

# Mapping Unstructured Parallelism to Series-Parallel DAGs

Yan Pan<sup>a</sup>, Wen-Jing Hsu<sup>a,b</sup>

<sup>a</sup>Singapore-MIT Alliance, E4-04-10, 4 Engineering Drive 3, Singapore 117576

<sup>b</sup>Nanyang Technological University, Singapore

**Abstract**—Many parallel programming languages allow programmers to describe parallelism by using constructs such as fork/join. When executed, such programs can be modeled as directed graphs, with nodes representing a computation and edges representing the sequence and dependency. However, because it does not coerce regularity in the computation, the general model is not amenable to efficient execution of the resulting program. Therefore, a more restrictive model called Series-Parallel DAG (Directed Acyclic Graph) has been proposed and adopted by several major parallel languages. As reported by the Cilk developers, many parallel computations can be easily expressed in the series-parallel model, and there are provably efficient scheduling algorithms for the SP DAGs. Nevertheless, it remains open how much *inherent* parallelism will be lost when conforming to the model, because expressing a computation in the series-parallel model may also induce performance losses. We will show that *any* general DAG can be converted into an SP DAG without violating the original precedence relations; moreover, the conversion can be carried out in essentially linear time and space, and the resulting DAG exhibits little loss in the parallelism. Since the resulting SP DAG can then be executed with high efficiency, it implies that the languages based on SP DAGs are not as restrictive as they were thought to be.

[Full Text Not Available]