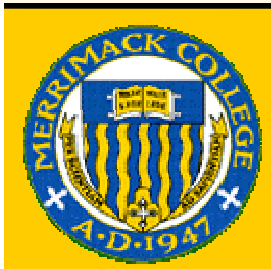
The Impact of Multi-Core Architectures on Task Retrieval Policies for Volunteer Computing

David Toth
Computer Science Department
Merrimack College
North Andover, Massachusetts, U.S.A.



Volunteer Computing

- Definition: Computing using resources on computers volunteered by people.
- “Supercomputer” for researchers who can’t afford one
- Projects
  - SETI@home – searches for extraterrestrial intelligence
  - GIMPS – searches for Mersenne primes
  - Folding@home – analyzes effects of protein folding
  - Grid.org – cancer, anthrax research
- Motivation: more projects, low participation rate

Task Distribution

- Task distribution may affect productivity
- Tasks have deadlines
- Tasks that are late are aborted
- Task distribution methods
  - Buffer No Tasks → Wasted time waiting to get next task
  - Buffer Multiple Tasks → Some tasks don't complete & get reassigned. Part of the time spent on uncompleted tasks could have been used to complete other tasks.

Contributions

1. Compare task retrieval policies under the more realistic assumption of multi-core CPUs

2. Show that for multi-core CPUs, policies that buffer tasks can perform almost as well as policies that do not buffer tasks – contrary to single-core CPUs

Task Distribution Policies

- Buffer None (in use – Grid.org)
- Buffer *n* Days of Tasks (in use – SETI@Home)
- Download Early (download next task at *n*% complete of current task – 95%)
- Buffer 1 Task

Methodology

- Calculate the expected number of completed tasks
  - Assumes each core works on a separate task without interfering with each other (linear scaling of past results)
- Determine number of tasks completed by simulation if each core works to complete a single task at a time

Methodology II

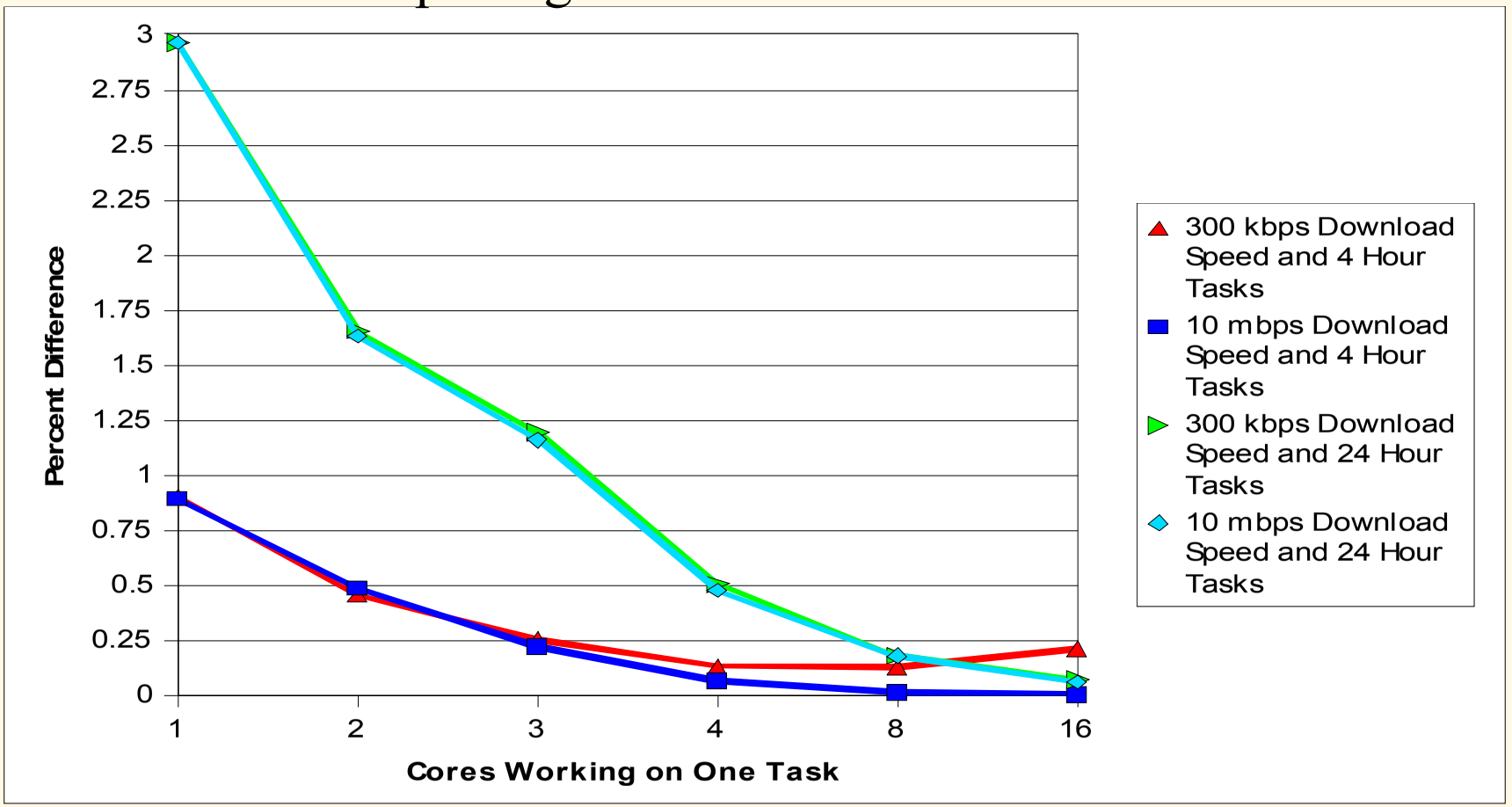
- Compare expected number to simulation results
- Use same traces and parameters as previous work
- Revise simulator to handle multi-core CPUs
- Simulate 2, 3, 4, 8, and 16 cores

Results

- Slightly better than linear increase in number of tasks completed in relation to number of cores used in 138 of 140 cases
- 2 cases where did not achieve predicted value were short tasks and slow download speed
- In general, more cores leads to higher percentage increase over expected number of tasks completed

Results

Difference Between the Number of Tasks Completed by Policies Completing the Most Tasks and the Fewest Tasks



Conclusions

- Multi-core CPUs can achieve slightly greater than linear increase in work completed when cores work together on a single task
- Difference between work completed by different policies is much less significant than it was for single-core CPUs
- Buffering policies much more attractive than before (less performance loss and fewer assumptions)
- Important to formulate tasks to allow multiple cores to work on them

Future Work

- Refine simulations for greater accuracy
- Study impact of green movement on task distribution policies
- Analyze trade-offs between screensavers, services, and web-based clients

Questions?

Thank you!

The Impact of Multi-Core Architectures on Task Retrieval Policies for Volunteer Computing

David Toth
Computer Science Department
Merrimack College
North Andover, Massachusetts, U.S.A.

david.toth@merrimack.edu