Review for the Midterm Exam

1. Three Questions of the Computational Science Prelim
   - scaled speedup
   - network topologies
   - work stealing
2. The in-class Spring 2012 Midterm Exam
   - pleasingly parallel computations
   - pipelining
   - synchronized computations
   - parallel linear algebra

MCS 572 Lecture 25
Introduction to Supercomputing
Jan Verschelde, 19 October 2016

Review for the Midterm Exam

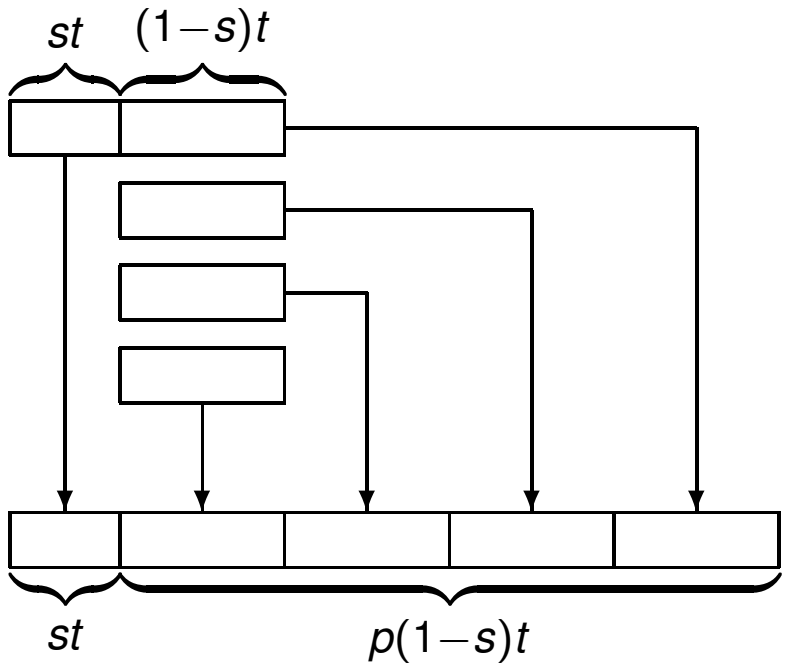
1. Three Questions of the Computational Science Prelim
   - scaled speedup
   - network topologies
   - work stealing
2. The in-class Spring 2012 Midterm Exam
   - pleasingly parallel computations
   - pipelining
   - synchronized computations
   - parallel linear algebra

scaled speedup

Benchmarking of a program running on a 12-processor machine shows that 5% of the operations are done sequentially,
i.e.: that 5% of the time only one single processor is working while the rest is idle.
Compute the scaled speedup.

solution

$p = 4$



Scaled speedup $S_s(p) \leq \dfrac{st + p(1-s)t}{t} = s + p(1-s) = p + (1-p)s.$

Evaluate for $s = 0.05, p = 12$: $S_s(12) = 12 + (1-12)0.05 = 11.45.$

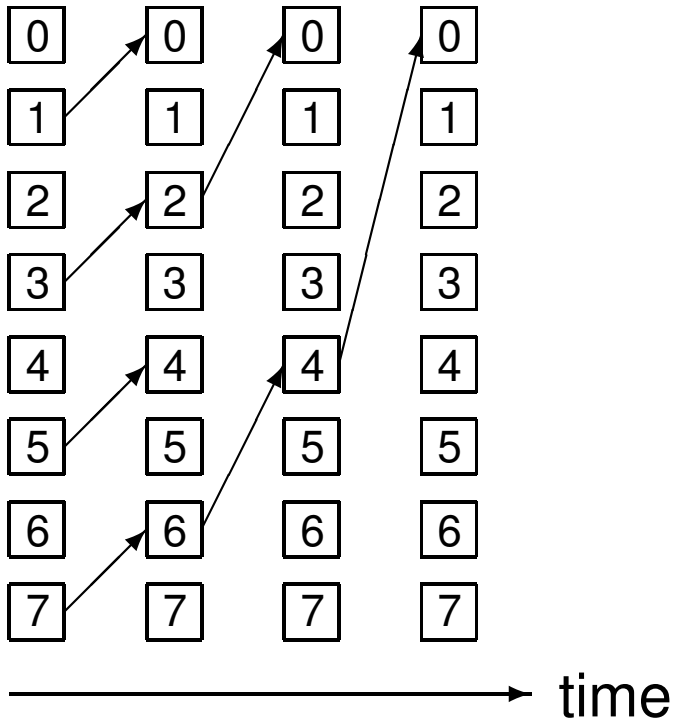
Review for the Midterm Exam

1. Three Questions of the Computational Science Prelim
   - scaled speedup
   - network topologies
   - work stealing
2. The in-class Spring 2012 Midterm Exam
   - pleasingly parallel computations
   - pipelining
   - synchronized computations
   - parallel linear algebra

network topologies

Show that a hypercube network topology has enough connections for a fan-in gathering of results.

solution for $8 = 2^3$ nodes



Three steps:

1. $001 \rightarrow 000; 011 \rightarrow 010; 101 \rightarrow 100; 111 \rightarrow 110$
2. $010 \rightarrow 000; 110 \rightarrow 100$
3. $100 \rightarrow 000$

proof by induction

- The base case: we verified for 1, 2, 4, and 8 nodes.
- Assume we have enough connections for 2^k hypercube.

  Need to show: have enough connections for 2^{k+1} hypercube:

  1. In the first k steps:
     - ★ node 0 gathers from nodes $1, 2, \ldots 2^k - 1$;
     - ★ node 2^k gathers from nodes $2^k + 1, 2^k + 2, \ldots, 2^{k+1} - 1$.
  2. In step $k + 1$: node 2^k can send to node 0,

     because only one bit in 2^k is different from 0.

Review for the Midterm Exam

1. Three Questions of the Computational Science Prelim
   - scaled speedup
   - network topologies
   - work stealing
2. The in-class Spring 2012 Midterm Exam
   - pleasingly parallel computations
   - pipelining
   - synchronized computations
   - parallel linear algebra

work stealing

Explain the concept of work stealing.
What is work stealing?
What is its purpose?
Describe an example of an environment
and an application that uses work stealing.

recall lecture 11 ...

In scheduling threads on processors,
we distinguish between work sharing and work stealing:

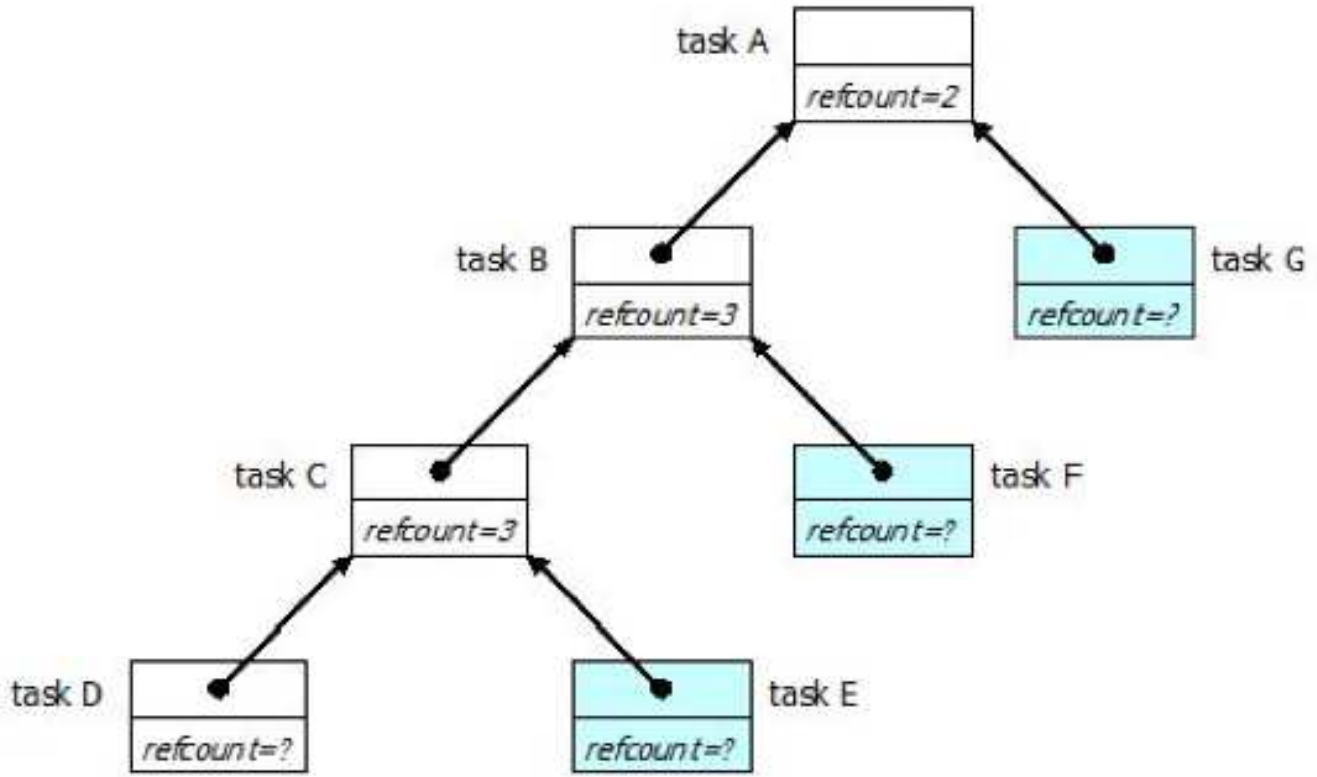
- In work sharing, the scheduler attempts to migrate threads to under-utilized processors in order to distribute the work.
- In work stealing, under-utilized processors attempt to steal threads from other processors.

The purpose of work stealing is thus to utilize all processors.

The Intel TBB task scheduler uses work stealing.

from the Intel TBB documentation

Snapshot of Task Graph for the Fibonacci Example



The figure above shows a snapshot of a task graph for the Fibonacci example where:

- Tasks A, B, and C spawned child tasks that they are waiting upon. Their *refcount* values are the number of children in flight plus one. The extra one is part of a convention for explicit waiting that is explained later in this section.
- Task D is running, but has not yet spawned any children, and so it has not had to set its reference count yet.
- Tasks E, F, and G have been spawned, but have not yet started executing.

Review for the Midterm Exam

1. Three Questions of the Computational Science Prelim
   - scaled speedup
   - network topologies
   - work stealing

2. The in-class Spring 2012 Midterm Exam
   - pleasingly parallel computations
   - pipelining
   - synchronized computations
   - parallel linear algebra

pleasingly parallel computations

Consider $\int_{-\infty}^{+\infty} e^{-x^2} dx$

as the area between the curve e^{-x^2} and the x-axis.

1. Write pseudo code for a parallel Monte Carlo method to approximate $\int_{-\infty}^{+\infty} e^{-x^2} dx$.

2. Explain the need for the dedicated software SPRNG to generate random numbers for distributed processing.
   Give at least two reasons for the need for SPRNG.

Review for the Midterm Exam

1. Three Questions of the Computational Science Prelim
   - scaled speedup
   - network topologies
   - work stealing

2. The in-class Spring 2012 Midterm Exam
   - pleasingly parallel computations
   - pipelining
   - synchronized computations
   - parallel linear algebra

pipelining

Consider the fast Fourier transform and the denoising of signals.

1. Describe the potential for a pipelined algorithm to remove noise from a sampled signal.
   Define the stages in the pipeline and draw a space-time diagram for an example signal.
2. Refer to the isoefficiency analysis we derived in class to describe the scalability of the pipelined denoising algorithm.

Review for the Midterm Exam

1. Three Questions of the Computational Science Prelim
   - scaled speedup
   - network topologies
   - work stealing

2. The in-class Spring 2012 Midterm Exam
   - pleasingly parallel computations
   - pipelining
   - synchronized computations
   - parallel linear algebra

synchronized computations

Consider the Poisson equation $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 1$, for $(x, y) \in [0, 1] \times [0, 1]$.

1. For a grid spanned by $(n+1)^2$ equidistant points $(x_i = i/n, y_j = j/n)$, $i = 0, 1, \ldots, n$ and $j = 0, 1, \ldots, n$, consider the linear system obtained after discretization to compute $u_{i,j} = u(x_i, y_j)$.
   What method would you recommend to solve this linear system on a parallel computer?
   Justify your answer.

2. Describe the scalability of your recommended parallel method for this problem.

Review for the Midterm Exam

1. Three Questions of the Computational Science Prelim
   - scaled speedup
   - network topologies
   - work stealing

2. The in-class Spring 2012 Midterm Exam
   - pleasingly parallel computations
   - pipelining
   - synchronized computations
   - parallel linear algebra

parallel linear algebra

Consider the LU factorization of a matrix A
on a shared memory parallel computer.

1. Explain why a tiled LU factorization as implemented in the PLASMA software package is more advantageous compared to a straightforward parallel implementation of the LU factorization with partial pivoting.

2. Discuss the numerical stability of the tiled LU factorization with partial pivoting in a tile.
   Hint: compare the case of a dense A with a sparse A.