

HW0: Exchange Problem

Course: Informal Parallel Programming Course for High School Students, Fall 2007
Title: Exchange Problem
Date Assigned: September 15, 2007
Date Due: September 25, 2007

1 Problem statement

Input: An array $A = A(0), \dots, A(n-1)$ and another array $B = B(0), \dots, B(n-1)$ of elements (any kind).
Task: The exchange problem is to swap all elements of A and B. For example if $A = (1, 5, 7, 3)$ and $B = (2, 6, 4, 8)$, after exchange $A = (2, 6, 4, 8)$ and $B = (1, 5, 7, 3)$.

Serial algorithm with $3n$ operations, $3n$ steps, 1 space (X):

```
for i=1 to n do
  X = A[i];
  A[i] = B[i];
  B[i] = X;
```

Parallel algorithm (1) with $3n$ operations, 3 steps, n spaces ($X[1..n]$):

```
for i=1 to n pardo
  X[i] = A[i];
  A[i] = B[i];
  B[i] = X[i];
```

Parallel algorithm (2) with $4n$ operations, 2 steps, $2n$ spaces ($X[1..n]$, $Y[1..n]$):

```
for i=1 to n pardo
  X[i] = A[i]; | B[i] = X[i];
  Y[i] = B[i]; | A[i] = Y[i];
```

2 Assignment

- Write XMTC programs realizing serial and both parallel algorithms.
- Discuss: Is the second parallel algorithm preferred to the first one?

Write XMTC programs realizing serial and both parallel algorithms. Is the second parallel algorithm preferred to the first one?

3 The program

You must name your XMTC files as follows: exchange.s.c (serial), exchange1.p.c (parallel 1), exchange2.p.c (parallel 2).

3.1 Setting up the environment

The header files and the binary files can be downloaded from `~swatson/xmtdata`. To get the data files, log in to your account in the class server and copy the `exchange.tgz` file from directory using the following commands:

```
$ cp ~swatson/xmtdata/exchange.tgz ~/
$ tar xzvf exchange.tgz
```

This will create the directory `exchange` with following folders: `data`, `src`, and `doc`. Data files are available in data directory.

3.2 Input format

You are given two arrays `A` and `B` that contains `n` integers.

<code>int n</code>	The size of the arrays
<code>int A[n]</code>	The array A
<code>int B[n]</code>	The array B

3.3 Data sets

Run all your programs (serial and parallel) using the following data files. You can directly include the header file into your XMTC code with `#include` or you can include the header file with the compile option `-include`. To run the compiled program you will need to specify the binary data with `-data-file` option.

Data Set	Header File	Binary file
<code>n = 16</code>	<code>data/16/exchange.h</code>	<code>data/16/exchange.32b</code>
<code>n = 3200</code>	<code>data/3200/exchange.h</code>	<code>data/3200/exchange.32b</code>
<code>n = 16000</code>	<code>data/16000/exchange.h</code>	<code>data/16000/exchange.32b</code>
<code>n = 64000</code>	<code>data/64000/exchange.h</code>	<code>data/64000/exchange.32b</code>

3.4 Output

Prepare and fill the following table: Create a text file named `table.txt` in `doc`. Remove any `printINT` and `printSTR` statements from your code while taking these measurements. `Printf` statements increase the clock and instruction count. Therefore the measurements with `printf` statements and may not reflect the actual time and work done. Is the second parallel algorithm preferred to the first one? After having filled the table, write your arguments.

Input size	<code>n = 16</code>	<code>n = 3200</code>	<code>n = 16000</code>	<code>n = 64000</code>
Serial Clock Cycles				
Parallel1 Clock Cycles				
Parallel2 Clock Cycles				

3.5 Submission

The use of the make utility for submission *make submit* is required. Make sure that you have the correct files at correct locations (*src* and *doc* directories) using the *make submitcheck* command. Run following commands to submit the assignment:

```
$ make submitcheck  
$ make submit
```

If you have any questions, please send an e-mail to Scott Watson, swatson@umd.edu