tr>
<td>Primes</td>
<td>1.2153</td>
<td>98.1</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>SPRINT</td>
<td>31.4665</td>
<td>2334.72</td>
<td>36</td>
<td>8</td>
</tr>
</tbody>
</table>

CPU, Memory and I/O Applications suitable for V-BOINC
# EVALUATION

## The Effect of Checkpointing

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Snapshot Time (s)</th>
<th>Memory Size (MB)</th>
<th>DepDisk Snapshot Size (KB)</th>
<th>VM Snapshot Size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1.1779</td>
<td>86.9</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>Memory</td>
<td>1.7142</td>
<td>56.76</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>I/O</td>
<td>0.9425</td>
<td>43.57</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>Disk</td>
<td>24.6023</td>
<td>1126.4</td>
<td>54374.4</td>
<td>8</td>
</tr>
<tr>
<td>Primes</td>
<td>1.2153</td>
<td>98.1</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>SPRINT</td>
<td>31.4665</td>
<td>2334.72</td>
<td>36</td>
<td>8</td>
</tr>
</tbody>
</table>

Disk-intensive applications also suitable however computation time affected!
# EVALUATION

## The Effect of Checkpointing

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Snapshot Time (s)</th>
<th>Memory Size (MB)</th>
<th>DepDisk Snapshot Size (KB)</th>
<th>VM Snapshot Size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1.1779</td>
<td>86.9</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>Memory</td>
<td>1.7142</td>
<td>56.76</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>I/O</td>
<td>0.9425</td>
<td>43.57</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>Disk</td>
<td>24.6023</td>
<td>1126.4</td>
<td>54374.4</td>
<td>8</td>
</tr>
<tr>
<td>Primes</td>
<td>1.2153</td>
<td>98.1</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>SPRINT</td>
<td>31.4665</td>
<td>2334.72</td>
<td>36</td>
<td>8</td>
</tr>
</tbody>
</table>

Average storage available: 63.6 GB

The Computational and Storage Potential of Volunteer Computing

Wednesday, 15 May 13
# EVALUATION

## The Effect of Checkpointing

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Snapshot Time (s)</th>
<th>Memory Size (MB)</th>
<th>DepDisk Snapshot Size (KB)</th>
<th>VM Snapshot Size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1.1779</td>
<td>86.9</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>Memory</td>
<td>1.7142</td>
<td>56.76</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>I/O</td>
<td>0.9425</td>
<td>43.57</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>Disk</td>
<td>24.6023</td>
<td>1126.4</td>
<td>54374.4</td>
<td>8</td>
</tr>
<tr>
<td>Primes</td>
<td>1.2153</td>
<td>98.1</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>SPRINT</td>
<td>31.4665</td>
<td>2334.72</td>
<td>36</td>
<td>8</td>
</tr>
</tbody>
</table>

Memory/CPU-intensive application suitability dependent on user preferences
1. Introduction

2. Design

3. Implementation

4. Evaluation

5. Conclusions/Future Work
CONCLUSIONS

• Provides solutions to the drawbacks of regular BOINC
  - makes it easier for application developers
  - more secure for volunteer users

• Allows applications with dependencies to run (easily) under BOINC

• Virtualization overhead can be large (acceptable for volunteer computing?)!
  - performance-advantage trade off!
  - overhead of implementation negligible

• Storage required for snapshots low
  - V-BOINC particularly useful for current CPU-intensive BOINC applications

• Future work:
  - Various VM Operating Systems/decrease virtualization overhead
  - Test with different hypervisors /decrease virtualization overhead
  - Ad hoc Cloud Computing

Available at: http://garymcgilvary.co.uk
THANK YOU!

Questions?

gary.mcgilvary@ed.ac.uk