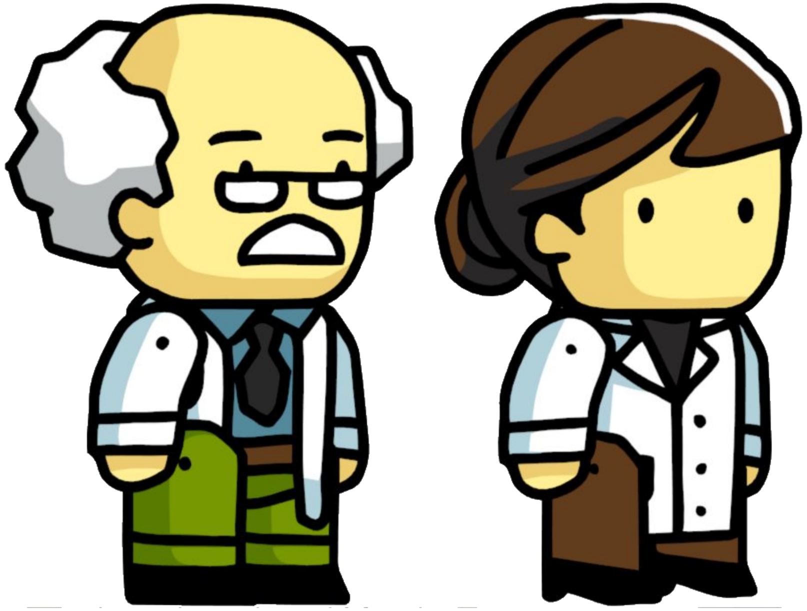
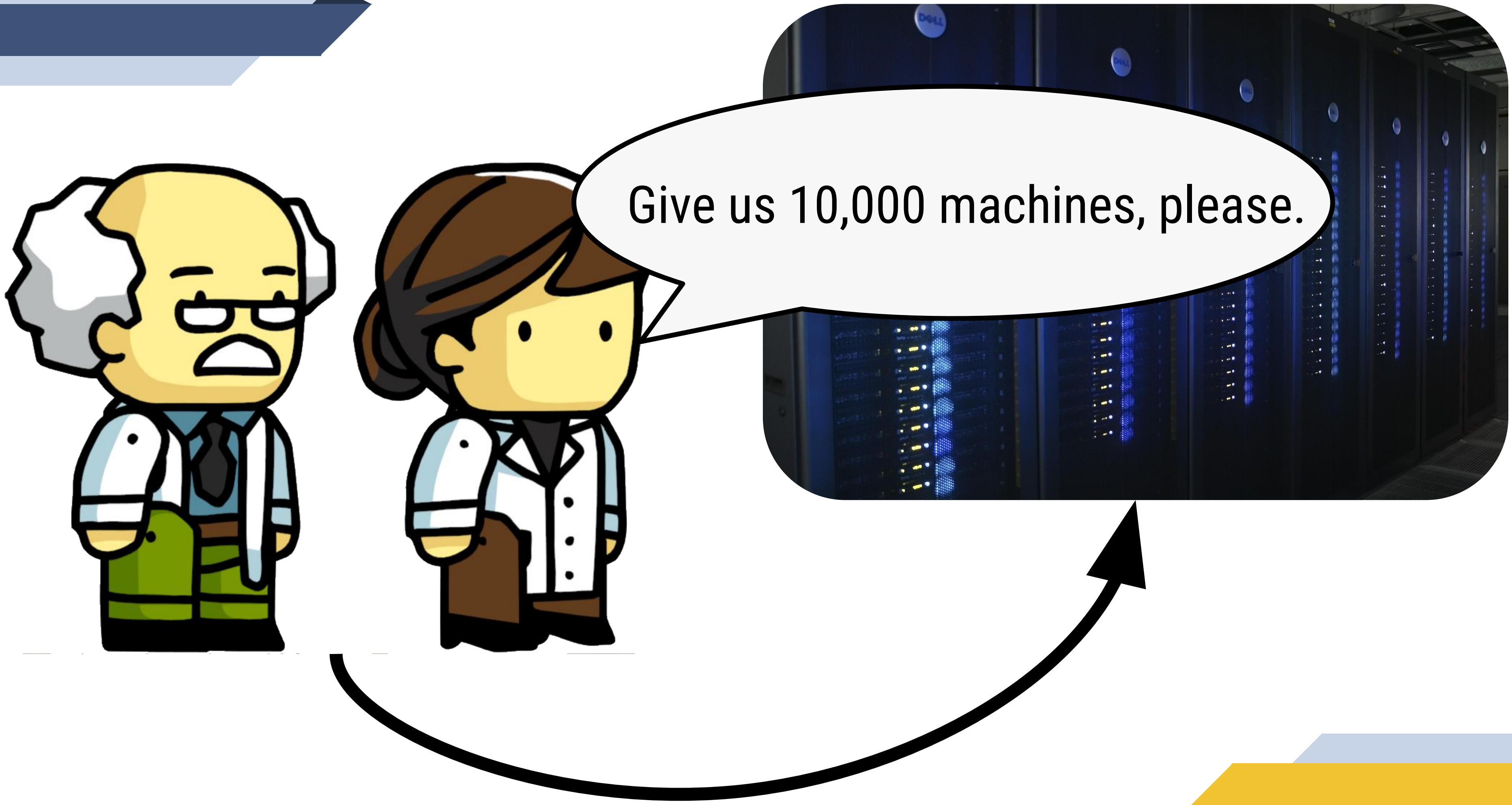
UNIVERSITY OF NOTRE DAME

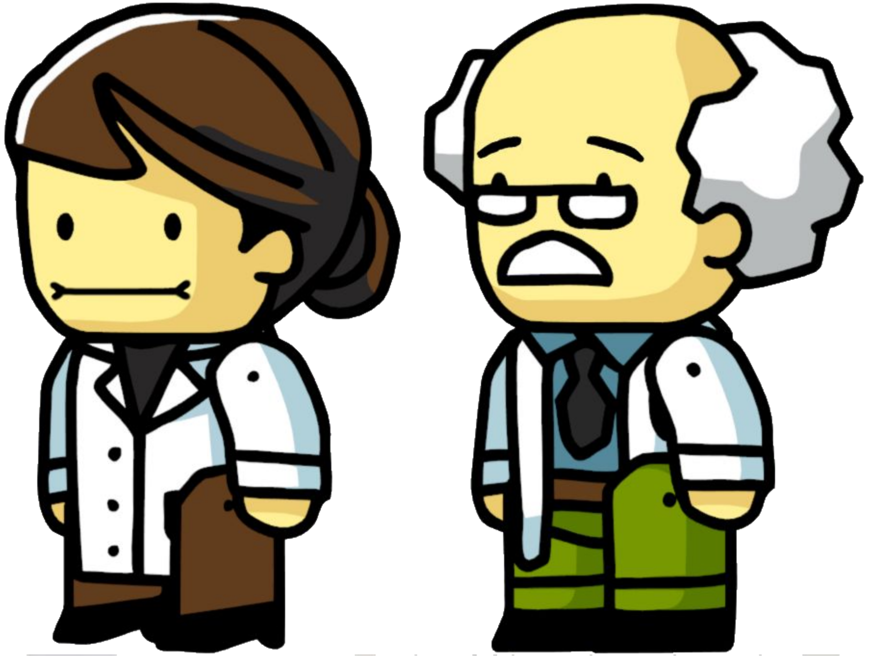
Depth-First Search and Its Use Case in Distributed Systems Debugging

Nate Kremer-Herman

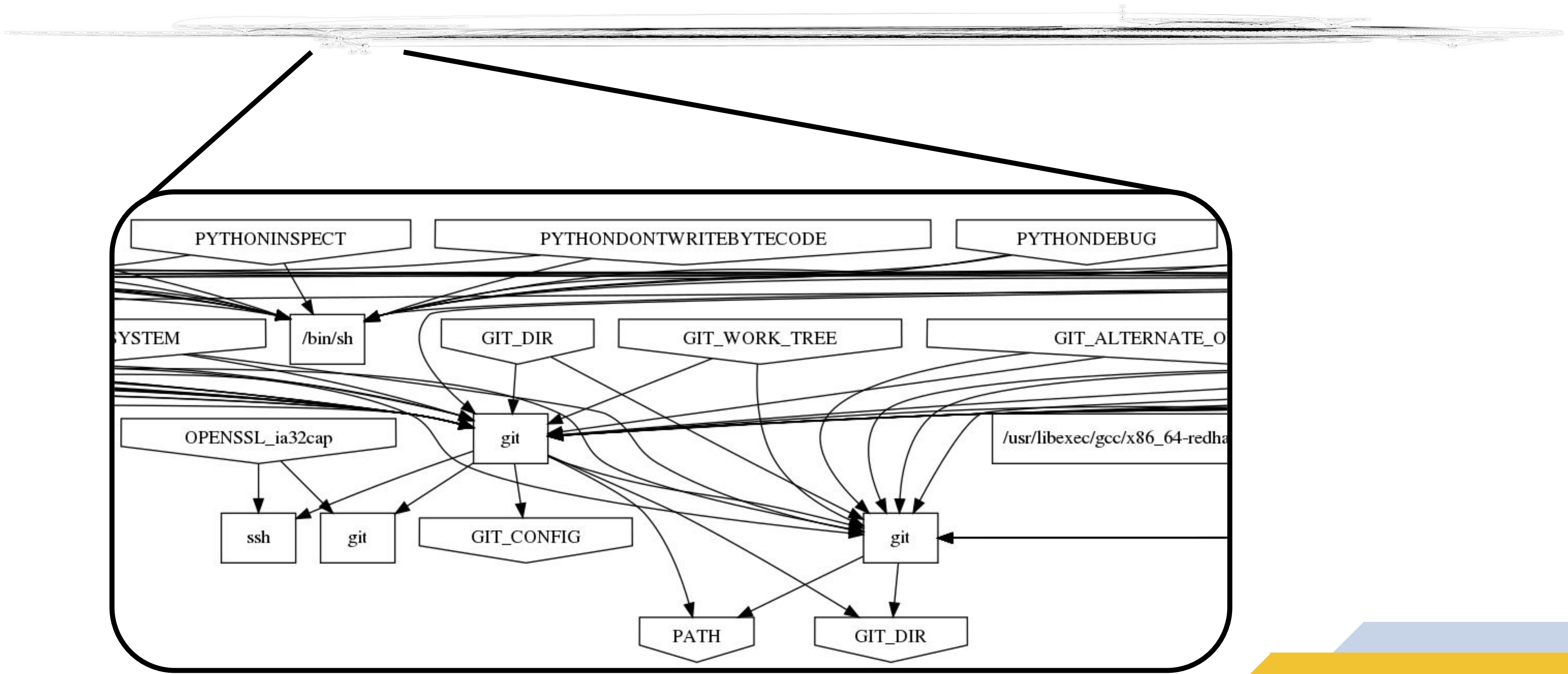


DELL


Give us 10,000 machines, please.

DELL


Let’s look at an example.


PYTHONINSPECT
PYTHONDONTWRITEBYTECODE
PYTHONDEBUG
YSTEM
/bin/sh
GIT_DIR
GIT_WORK_TREE
GIT_ALTERNATE_O
OPENSSL_ia32cap
git
/usr/libexec/gcc/x86_64-redha
ssh
git
GIT_CONFIG
git
PATH
GIT_DIR

Debugging distributed systems

Hard to determine if an early task caused one later on to fail.
Tasks run on different machines, OSs, and clusters.

How do we track *how/when* task n set task n+100 up to fail?

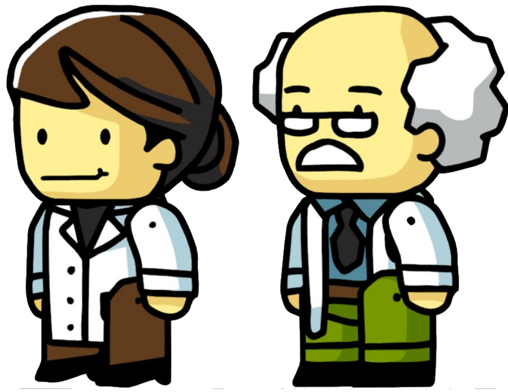
How do we comb through a *lineage* of processes and configurations? BFS? DFS? Heuristic traversal algorithms?

What is the end-user application?

Given a very large and complex task lineage trace.
In each node are env. variables set/consumed for that task.

Use DFS to explore this tree and track the env. variables.
Tool assists user in correcting configurations for bad tasks.

How can we parallelize this tree exploration *efficiently*?
At what scale does it *become* efficient?



Depth-first search kernel

Given a large graph.

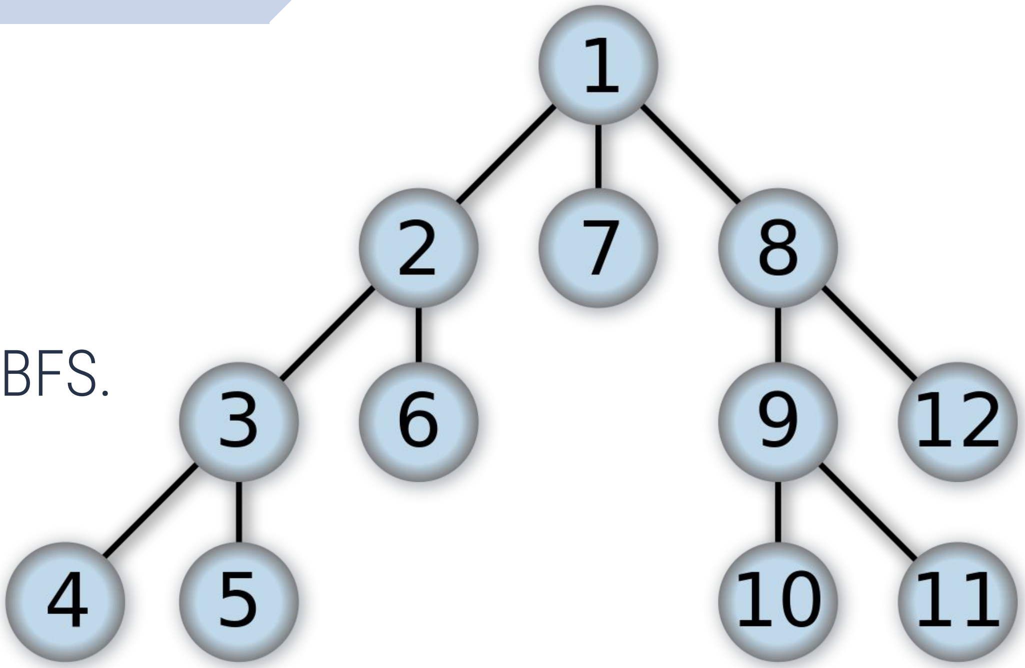
Start at a root node.

Find all reachable vertices.

Measured in TEPS, just like BFS.

Worst case performance:

O(|V| + |E|)



Notional sequential algorithm

1. Select some arbitrary node (root node if tree).
2. Explore each branch as far as possible, keeping track of visited nodes.
3. When the current branch is exhausted, backtrack to last unexplored branch.

Iterative pseudocode

```
procedure DFS-iterative($G$,$v$):
    let $S$ be a stack
    $S$.push($v$)
    while $S$ is not empty
        $v$ = $S$.pop()
        if $v$ is not labeled as discovered:
            label $v$ as discovered
            for all edges from $v$ to $w$ in $G$.adjacentEdges($v$) do
                $S$.push($w$)
```

Recursive pseudocode

```
procedure DFS($G$,$v$):
    label $v$ as discovered
    for all edges from $v$ to $w$ in $G$.adjacentEdges($v$) do
        if vertex $w$ is not labeled as discovered then
            recursively call DFS($G$,$w$)
```

Questions?

UNIVERSITY OF NOTRE DAME

Please clap.