

Data Collection via BOINC

Divya Ramachandran, Yang Zhang & Steven Stanek
January 10, 2005
UC Berkeley

Motivation

- Projects require collection of large amounts of data
- Data from institutions may show trends that cannot be generalized to the behavior of all computer users
- Collection should represent the behavior of the diverse population of computer users

What is BOINC?

- Berkeley Open Infrastructure for Network Computing
- Dedicate idle resources on personal computers to scientific research
- Research projects can use the infrastructure to gain access to volunteered resources on personal computers

Why should we use BOINC?

- BOINC users are volunteering to share their computers with researchers
- Some BOINC projects have over thousands of users
- We can gather computer failure and usage data from these users
- This data will come from a diverse pool of personal computer users

Our BOINC projects

- Crash Collection
 - Windows users send us copies of the minidumps of their system and application crashes
- Resource Measurement
 - Every ten minutes, client measures the resource usage of the system, including CPU activity, bytes of free memory etc.

How BOINC works

- Client-server model
 - Users download the BOINC core client and register for projects of their choice
 - Clients send requests for *workunits* and application to project-specific server
 - Server sends workunits to client along with a time within which results should be returned
 - Client sends result when the workunit completes
 - Server handles the result, evaluates it and credits the client accordingly

Design of our BOINC projects

- BOINC was originally designed to distribute CPU-intensive computations across resources on PCs
- The BOINC core client uses a round robin scheme to alternate the CPU utilization between projects
- Our projects are not compute-intensive and can therefore run all the time and share the CPU with other projects

Crash Collection

- After an OS crash, Windows saves a snapshot of the stack, called a minidump
- After application crashes minidumps may be saved in temporary locations if the user chooses to send Microsoft an error report
- The BOINC crash collection application checks these locations for new minidumps every ten minutes and sends them back to the server
- The server collects the minidumps for each unique user machine

Resource Measurement

- The BOINC resource measurement application uses Windows performance data tools to measure the status of resources on the machine
- Measurements include those of available memory, CPU usage, number of processes running, etc.
- The resources are measured every 10 minutes, and a summary is sent to the server

Project Status

- Both projects are in initial stages of deployment
- Have at least 10-15 users at a given time
- Start with small pool of users to iron out bugs
- With more publicity we can have more users
- Visit <http://roc.cs.berkeley.edu/projects/boinc> to join!

Next steps for Crash Collection

- Collect more data via more users
- Analyze the minidumps
- Attach a survey to assess the general usage behavior of the PC user
- With permission, poll processes on user machines

Next steps for Resource Measurement

- Collect more data via more users
- Plot measurements over time as incentive for more users
- Analysis of data as it is collected
- Modify application for use on other operating systems
- New ties with Intel Labs at Berkeley

Conclusions

- Identified BOINC as a good way to reach general PC user pool to collect data about machines
- Projects for both crash collection & resource measurement applications have been started
- This is still a work in progress
- Need to continue to collect more data
- Should have form of ongoing analysis as the data collection keeps growing
- Questions, comments & feedback?