Abstract

- Core counts are increasing, making parallel programming an increasingly powerful paradigm.
- Decomposition of computations into very fine-grained tasks is necessary to effectively utilize many-core systems.
- We present XQueue, a novel design for a queuing system, analyze its efficiency, suggest improvements, and compare performance to existing designs, all within the context of XTask, a custom, task-based runtime for shared memory systems.

Motivation

- Swift/T is an implicitly parallel programming language used to implement scientific dataflow programs.
- Swift/T uses MPI for internode and intranode communication.
- Today's processors can run billions of instructions per second, however we are limited to 100K tasks per second.
- Traditional concurrent data structures and synchronization mechanisms do not scale to hundreds of cores.

Can high-level programming be applied to modern parallel architectures with strong scaling workloads?