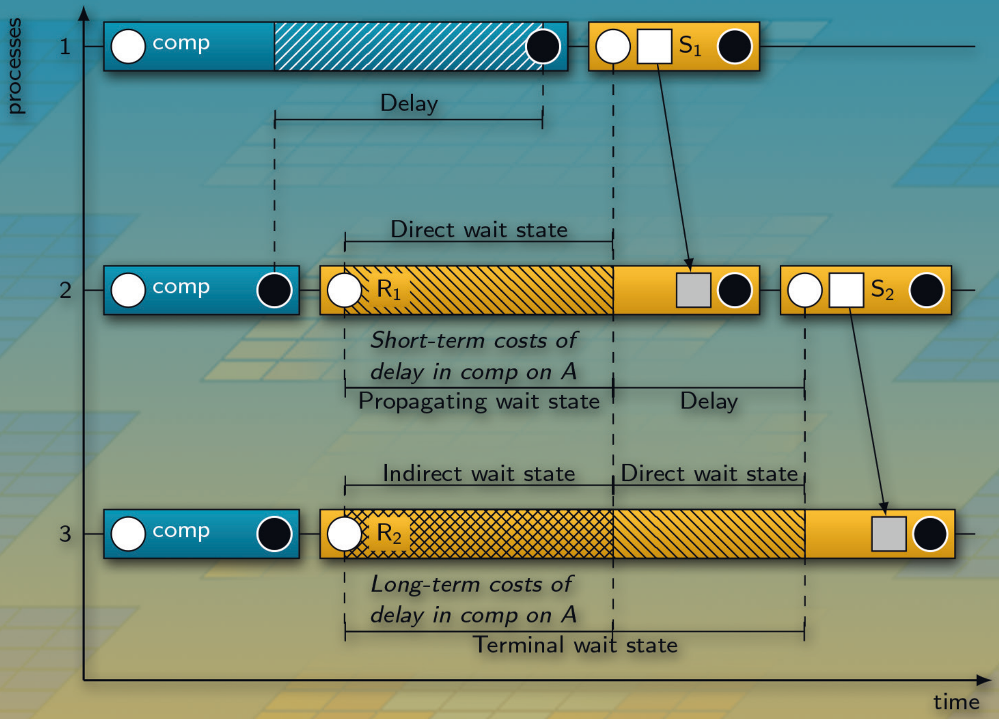

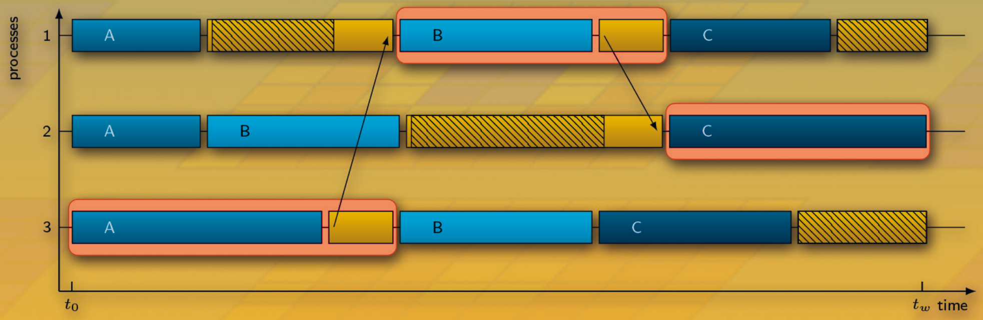

Characterizing Load and Communication Imbalance in Parallel Applications

David Böhme

JÜLICH FORSCHUNGSZENTRUM

Forschungszentrum Jülich GmbH
Institute for Advanced Simulation (IAS)
Jülich Supercomputing Centre (JSC)

Characterizing Load and Communication Imbalance in Parallel Applications

David Böhme

Schriften des Forschungszentrums Jülich

IAS Series Volume 23

ISSN 1868-8489 ISBN 978-3-89336-940-9

Contents

The amount of parallelism in modern supercomputers currently grows from generation to generation. Further application performance improvements therefore depend on software-managed parallelism: the software must organize data exchange between processing elements efficiently and optimally distribute the workload between them. Performance analysis tools help developers of parallel applications to evaluate and optimize the parallel efficiency of their programs. This dissertation presents two novel methods to automatically detect imbalance-related performance problems in MPI programs and intuitively guide the performance analyst to inefficiencies whose optimization promise the highest benefit.

The first method, the delay analysis, identifies the root causes of wait states. A delay occurs when a program activity needs more time on one process than on another, which leads to the formation of wait states at a subsequent synchronization point. Wait states are the primary symptom of load imbalance in parallel programs. While wait states themselves are easy to detect, the potentially large temporal and spatial distance between wait states and the delays causing them complicates the identification of wait-state root causes.

The delay analysis closes this gap, accounting for both short-term and long-term effects. The second method is based on the detection of the critical path, which determines the effect of imbalance on program runtime. The critical path is the longest execution path in a parallel program without wait states: optimizing an activity on the critical path will reduce the program's runtime. Comparing the duration of activities on the critical path with their duration on each process yields a set of novel, compact performance indicators. These indicators allow users to evaluate load balance, identify performance bottlenecks, and determine the performance impact of load imbalance at first glance by providing an intuitive understanding of complex performance phenomena.

Both analysis methods leverage the scalable event-trace analysis technique employed by the Scalasca toolset: by replaying event traces in parallel, the bottleneck search algorithms can harness the distributed memory and computational resources of the target system for the analysis, allowing them to process even large-scale program runs.

IAS Series
Volume 23
ISBN 978-3-89336-940-9

JÜLICH
FORSCHUNGSZENTRUM