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

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!

Fast Turnaround Scheduling in WaveGrid
- Organize host according to geographic information
- Timezone-aware resource discovery
- Migration from busy host to idle host

WaveGrid Preliminary Simulation Study
- Evaluate the performance of WaveGrid
- Compare the performance of migration schemes to non-migration scheme
- Evaluate the performances of different migration strategies

<table>
<thead>
<tr>
<th>When to migrate</th>
<th>Immediate</th>
<th>Linger</th>
<th>Adaptive (Only linger when cannot immediately migrate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Host</td>
<td>Migration-linger</td>
<td>Migration-linger</td>
<td>Migration-adaptive</td>
</tr>
<tr>
<td>Night-time Host</td>
<td>Wave-linger</td>
<td>Wave-linger</td>
<td>Wave-adaptive</td>
</tr>
<tr>
<td>Adaptive (Only use a random host when no available night-time hosts)</td>
<td>Wave-adaptive</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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 jobs that successfully complete execution

Conclusion
WaveGrid outperforms the others with low slowdown and minimal number of migrations.