

James Hahn

CS1645

Homework #1

1. Below are the answers to A, B, and C:

A.

Machine Type	Number of Processors	Location
Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband	2,282,544	Oak Ridge National Laboratory, USA
Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway	10,649,600	National Supercomputing Center in Wuxi, China
Sierra - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband	1,572,480	DOE/NNSA/LLNL, United States
Tianhe-2A - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000	4,981,760	National Super Computer Center in Guangzhou, China
AI Bridging Cloud Infrastructure (ABCI) - PRIMERGY CX2550 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR	391,680	National Institute of Advanced Industrial Science and Technology, Japan

B.

Machine Type	Number of Processors	Location
Oakforest-PACS - PRIMERGY CX1640 M1, Intel Xeon Phi 7250 68C 1.4GHz, Intel Omni-Path	556,104	Joint Center for Advanced High Performance Computing , Japan
Stampede2 - PowerEdge C6320P/C6420, Intel Xeon Phi 7250 68C 1.4GHz/Platinum 8160, Intel Omni-Path	367,024	Texas Advanced Computing Center/Univ. of Texas, United States
Marconi Intel Xeon Phi - CINECA Cluster, Lenovo SD530/S720AP, Intel Xeon Phi 7250 68C 1.4GHz/Platinum 8160, Intel Omni-Path	312,936	CINECA, Italy
TSUBAME3.0 - SGI ICE XA, IP139-SXM2, Xeon E5-2680v4 14C 2.4GHz, Intel Omni- Path, NVIDIA Tesla P100 SXM2	135,828	GSIC Center, Tokyo Institute of Technology, Japan
MareNostrum - Lenovo SD530, Xeon Platinum 8160 24C 2.1GHz, Intel Omni-Path	153,216	Barcelona Supercomputing Center , Spain

C. $122300 / 8162 = 14.98$, or about **15x faster**

2. Below are the results:

A. $\text{Speedup} = 62 / 17 = \mathbf{3.65}$

$\text{Efficiency} = 62 / (17 \times 4) = 62 / 68 = \mathbf{91.18\%}$

B. $\text{Speedup} = 242 / 62 = \mathbf{3.90}$

$\text{Efficiency} = 242 / (62 \times 4) = 242 / 248 = \mathbf{97.58\%}$

C. Maximum speedup for 1GB = $62 / 2 = 31$

Maximum speedup for 10GB = $242 / 2 = 121$

D. Efficiency(1000 processors) = $242 / (2.24 * 1000) = 242 / 2240 = 10.80\%$

Efficiency(3000 processors) = $242 / (2.08 * 3000) = 242 / 6240 = 3.88\%$

Efficiency gain = $10.80 - 3.88 = 6.92\%$

3. Below is the pseudocode (**NOTE: I swapped out Pid == 0 with Pid == 1 for ease of**

calculations in the pseudocode):

```
If (Pid == 1) { /* this part is executed by processor whose Pid = 1*/
    max = my_max;
    /* receive maximum_value_1 and maximum_value_2 from processor i*2 and (i*2)+1; */
    maximum_value = max(maximum_value_1, maximum_value_2);
    if (maximum_value > max) {
        max = maximum_value;
    }
}
else { /* this part is executed by all other processors with Pid != 1 */
    /* receive max_value_1, max_value_2 from processor i*2 and (i*2)+1; */
    my_max = max(max_value_1, max_value_2);
    if (pid % 2 == 0) {
        /* send my_max to the processor whose Pid = pid/2 */ ;
    }
    else {
        /* send my_max to the processor whose Pid = (pid - 1)/2 */ ;
    }
}
```

4. Below is the pseudocode (**NOTE: I swapped out $Pid == 0$ with $Pid == 1$ for ease of calculations in the pseudocode**):

```
Let k = power of n (i.e. 2, 3, 4, ...)
If (Pid == 1) { /* this part is executed by processor whose Pid = 1 */
    max = my_max;
    /* receive maximum_value_list = [ maximum_value_1, maximum_value_2, ..., maximum_value_k from processor (i*k),
    (i*k)+1, ..., (i*k)+(k-1) ]; */
    for ( maximum_value : maximum_value_list ) {
        if (maximum_value > max) {
            max = maximum_value;
        }
    }
}
else { /* this part is executed by all other processors with Pid != 1 */
    /* receive max_value_list = [ max_value_1, max_value_2, ..., max_value_k from processor (i*k), (i*k)+1, ...,
    (i*k)+(k-1) ]; */
    my_max = max_value_list[0];
    for ( max_value : max_value_list ) {
        if max_value > max {
            my_max = max_value;
        }
    }
    if (pid % k == 0) {
        /* send my_max to the processor whose Pid = pid/k */ ;
    }
    else {
        for (i = pid; i > k; i--) {
            if (i % k == 0) {
                /* send my_max to the processor whose Pid = (pid - 1)/k */ ;
            }
        }
    }
}
}
```