Los Alamos NATIONAL LABORATORY

# SchedMark:

## Evaluating Scheduler Performance across Platforms and Workloads

Eitan Frachtenberg
eitanf@lanl.gov
Modeling, Algorithms and Informatics Group (CCS-3)
Computer and Computational Science Division
Los Alamos National Laboratory

## Motivation

Many different process schedulers exist:
- In production operating systems:
  MS-Windows, Linux, Solaris/SunOS, IRIX, Tru64, QNX, BeOS,...
- In domain-specific patches/modifications:
  RTLinux, Con Kolivas' patchset, Ingo Molnar's SRT, SSST,...
- In research projects:
  [Etsion04], [Goel02], [Nieh97], [Rau99], [Snavely00], [Zheng04],...

Which raises many questions:
- How well do these schedulers perform?
- Which scheduler is better for continuous media? technical workloads? interactive programs? real-time? parallel programs? P2P programs?
- How do different schedulers react under increasing load?
- Which scheduler does well with hyperthreaded chips? multicore chips? Java phones? SMPs? How do they handle memory hierarchies?
- How can we quantify these issues?

To answer these questions, we are developing SchedMark:
- Allows direct evaluation of scheduler effect on applications, unlike current benchmarks that focus on kernel primitives, such as context-switch latency.
- Provides insights such as "Scheduler X is preferential to interactive applications as load increases" or "Scheduler Y doesn't handle parallel programs well".
- Designed for growth with future applications and architectures, with special emphasis on commodity parallel processors such as multicore chips.

## Challenges and Solutions

| Challenge | Solution |
|---|---|
| - Portability: different schedulers run on very different architectures and operating systems, so a scheduling benchmark is required to be meaningful with many different platforms. | + Workload is built around portable synthetic applications that emulate various real-world scheduling and resource requirements, such as multithreaded, batch, and interactive applications.<br>+ Minimize dependence on operating system by using a portable multithreading library.<br>+ Resource requirements of applications are calibrated across architectures to provide comparable workloads.<br>+ Parallel applications grow to fit number of processors. |
| - Validity: different applications care about different metrics (e.g., media players require responsiveness, but not throughput; parallel programs require coscheduling but not dynamic priorities). | + Applications measure their own metrics of interest e.g., dropped frames for movies, speedup for parallel programs.<br>+ Only relative metrics are used to facilitate comparison e.g., slowdown instead of response time, which compares better than absolute times across workloads/architectures. |
| - Resilience to workload: feedback effects and interactions between applications in a mixed workload produce a different picture than applications in isolation. | + Suite comprises static workloads and dynamic workloads. Static workloads help to measure the scheduling of different application classes in isolation and small groups. Long dynamic workloads with representative interarrival times paint a more realistic performance picture.<br>+ Measure a series of dynamic workloads with increasing offered load, to find saturation point and application-class specific issues. |

## Status

- Designed infrastructure for launching and timing a benchmark suite. The workload can be based on statistical distribution or a recorded trace.
- Characterization of scheduling and resource requirements of various commodity applications is underway.
- Early stages of implementation of synthetic applications and libraries.
- First release expected 2H 2005.

For more information...

Contact me at: eitanf@lanl.gov
Or search for "SchedMark"