Parallel Topological Sorting

Design of High Performance Computing, Fall 2015

Kevin Wallimann Johannes Baum Matthias Untergassmair

ETH Zürich

October 31, 2015

Problem Description

Input	Directed acyclic graph (DAG) with N nodes
Output	Topological Sortings of DAG

"Efficient" Parallel and Distributed Topological Sort Algorithms

- Runtime: $\mathcal{O}(\log^2 N)$
- Reduces to matrix-matrix multiplication problem

Problem:

$\mathcal{O}(N^3)$ execution units required

Serial Code

```
std::list<Node*> currentnodes;
. . .
Node* parent, child;
unsigned childcount = currentvalue = 0;
. . .
while(!currentnodes.empty()) {
  parent = currentnodes.front();
  currentnodes.pop_front();
  currentvalue = parent->getValue();
  ++currentvalue;
  childcount = parent->getChildCount();
  for(unsigned i=0; i<childcount; ++i) {
    child = parent->getChild(i);
    currentnodes.push_back(child);
    child->setValue(currentvalue);
  }
}
```

Parallelization Ideas

-

¡++¿

Challenges

- Task/Load balancing

Questions

- Which platform?
- Memory distribution?