Operating Systems as Parallel Pipelines

Anton Burtsev John Regehr
University of Utah, School of Computing

Abstract

Mainstream operating systems are starting to face the need for support of multi-core platforms. We argue that pipeline parallelism is one of the practical ways to utilize performance offered by multiple cores with minimal changes to the existing software stack. Similar to the organization of hardware processors, execution of a system can be structured as a set of pipeline stages running in parallel on separate cores. Explicit data path defined by the pipeline provides advantages of simple isolated stage-to-stage synchronization, controlled data sharing and possibility to develop communication primitives managing memory as a first-class communication medium.

Experimenting with simplest pipelines, we realized that a throughput-optimized architecture of traditional kernels inherently allows support for pipeline extensions. At the same time, we found no means to reason about performance of a pipeline. Traditional CPU utilization metrics and sample based profiling provide no means to analyze, debug, and tune performance of any form of execution spread across multiple cores. Spatial separation of stages magnifies performance impact of cross-stage dependencies. A stall of a single pipeline stage blocks the entire pipeline.

This work aims to analyse potential performance benefits of introducing parallel pipelines to a traditional operating system. We determine core components of this architecture and create a set of rules to reason about performance and optimal scheduling of pipelines.

Students: Burtsev. Corresponding author: aburtsev@cs.utah.edu.