EagleTree: Exploring the Design Space of SSD-Based Algorithms

Niv Dayan (ITU), Martin K. Svendsen (ITU), Matias Bjørling (ITU), Philippe Bonnet (ITU), Luc Bouganim (INRIA & UVSQ)

How do YOU evaluate algorithms for SSD-based systems?
- Undocumented black boxes → No performance model
- Opaque internal state → No experimental repeatability
- Behavior varies across devices → No general results
- SSD internals impact your algorithms → No control of SSD configuration (HW / SW)

YOU need a good command of SSD internals
- Internal IOs (garbage-collection & wear-leveling) are in the way of application IOs
- Multiple levels of parallelism outside the control of the application
- How to explore the interplay of SSD internals and algorithms?
- How can we experiment with a cross-layer design?

Our contribution: EagleTree
- Simulates
  - SSD (HW & SW),
  - OS & Application
  - Experiments with cross layer design
- Experimental suite for
  - calibration,
  - repeatability,
  - graph generation

Moreover...

SSDs are fast and energy efficient!

However...

Moreover...

Demonstration outline
- Importance of scheduling
- Reads are problematic!
- Opening the interface

clyde.itu.dk