SCAF: Scheduling and Allocation with Feedback
Tim Creech, Rajeev Barua

Inspired by work on AESOP, a parallelizing compiler for next-generation many-core satellites
- Need parallel programs for performance
- Need to run multiple programs
- How can we manage multiple parallel programs?

• AESOP already open-source:
• SCAF soon to be open-sourced
• Investigating SCAF extensions for space-sharing VMs

Previous systems:
Unmodified systems: each process assumes it can occupy an entire physical machine

Result: oversubscription, context-switching, hardware contention

Equi-partitioning: each process occupies an equal partition of the hardware contexts

Result: dedicated HW contexts, no switching. But what if Speedup_A(4) >> Speedup_B(4)?

SCAF: Perform on-line experiments to reason about speedups

Coordinate an improved allocation dynamically

Example: 3-way SCAF multiprogramming

Summary of results:
Mean improvement of 1.11-1.22X sum-speedup for 80-89% of pairs vs. EQ; 1.27-1.7X for 72-100% of pairs vs. unmod on 4 platforms

This work was supported by a NASA Office of the Chief Technologist’s Space Technology Research Fellowship, NASA Grant #NNX11AM99H

Systems and Computer Architecture Lab

Source: https://wiki.smartos.org/display/DOC/SmartOS+Virtualization

Source: http://aesop.ece.umd.edu

http://aesop.ece.umd.edu