Failure-Aware Rescheduling for Large-Scale Clusters

Zhiling Lan, Yawei Li, Prashasta Gujrati,
Felix Laurens, Rohit Agrawal, and Aarti Agarwal

Over the past decades, the insatiable demand for more computational power in science and engineering has driven the development of ever-growing supercomputers. High performance computing (HPC) systems with hundreds to thousands of processors, ranging from tightly coupled proprietary clusters to loosely coupled commodity-based clusters, are being designed and deployed. For systems of this scale, reliability is becoming a major concern as the system-wide mean-time-between-failure (MTBF) decreases exponentially with the increasing count of system components. According to a recent study, the MTBF for today's 10-20 Tflops machines is only on the order of 10-40 hours. Hence, it is crucial to provide a system-wide support to manage the unavoidable reality of failures on large-scale clusters.

We propose to enhance the system resilience to failures by developing *Failure-Aware ReScheduling (FARS).* In particular, we investigate the feasibility of utilizing failure prediction to dynamically adjust the placement of *active jobs* in response to pre-failure conditions. Here, *active jobs* denote those jobs that are already scheduled and running on the system, while *inactive jobs* denote those that are still hold in the job queues (i.e. the focus of failure-aware job scheduling). With failure-aware rescheduling, a key question is, "which process(es) should be migrated in case of foreseeable failures?" Prediction may vary in its accuracy and different failures generally associate with different downtimes, hence the movement of different process(es) can result in different performance gain in terms of system-level utilization rate. Considering these factors, we propose a heuristic rescheduling policy that can be used with regular batch schedulers to make them failure-aware. An event-based simulation is developed to emulate the proposed rescheduling policy with the widely-used FIFO/EASY Backfilling as well as other batch schedulers. The simulator requires the following inputs: a job log, a failure log and a predictor.