Dynamic Adaptive Operating Systems -- I/O

Seetharami R. Seelam
Patricia J. Teller
University of Texas at El Paso
El Paso, TX
Dynamic Adaptability in Support of Extreme Scale

Goals

- Present a summary of our ongoing work
- Solicit assistance from the greater community
- Hear comments (+/-)
Enhanced Performance

Generalized \rightarrow \textbf{Customized} resource management

Fixed \rightarrow \textbf{Dynamically Adaptable} OS/runtime services
Project Challenges

Dynamic Adaptability in Support of Extreme Scale

Determining

• **What** to adapt
• **When** to adapt
• **How** to adapt
• **How** to measure effects of adaptation
Dynamic Adaptability in Support of Extreme Scale

- Identify adaptation targets
- Characterize workload resource usage patterns
- Determine/redetermine feasible adaptation ranges
- Define/adapt metrics/heuristics to trigger adaptation
- Generate/adapt monitoring, triggering and adaptation code, and attach it to OS

KernInst

- Monitor application execution, assessing performance (gain) and triggering adaptation as necessary

Project Methodology
Dynamic Adaptability in Support of Extreme Scale

Project Challenges

Determining

- **What** to adapt
- **When** to adapt
- **How** to adapt
- **How** to measure effects of adaptation
Dynamic Adaptability in Support of Extreme Scale

I/O Schedulers

I/O Subsystem

Schedulers

AS  Deadline  CFQ  NOOP

Schedulers/Controller

Storage System
Dynamic Adaptability in Support of Extreme Scale

Introduction

• Linux 2.6 provides four I/O schedulers:
  – anticipatory (AS)
  – deadline
  – completely fair queuing (CFQ)
  – noop

• Selection at
  – boot time: one scheduler for all drives
  – runtime: one scheduler per drive

• Default: AS or CFQ
Dynamic Adaptability in Support of Extreme Scale

Solution

Automatic and dynamic selection of scheduler or scheduling policy (within a monolithic scheduler)
Dynamic Adaptability in Support of Extreme Scale

Determining

- **What** to adapt - I/O Scheduler
- **When** to adapt - ??
- **How** to adapt - Explicit/Implicit Selection
- **How** to measure effects of adaptation - Execution Time (??)
Example Adaptation

Dynamic Adaptability in Support of Extreme Scale

- Automatic and dynamic scheduler selection
- Proof of concept with two schedulers: one providing bounded latency on requests, the other providing fair allocation of bandwidth (presented at OSIHPA Workshop)
Dynamic Adaptability in Support of Extreme Scale

ADO Scheduler

I/O Subsystem

Monitor → Scheduler Selection

DASS

Request completion feedback

CFQ

Deadline

Controller

Storage System

P₁, P₂, ..., Pₙ

R_d, R_b, W_d, W_b
Dynamic Adaptability in Support of Extreme Scale

Experimental Evaluation

Enforcing Bandwidth

- Disk utilization
- Draining time is a factor in reduced bandwidth
- tiobench: 2 GBs data in 4 KB blocks with 2 to 64 concurrent threads
- Several scheduler swaps
Implicit Selection

Dynamic Adaptability in Support of Extreme Scale

I/O Subsystem

Adaptive Scheduler

Deadline  CFQ  SCAN  NOOP

Adaptive Scheduler/Controller

Storage System
Goals -- Revisited

Dynamic Adaptability in Support of Extreme Scale

- Present a summary of our ongoing work
- Solicit assistance from the greater community
  - I/O scheduling -- Interesting Applications
  - VM -- Memory Hogs
  - Other areas
- Hear comments (+/-)
Acknowledgements

• We thank:
  – DOE (Grant # DE-FG02-04ER25622) and UTEP for financial support
  – IBM Corporation for IBM Shared University Research Grant (SUR)
  – SC|05 and Harvey Wasserman for creating and conducting this session
  – You for your interest
Dynamic Adaptability in Support of Extreme Scale

- Choosing an I/O Scheduler for Red Hat Enterprise Linux 4 and the 2.6 kernel, [http://www.redhat.com/magazine/008jun05/features/schedulers/](http://www.redhat.com/magazine/008jun05/features/schedulers/)
Dynamic Adaptability in Support of Extreme Scale

Questions / Thoughts?
seelam@cs.utep.edu
pteller@utep.edu

Thank You