Dependencies

1/17/16
But first, the homework (Java threads)

class Task implements Runnable {
    public Task(...) { ... }
    public void run() { ... }
}

Thread t1 = new Thread(new Task(...));
Thread t2 = new Thread(new Task(...));
t1.start();  t2.start();
t1.join();   t2.join();
Instruction dependencies

• From instruction $I_1$ to $I_2$ there is a
  – Flow dependence (aka true dependence) if $I_1$ computes a value used by $I_2$
  – Anti-dependence if $I_1$ takes an operand from a register into which $I_2$ later stores a result
  – Output dependence if $I_1$ and $I_2$ use the same register to store their results
What dependence do the instructions $I_1: R_1 \leftarrow R_2 + R_3$ and $I_2: R_3 \leftarrow R_4 + R_5$ have?

A. A flow dependence  
B. An anti-dependence  
C. An output dependence  
D. More than one of the above  
E. None; the instructions are independent
What dependence do the instructions $I_1: R_1 <- R_2 + R_3$ and $I_2: R_3 <- R_4 + R_5$ have?

A. A flow dependence
B. An anti-dependence (register $R_3$)
C. An output dependence
D. More than one of the above
E. None; the instructions are independent
What dependence do the instructions $I_1: R_1 <- R_2 + R_3$ and $I_2: R_1 <- R_1 + R_4$ have?

A. A flow dependence
B. An anti-dependence
C. An output dependence
D. More than one of the above
E. None; the instructions are independent
What dependence do the instructions \( I_1: R_1 \leftarrow R_2 + R_3 \) and \( I_2: R_1 \leftarrow R_1 + R_4 \) have?

A. A flow dependence (on \( R_1 \))
B. An anti-dependence
C. An output dependence (on \( R_1 \))
D. More than one of the above
E. None; the instructions are independent
Task graphs

• Representation of higher-level dependencies
  – nodes = strands (serial parts w/o parallel constructs)
  – edges = showing which strands must complete first
Draw a dependence graph for heat diffusion on domain  \( \begin{array}{ccc}
A & B & C \\
D & E & F 
\end{array} \)  where \( X_1, X_2, \ldots \) are updates for \( X \)

1) For the serial algorithm
Draw a dependence graph for heat diffusion on domain \[
\begin{array}{ccc}
A & B & C \\
D & E & F \\
\end{array}
\] where \(X_1, X_2, \ldots\) are updates for \(X\)

2) For the SimpleExecutor version
Draw a dependence graph for heat diffusion on domain where $X_1, X_2, \ldots$ are updates for $X$

3) For the version using Thread and CyclicBarrier
Draw a dependence graph for heat diffusion on domain

\[
\begin{array}{ccc}
A & B & C \\
D & E & F \\
\end{array}
\]

where \(X_1, X_2, \ldots\) are updates for \(X\)

4) For the inherent problem (what \textit{could} be done in parallel?)
Draw a dependence graph for heat diffusion on domain where $X_1, X_2, \ldots$ are updates for $X$

5) For this proposed implementation:

```java
//in runSim:
...
for(int i=0; i<timeSteps; i++) {
    TSTask t = new TSTask(); //update data for //entire grid
    e.submit(t);
}
```