http://ccof.cs.uoregon.edu/ NCR-9714680

WaveGrid : A Self-organized Desktop Grid System for Fast Turnaround

Dayi Zhou and Virginia M. Lo
NRG Group, CIS Department, University of Oregon, Eugene,OR

What is WaveGrid?

A hybrid of desktop grid systems and peer-to-peer networks, taking the best from both worlds

- A lightweight Internet-wide cycle sharing system
- Self-organized using P2P principles
- Allows anyone to submit tasks
- Allows anyone to donate cycles
- Non-intrusive so people are willing to use it

What are the Challenges?

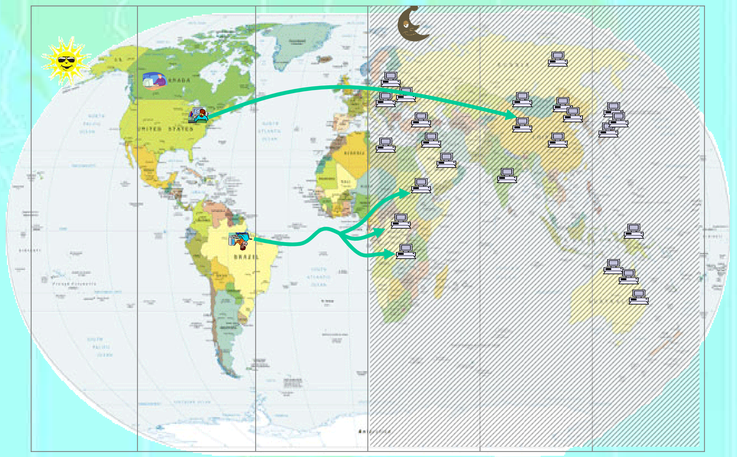
- Scalable discovery of idle hosts without central server as in SETI@Home; without institutional infrastructure as in Condor.
  - Large number of hosts
  - Unknown network topology
- Scheduling for Fast turnaround time when faced with volatile resources and imprecise resource information
  - Peers join and leave dynamically
  - Hosts withdraw cycles at any time

Fast Turnaround Scheduling in WaveGrid

- Organize host according to geographic information
- Timezone-aware resource discovery
- Migration from busy host to idle host

Cycle Sharing with WaveGrid

- Thousands of applications need cycles (gene sequencing, scientific simulation, ray tracing)
- Millions of compute cycles sit idle
- WaveGrid Rides the Wave of the Idle Hosts !



Details of Scheduling with WaveGrid*

- A timezone-aware structured overlay network
  - Uses the CAN Cartesian coordinate space (Sigcomm'01)
  - Divides CAN virtual space into wavezones, each representing several timezones
  - Hosts join a wavezone based on their timezones
- Clients schedule jobs onto hosts currently at night
  - Choose a random host in a night-time wavezone
  - Do expanding ring search to find more hosts
- Jobs migrate to next wavezone that just entered night-time, when current hosts are no longer available

*** (IPTPS'04, CCGrid'04, JSSPP'05)**

WaveGrid Preliminary Simulation Study

- Evaluate the performance of WaveGrid
- Compare the performance of migration schemes to no-migration scheme
- Evaluate the performances of different migration strategies

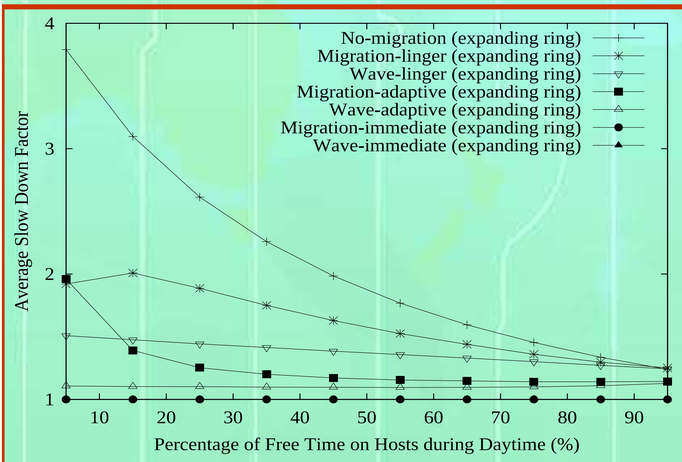
When to migrate / Where to migrate	Immediate	Linger	Adaptive (Only linger when cannot immediately migrate)
Random Host	**Migration-immediate**	**Migration-linger**	**Migration-adaptive**
Night-time Host	**Wave-immediate**	**Wave-Linger**	**Wave-adaptive**
Adaptive (Only use a random hosts when no available night-time hosts)	———	———	**Wave-adaptive**

Metrics

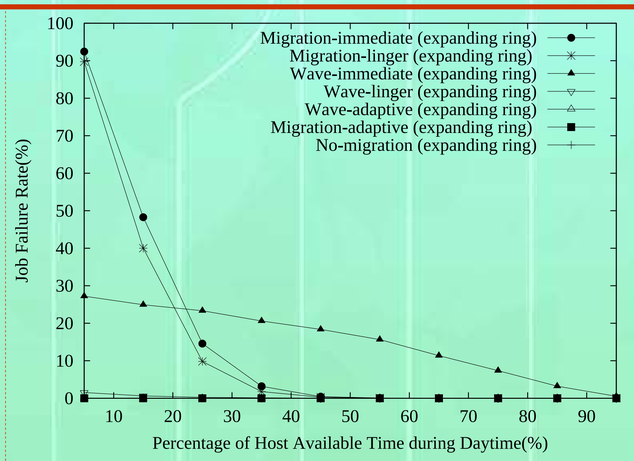
- **% of jobs that fail to complete (job failure rate):** # of jobs failed with first scheduling attempts over the total # of jobs submitted to the system.
- **Average slowdown factor**: turnaround time of the job over the job runtime (time to complete execution on a dedicated machine)
- **Average number of migrations per job**: # of times a job migrates during its lifetime in the system, averaged over all job that successfully complete execution

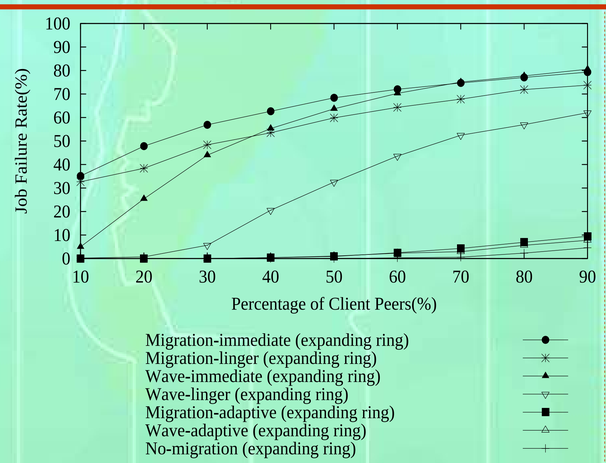
Conclusion

WaveGrid outperforms the others with low slow-down and minimal number of migrations.

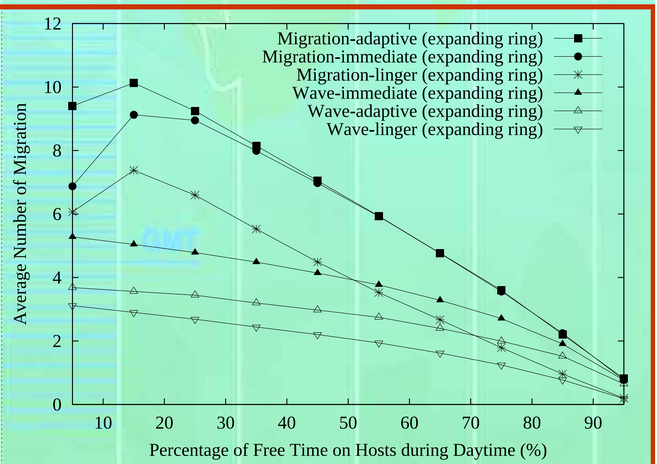


Turnaround time compared to a dedicated machine





Job failure rate for single scheduling attempt (no retry)



Migration Overhead