

# 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

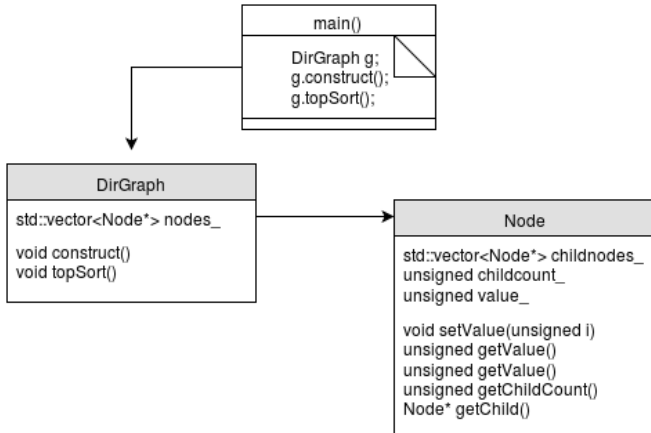
# "EFFICIENT" PARALLEL AND DISTRIBUTED TOPOLOGICAL SORT ALGORITHMS

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

PROBLEM:

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

# UML DIAGRAM



# SERIAL CODE

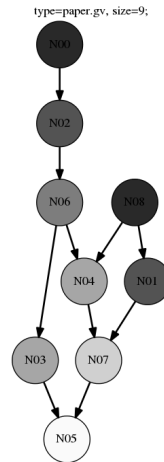
```
std::list<Node*> currentnodes;

Node* parent;
Node* child;
unsigned childcount = 0;
unsigned currentvalue = 0;

while(!currentnodes.empty()) { // stop when queue is empty
    parent = currentnodes.front();
    currentnodes.pop_front(); // remove current node - already visited
    currentvalue = parent->getValue();
    ++currentvalue; // increase value for child nodes
    childcount = parent->getChildCount();
    for(unsigned i=0; i<childcount; ++i) {
        child = parent->getChild(i);
        currentnodes.push_back(child); // add child node at end of queue
        child->setValue(currentvalue); // set value of child node to parentvalue+1
    }
}
```

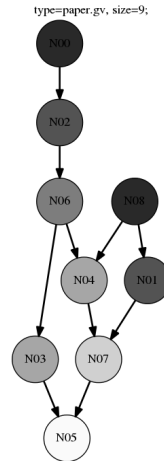
# PARALLELIZATION IDEAS

- Start in parallel at source nodes
- Spawn a new thread for every child node
- Synchronization needed when a node has multiple parents
- Performance dependent on topology of graph. Worst case  $\mathcal{O}(n)$



# SYNCHRONIZATION IDEAS

- Each thread enumerates the nodes it visits.
- If a node already has a number, take the higher number
- Problem: Multiple threads might “follow” each other.
- Idea: Only last thread arriving at a node may continue.



# HARDWARE / TOOLS

- Shared memory parallelization
- OpenMP or C++11 threads.
- On Euler: 12-core Intel Xeon E5
- Intel Xeon Phi



# CHALLENGES

- Task queue
- Find a way to cope with chain-like graphs.
- Optional goal: Find all possible topological sortings.

# QUESTIONS

- Up to how many execution units should we parallelize at least?
- From your experience, what are the main time-consuming, maybe unexpected obstacles when running on Intel Xeon Phi?