

## CSCI 4060U Lecture 10 - OpenMP Topics Review

### Parallel

- If Clause
- private/first private/shared/none

### Critical region

### Atomic statement

- Atomic clause
  - Read
  - Write
  - Update
  - capture

### Explicit Barrier

### For Loop, many clauses:

- Nowait
- Ordered
- Collapse for parallelizing nested loops
- Reduction (op: list) – example `#pragma omp parallel for reduction (+: sum)`
  - What operators exist for reduction in C++?

Reduction op	Initial value
+	0
*	1
-	0
min	Max +ve number
max	Min -ve number
&	~0
	0
^	0
&&	1
	0

- & is bitwise and while && is logical and

- **Example:**

a = 7, b = 5

(a == 6) & (b++==6)

-> evaluates both sides (also incrementing b) and returns false

(a == 6) && (b++==6)

-> evaluates left side which is false and returns false (right side not evaluated)

- **schedule**

Schedule clause	Impact on loop iteration mapping to threads
static	Decided at compile time and predictable by the programmer
dynamic	Decided at runtime – useful when loop iterations vary in workload
guided	(not really go into) Special case of dynamic to reduce overhead
auto	Runtime “learns” mapping from previous executions of the thread

- #pragma omp parallel for schedule(static [,chunk])  
where chunk is iteration size being allocated per thread
- #pragma omp parallel for schedule(dynamic [,chunk])
- #pragma omp parallel for schedule(guided [,chunk])
- #pragma omp parallel for schedule(runtime) where you don't specify the schedule and chunk size (comes from runtime library or OMP\_SCHEDULE variable)
- #pragma omp parallel for schedule(auto)