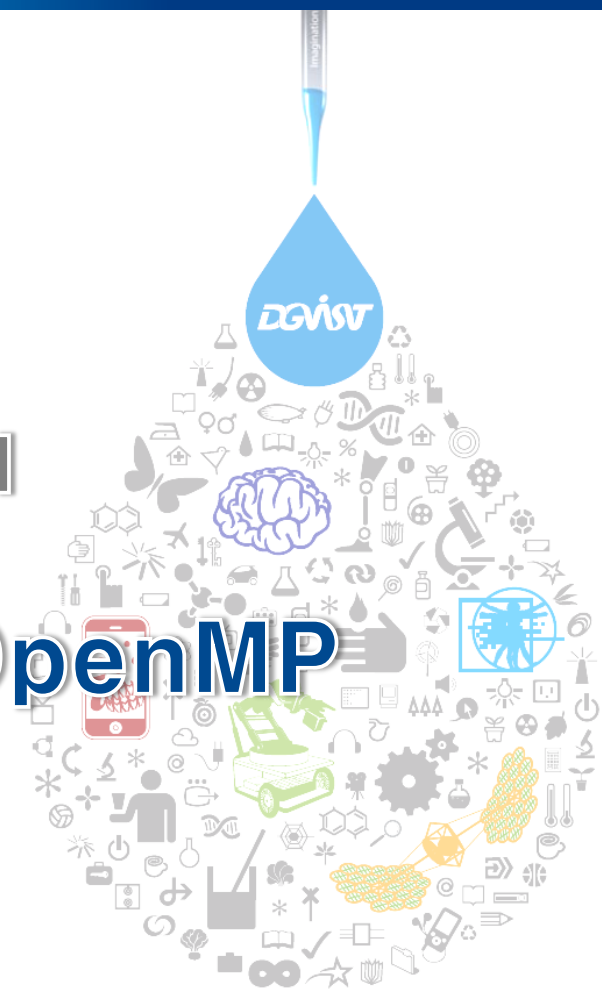


IC621: Distributed and Parallel Computing

Lab 06: Scheduling in OpenMP

Min-Soo Kim



Test performance of various scheduling

■ Implement odd-even transposition sort using OpenMP

- remove loop-carried dependences
- reuse the existing threads by using a `parallel` directive and a `for` directive
- three scheduling : **static, dynamic, guided**
 - `chunksize` of static and dynamic : **1, 100, 10000**

■ Create an input lists of **0.1M** random integer numbers

- range is between 1 and 1,000,000,000
- numbers should be generated by `rand()`

■ Measure the elapsed times of the program while varying scheduling and chunksize

- # of threads (or processes) : **4**

```

1  # pragma omp parallel num_threads(thread_count) \
2      default(none) shared(a, n) private(i, tmp, phase)
3      for (phase = 0; phase < n; phase++) {
4          if (phase % 2 == 0)
5              # pragma omp for
6                  for (i = 1; i < n; i += 2) {
7                      if (a[i-1] > a[i]) {
8                          tmp = a[i-1];
9                          a[i-1] = a[i];
10                         a[i] = tmp;
11                     }
12                 }
13             else
14                 # pragma omp for
15                     for (i = 1; i < n-1; i += 2) {
16                         if (a[i] > a[i+1]) {
17                             tmp = a[i+1];
18                             a[i+1] = a[i];
19                             a[i] = tmp;
20                         }
21                     }
22         }

```

Submission

- Source code

- Report

	scheduling		
chunksize	static	dynamic	guided
1			
100			
10000			

- Submit your codes by 1:00pm, 11/14(Tue) to TAs (bm010515@dgist.ac.kr, chan150@dgist.ac.kr)

Thank you!

