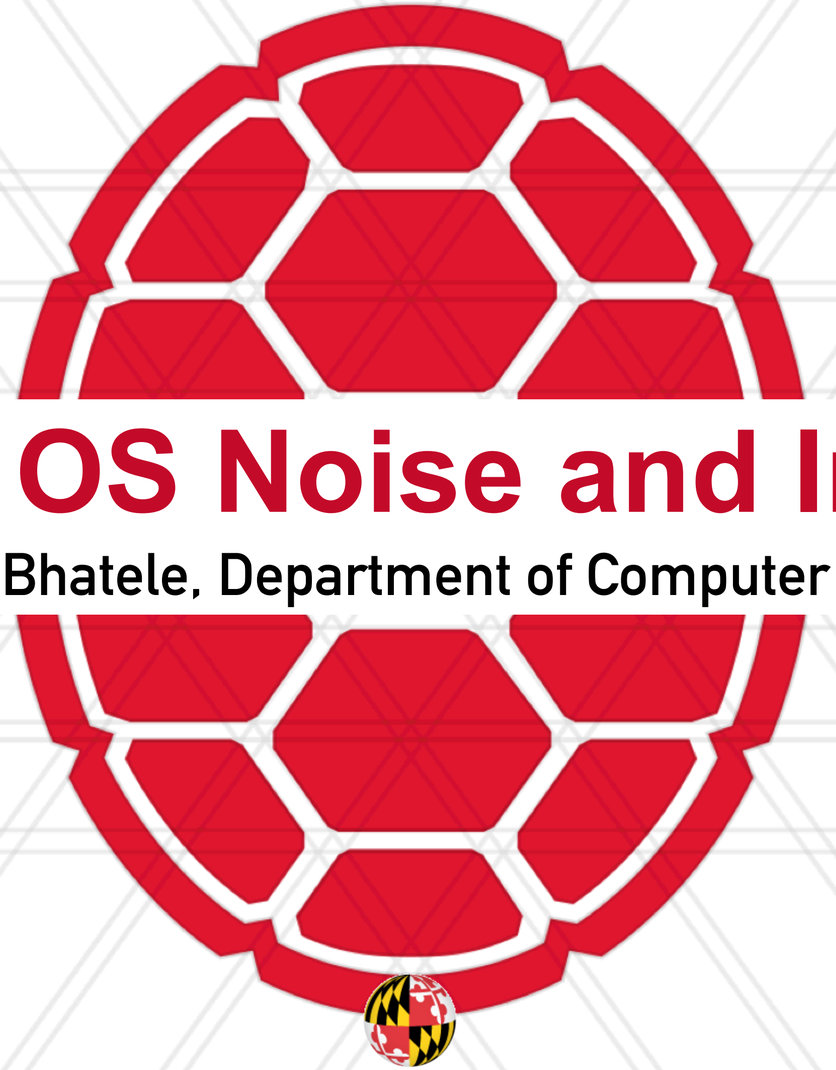
High Performance Computing Systems (CMSC714)



Lecture 15: OS Noise and Interference

Abhinav Bhatele, Department of Computer Science

UNIVERSITY OF MARYLAND

Summary of last lecture

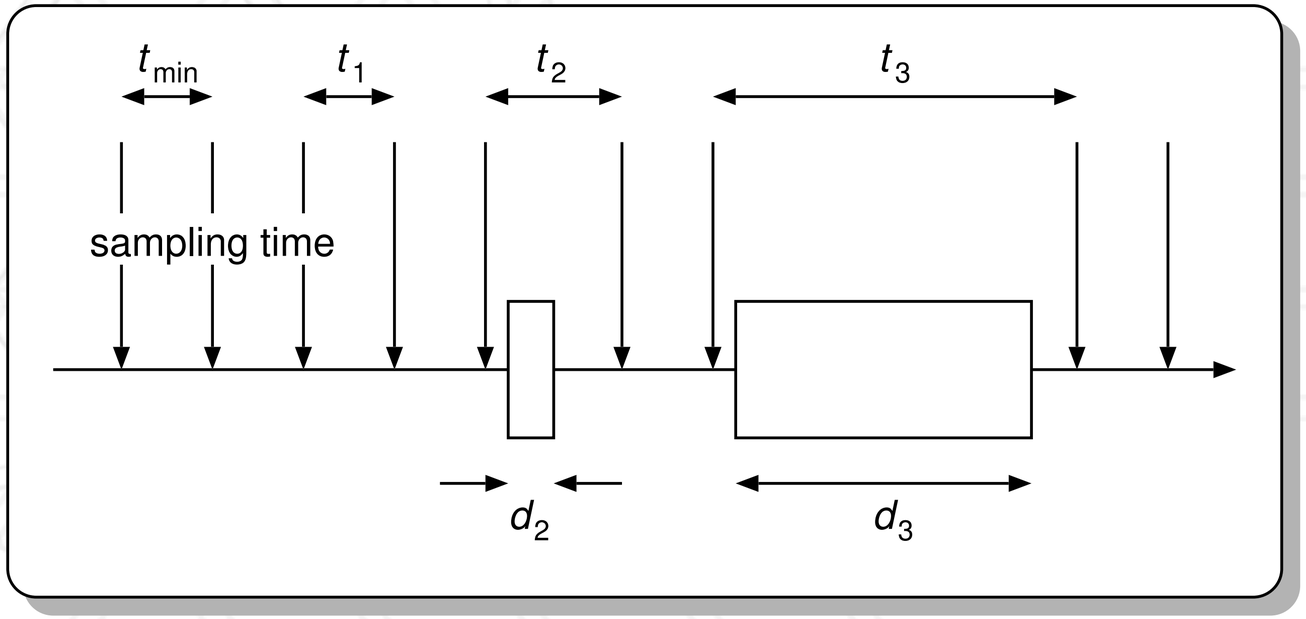
- Goal of auto-tuning: performance portability
- Selecting code variants, applications/system/parameters
- Model free vs. model-based
- Modeling: analytical, empirical, machine learning

Operating System

- Node on an HPC cluster may have:
  - A "full" linux kernel, or
  - A light-weight kernel
- Decides what services/daemons run
- Impacts performance predictability

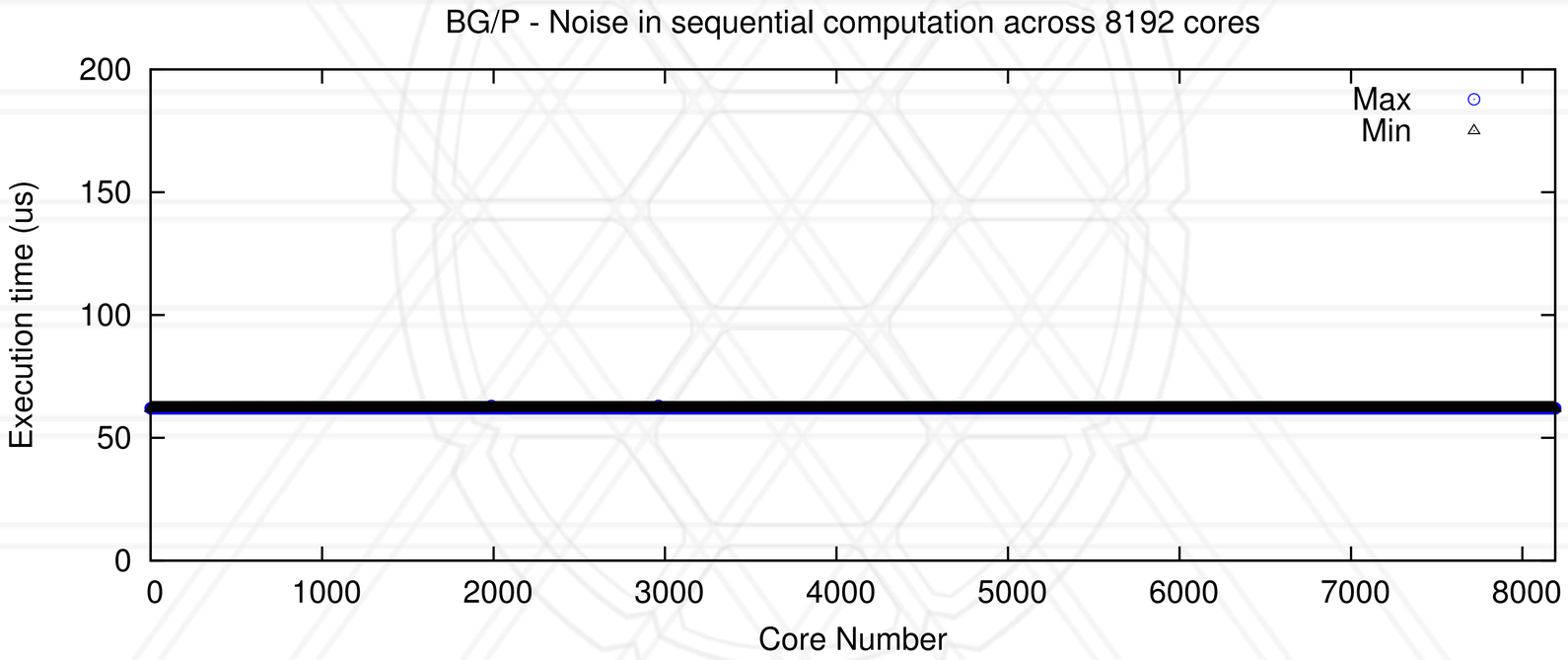
Operating System (OS) Noise

- Also called "jitter"
- Impacts computation due to interrupts by OS



Measuring OS Noise

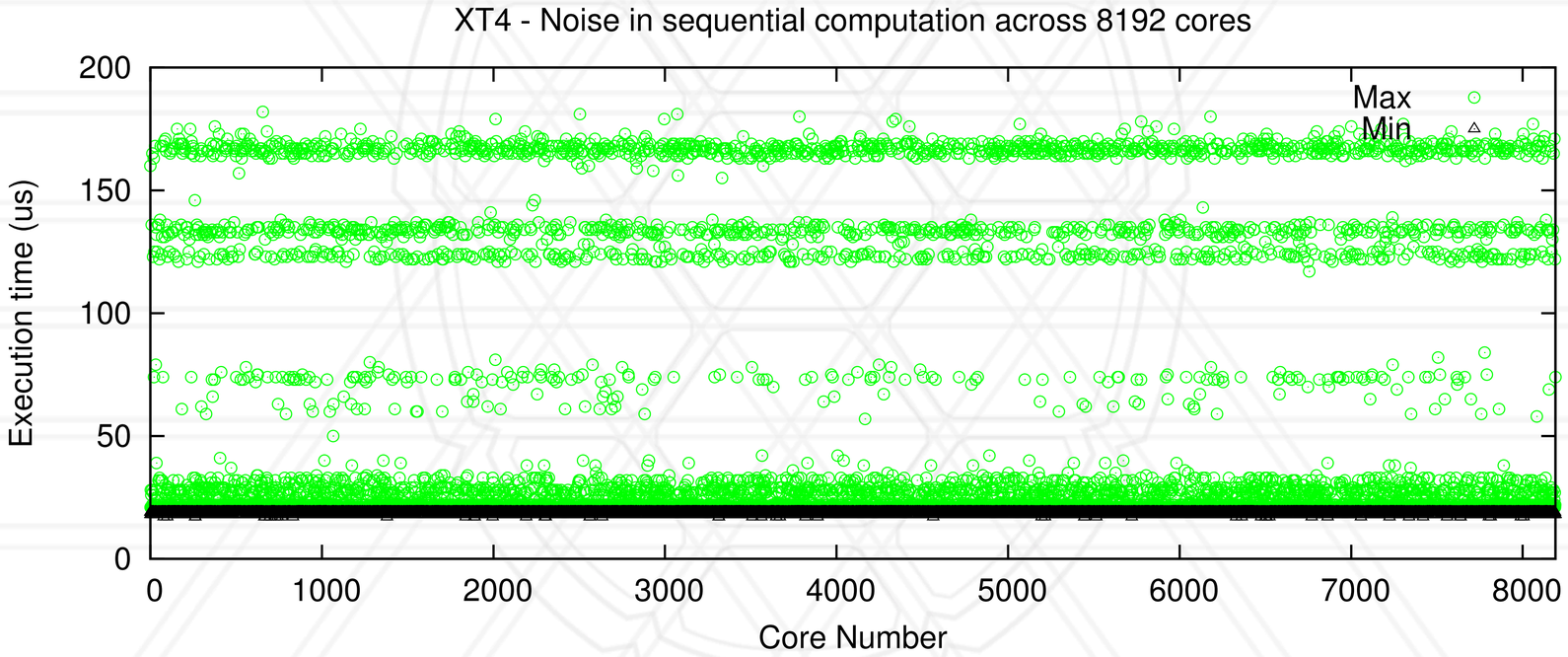
- Fixed Work Quanta (FTW) and Fixed Time Quanta (FTQ)



Benchmarks: https://asc.llnl.gov/sequoia/benchmarks/FTQ_summary_v1.1.pdf

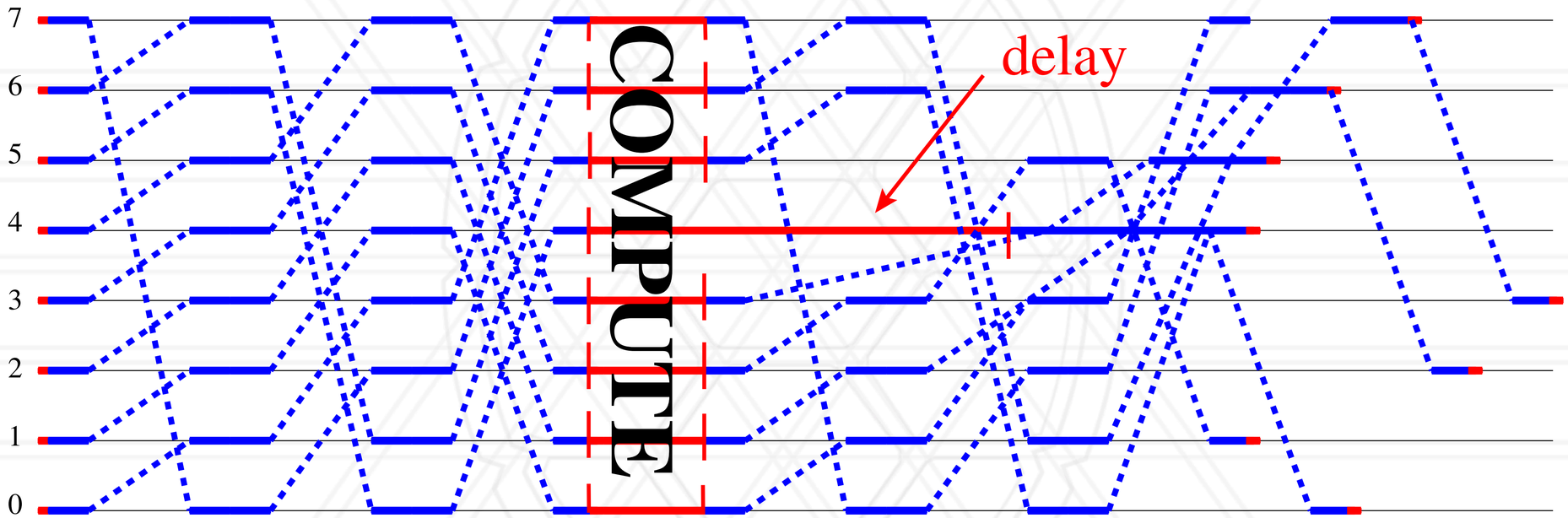
Measuring OS Noise

- Fixed Work Quanta (FTW) and Fixed Time Quanta (FTQ)



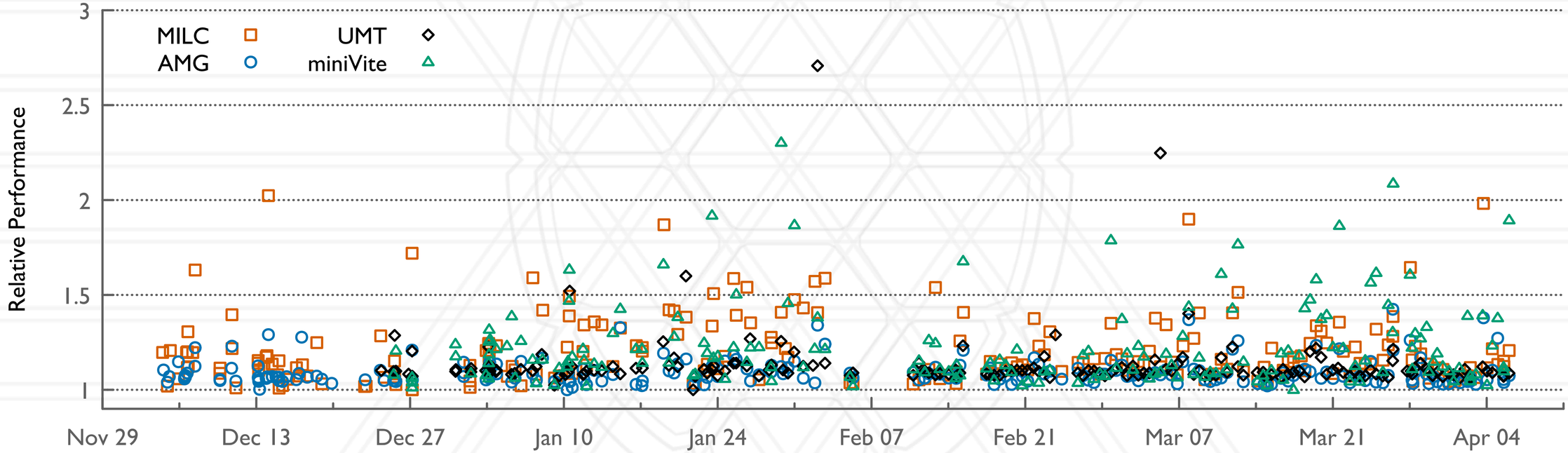
Benchmarks: https://asc.llnl.gov/sequoia/benchmarks/FTQ_summary_v1.1.pdf

Impact on communication



Hoefler et al.: https://htor.inf.ethz.ch/publications/img/hoefler-noise-sim.pdf

Impact on application codes



Leads to several problems ...

- Individual jobs run slower:
  - More time to complete science simulations
  - Increased wait time in job queues
  - Inefficient use of machine time allocation/core-hours
- Overall lower throughput
- Increase energy usage/costs

Also affects software development

- Debugging performance issues
- Quantifying the effect of various software changes on performance
  - code changes
  - compiler/software stack changes
- Requesting time for a batch job
- Writing allocation proposals

Questions

The Case of the Missing Supercomputer Performance

- Why does using 1, 2, 3 processes per node work as expected with the interference of system noise?
- How can we coschedule system noise in practice?
- What is the meaning of quadrics network?
- I am confused with the definition of computational granularity. Even if there is no message exchange, I/O, or memory access, I think context switches still happen and the CPU time can be handed from the application to system processes within a "computation phase" (p. 7). So, are granularities such as 1ms referring to the running time on a hypothetical noiseless machine and never precise on a real system? Why don't we measure the "actual" granularities?
- (p. 13, Sec. 6) Why "with a coarse-grained application the fine-grained noise becomes coscheduled"? It seems that coscheduling needs a special kernel module (Sec. 3.3) but no alteration on the system is done here. Does this happen automatically because of the length of the noise and the length of the computations?
- Back in the "Blue Gene/Q" paper, it is mentioned that there is one processor on the chip dedicated to OS services. Are that kind of systems immune to the types of noise discussed in this paper?
- The approach presented in this paper is highly systematic. Given a set of microbenchmarks and known types of noise, is it possible to make the identification of the potential causes of suboptimal performance automatic, like in the case of auto-tuning?

Questions

There Goes the Neighborhood

- The paper shows that the contention from other jobs is the main factor leading to the variability of performances, but is there a way to build a model that can quantify how much each candidate factor affects the messaging rate?

- The paper sets configurations in a way that similarity in the message passing characteristics of these three systems is maximized. How is it achieved?

- Sec. 5.2 and Sec. 5.3 investigate allocation shape (continuity) and contention from other jobs respectively. However, I think there is some extent of correlation between these two factors: jobs with lower continuity are in general more likely to suffer from contention because they usually have to use more links that are shared with other jobs. Therefore, how do we decouple the two factors and conclude that allocation shape is not a major one?

- Is there any node allocation policy that, if given an estimated communication load in addition to the expected running time of a job, can utilize this kind of information to alleviate the "conflicting router" problem and make a better allocation?

Questions?

UNIVERSITY OF
MARYLAND

Abhinav Bhatele

5218 Brendan Iribe Center (IRB) / College Park, MD 20742

phone: 301.405.4507 / e-mail: bhatele@cs.umd.edu