Configuring the OS for Tomorrow’s Robots

Madhav Tummala, Daehyeok Kim, Joydeep Biswas, Aditya Akella

The University of Texas at Austin
Tomorrow’s Autonomous Mobile Service Robots

source: zdnet.com, robotics247.com

source: ieee.org, wikipedia.org

source: cleanpng.com

source: pngall.com
Configuring the Operating System

Applications
- NAV
- SLAM
- YOLO

ROS
- VFS
- cgroups
- Sched
- VMM

Device Drivers

Hardware

QoS settings
- buffer sizes
- congestion control algorithms

i/o queue depths
- cpu freq scaling
- power management

Hardware

scheduling algorithms
- priorities
- time slices
- cpu allocation

Page size
- transparent huge pages
- allocation policies
- replacement policies

Memory limits
- cpu shares
- swappiness
Why is configuration **difficult**?

1. Some knobs are per-process vs some are system-wide

2. Configuration for a application can have **counterintuitive** effects on performance of other applications

3. Optimal configuration depends on the **environment**

4. Global optimal configuration needs proper **objective function** defined over each application objective (slo)
Why should configuration change?

1. Environment (nature of input to the programs)
2. Contention (nature of load from each application)
Observations

frames per second
+ 99th percentile time

1. Scheduling algorithm for NAV to DEADLINE, with appropriate settings NAV ↑ YOLO ↓

2. Increasing CPU Shares for YOLO
NAV ↓ YOLO ↑

3. Adding memory limits for YOLO
NAV ↑ YOLO ↓

4. Transparent Huge Pages (system-wide)
NAV ↓ YOLO ↑
Learning-based approach

Dynamic configuration updates through an ML model
- Prototype model that ignores environment

![Diagram](image.png)
Thank You!

Experimental details, numbers, model performance,... etc!
Poster session later today (3:30 - 5:00 pm)

For other Learning in OS works - https://utns.cs.utexas.edu/
Robotics Lab - https://amrl.cs.utexas.edu/