Noise resilience or mitigation for HPC measurements & noise generation

Bernd Mohr

b.mohr@fz-juelich.de

Feedback to Presented Juelich Approach

- Further ways to “model” effort
  - Combine basic block counts with static binary analysis info (e.g. from MAQAO)
- Micro benchmarks to detect and classify noise sources would be useful
  - Was task of ExtraNoise work packages on noise characterization and noise sensitivity analysis of applications
- ML techniques to classify noise influences
- Detecting noise in applications by tracking progress over time (figure of merit)

Noise Generation

- Adding restrictions on resources (via HW settings)
  - Frequency settings
  - Power capping
  - Thermal capping
  - Cache size (via RDT) available on Intel + AMD
  - Traffic shapping

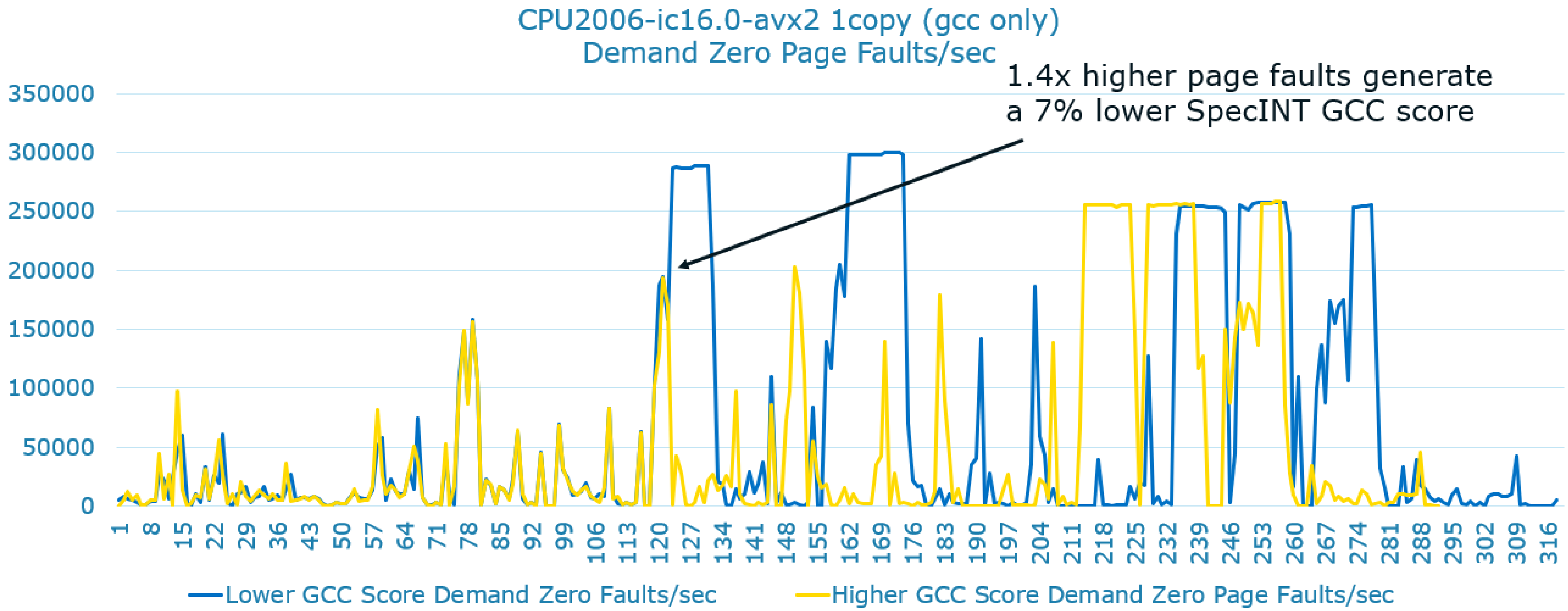
- Check out https://github.com/llnl/Gremlins
- Check out HPAS https://dl.acm.org/doi/pdf/10.1145/3337821.3337907
- Check-out fault injection frameworks for network and memory

(Surprising?) noise sources

- (Data presented by William + Martin)
  - Higher changes of frequency throttling when all cores in the nodes are used
  - Different frequencies between cores in same socket
- Weird SMT effects
- Garbage-collector
- Page-faults
- HW Telemetry services

Page faults example from Michael Chynoweth

10 Runs Showing Demand Zero Page Faults Generating Variance



1.4x higher page faults generate a 7% lower SpecINT GCC score

Average demand zero page faults/sec

gcc lower score → 65445 page faults

gcc higher score→ 47690 page faults

← Synchronous vs. async limits