OpenMP Directives (3A)

Young Won Lim
3/4/19

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice.

Based on

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

parallel

Forms a team of threads and starts parallel execution.

#pragma omp parallel [clause[[,]clause] ...]
        structured-block
Clause:
        if(scalar-expression)
        num_threads(integer-expression)
        default(shared |none)
        private(list)
        firstprivate(list)
        shared(list)
        copyin(list)
        reduction(reduction-identifier: list)
        proc_bind(master | close | spread)

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

loop

Specifies that the iterations of associated loops will be executed in parallel by threads in the team in the context of their implicit tasks.

#pragma omp for [clause[[,]clause] ...]

for-loops clause:private(list)

firstprivate(list)

lastprivate(list)

redu**c**tion(reduction-identifier: list)

schedule(kind[, chunk_size])

collapse(n)

orderedn

owait

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

Loop kinds

Kind:

- **static**: Iterations are divided into chunks of size chunk_size and assigned to threads in the team in round-robin fashion in order of thread number.
- **dynamic**: Each thread executes a chunk of iterations then requests another chunk until none remain.
- **guided**: Each thread executes a chunk of iterations then requests another chunk until no chunks remain to be assigned.
- **auto**: The decision regarding scheduling is delegated to the compiler and/or runtime system.
- **runtime**: The schedule and chunk size are taken from the run-sched-var ICV.

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

sections

A noniterative worksharing construct that contains a set of structured blocks that are to be distributed among and executed by the threads in a team.

```
#pragma omp sections [clause[ [, ] clause] ...]
        {       [#pragma omp section]   structured-block
                [#pragma omp section]   structured-block       …
        }
```

Clause:

```
        private(list)
        firstprivate(list)
        lastprivate(list
        )reduction(reduction-identifier: list)
        nowait
```

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

single

Specifies that the associated structured block is executed by only one of the threads in the team.

#pragma omp single [clause[[,]clause] ...]

structured-block

Clause:

- **private**(list)
- **firstprivate**(list)
- **copyprivate**(list)
- **nowait**

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

declare simd

Enables the creation of one or more versions that can process multiple arguments using SIMD instructions from a single invocation from a SIMD loop.

```
#pragma omp declare simd [clause[ [, ]clause] ...]
        [#pragma omp declare simd [clause[ [, ]clause] …]
        ] [...]
                function definition or declaration
```

Clause:

simdlen(length)

linear(argument-list[:constant-linear-step])

aligned(argument-list[:alignment])

uniform(argument-list)

inbranch / notinbranch

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

declare simd

Enables the creation of one or more versions that can process multiple arguments using SIMD instructions from a single invocation from a SIMD loop.

```
#pragma omp declare simd [clause[ [, ]clause] ...]
        [#pragma omp declare simd [clause[ [, ]clause] …]
        ] [...]
                function definition or declaration
```

Clause:

- **simdlen**(length)
- **linear**(argument-list[:constant-linear-step])
- **aligned**(argument-list[:alignment])
- **uniform**(argument-list)
- **inbranch / notinbranch**

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

loop simd

Specifies that a loop that can be executed concurrently using SIMD instructions, and that those iterations will also be executed in parallel by threads in the team.

```
#pragma omp for simd [clause[ [, ]clause] ...]
        for-loops
```

Clause:

Any accepted by the simd or for directives with identical meanings and restrictions.

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

target

These constructs create a device data environment for the extent of the region. target also starts execution on the device.

```
#pragma omp target data [clause[ [, ]clause] ...]
        structured-block
#pragma omp target [clause[ [, ]clause] ...]
        structured-block
```

Clause:

- **device**(integer-expression)
- **map**([map-type:] list)
- **if**(scalar-expression)

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

target update

Makes the corresponding list items in the device data environment consistent with their original list items, according to the specified motion clauses.

```
#pragma omp target update clause[ [, ]clause] ,...]
```

clause is motion-clause or one of:

device(integer-expression)

if(scalar-expression)

Motion-clause:

to(list)

from(list)

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

declare target

A declarative directive that specifies that variables and functions are mapped to a device.

```
#pragma omp declare target
        declarations-definition-seq
#pragma omp end declare target
```

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

teams

Creates a league of thread teams where the master thread of each team executes the region.

#pragma omp teams [clause[[,]clause] ,...]
    structured-block

Clause:

- **num_teams**(integer-expression)
- **thread_limit**(integer-expression)
- **default**(shared | none)
- **private**(list)
- **firstprivate**(list)
- **shared**(list)
- **reduction**(reduction-identifier: list)

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

distribute

distribute specifies loops which are executed by the thread teams.

distribute simd specifies loops which are executed concurrently using SIMD instructions.

#pragma omp distribute [clause[[,]clause] ...]

    for-loops

#pragma omp distribute simd [clause[[,]clause] ...]

    for-loops

Clause:

- **private**(list)
- **firstprivate**(list)
- **collapse**(n)
- **dist_schedule**(kind[, chunk_size])

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

distribute parallel for [simd]

These constructs specify a loop that can be executed in parallel [using SIMD semantics in the simd case] by multiple threads that are members of multiple teams.

#pragma omp distribute parallel for [clause[[,]clause] ...]

  for-loops

#pragma omp distribute parallel for simd [clause[[,]clause] ...]

  for-loopsclause: See clause for distribute

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

parallel loop

Shortcut for specifying a parallel construct containing one or more associated loops and no other statements.

#pragma omp parallel for [clause[[,]clause] ...] for-loop

clause: Any accepted by the parallel or for directives, except the nowait clause, with identical meanings and restrictions.

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

parallel sections

Shortcut for specifying a parallel construct containing one sections construct and no other statements.

```
#pragma omp parallel sections [clause[ [, ]clause] ...]
        { [#pragma omp section]
                structured-block
          [#pragma omp section
                structured-block]
        …}
```

clause: Any of the clauses accepted by the parallel or sections directives, except the nowait clause, with identical meanings and restrictions.

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

parallel loop simd

Shortcut for specifying a parallel construct containing one loop SIMD construct and no other statements.

#pragma omp parallel for simd [clause[[,]clause] ...]

        for-loops

clause: Any accepted by the parallel, for or simddirectives, except the nowait clause, with identical meanings and restrictions.

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

target teams

Shortcut for specifying a target construct containing a teams construct.

#pragma omp target teams [clause[[,]clause] ...]
        structured-block

clause: See clause for target or teams

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

teams distribute [simd]

Shortcuts for specifying a teams construct containing a distribute [simd] construct.

#pragma omp teams distribute [clause[[,]clause] ...]

for-loops

#pragma omp teams distribute simd [clause[[,]clause] ...]

for-loops

clause: Any clause used for teams or distribute [simd]

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

target teams distribute [simd]

Shortcuts for specifying a target construct containing a teams distribute [simd] construct.

#pragma omp target teams distribute [clause[[,]clause] ...]

    for-loops

#pragma omp target teams distribute simd [clause[[,]clause] ...]

    for-loops

clause: Any clause used for target or teams distribute [simd]

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

teams distribute parallel for [simd]

Shortcuts for specifying a teams construct containing a distribute parallel for [simd] construct.

#pragma omp teams distribute parallel for [clause[[,]clause] ...]

        for-loops

#pragma omp teams distribute parallel for simd [clause[[,]clause] …]

        for-loops

clause: Any clause used for teams or distribute parallel for [simd]

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

target teams distribute parallel for [simd]

Shortcut for specifying a target construct containing a teams distribute parallel for [simd] construct.

```
#pragma omp target teams distribute parallel for \
        [clause[ [, ]clause] ...]
        for-loops
#pragma omp target teams distribute parallel for simd \

        [clause[ [, ]clause] ...]
        for-loops
```

clause: Any clause used for target or teams distribute parallel for [simd]

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

task (1)

Defines an explicit task. The data environment of the task is created according to data-sharing attribute clauses on task construct and any defaults that apply.

#pragma omp task [clause[[,]clause] ...] structured-block

clause:

- **if**(scalar-expression)
- **final**(scalar-expression)
- **untieddefault**(shared | none)
- **mergeableprivate**(list)
- **firstprivate**(list)
- **shared**(list)
- **depend**(dependence-type: list)

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

task (2)

#pragma omp task [clause[[,]clause] ...] structured-block

The list items that appear in the depend clause may include array sections.dependence-type: The generated task will be a dependent task of all previously generated sibling tasks that reference at least one of the list items...

- in: ...in an out or inout clause.
- out and inout: ...in an in, out, or inout clause.

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

taskyield

Specifies that the current task can be suspended in favor of execution of a different task.

#pragma omp taskyield

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

master

Specifies a structured block that is executed by the master thread of the team.

#pragma omp master structured-block

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

critical

Restricts execution of the associated structured block to a single thread at a time.

#pragma omp critical [(name)]   structured-block

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

barrier

Specifies an explicit barrier at the point at which the construct appears.

#pragma omp barrier

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

taskwait / taskgroup

These constructs each specify a wait on the completion of child tasks of the current task. taskgroup also waits for descendant tasks.

#pragma omp taskwait

#pragma omp taskgroup structured-block

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

atomic (1)

Ensures that a specific storage location is accessed atomically. [seq_cst] is 4.0.

#pragma omp atomic [read | write | update | capture] [seq_cst]expression-stmt

#pragma omp atomic capture [seq_cst]structured-blockwhere expression-stmt may be one of:

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

atomic (2)

if clause is...	expression-stmt:
read	v = x;
write	x = expr;
update or	x++; x--; ++x; --x;
is not present	xbinop= expr; x = x binop expr; x = expr binop x;
capture	v=x++; v=x--; v=++x; v= --x;

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

atomic (3)

and where structured-block may be one of the following forms:

```
{v = x; x binop= expr;}        {x binop= expr; v = x;}
{v = x; x = x binop expr;}     {v = x; x = expr binop x;}
{x = x binop expr; v = x;}     {x = expr binop x; v = x;}
{v = x; x = expr;}             {v = x; x++;}
{++x; v = x;}                  {x++; v = x;}
{v = x; x--;}                  {v = x; --x;}
{--x; v = x;}                  {x--; v = x;}
```

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

flush

Executes the OpenMP flush operation, which makes a thread’s temporary view of memory consistent with memory, and enforces an order on the memory operations of the variables.

#pragma omp flush [(list)]

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

ordered

Specifies a structured block in a loop region that will be executed in the order of the loop iterations.

```
#pragma omp ordered
    structured-block
```

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

cancel

Requests cancellation of the innermost enclosing region of the type specified. The cancel directive may not be used in place of the statement following an if, while, do, switch, or label.

#pragma omp cancel construct-type-clause[[,] if-clause]

construct-type-clause:

- **parallel**
- **sections**
- **for**
- **taskgroup**

if-clause:

- if(scalar-expression)

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

cancellation point

Introduces a user-defined cancellation point at which tasks check if cancellation of the innermost enclosing region of the type specified has been requested.

#pragma omp cancellation point construct-type-clause

construct-type-clause:

- **parallel**
- **sections**
- **for**
- **taskgroup**

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

threadprivate

Specifies that variables are replicated, with each thread having its own copy. Each copy of a threadprivate variable is initialized once prior to the first reference to that copy.

#pragma omp threadprivate(list)

list: A comma-separated list of file-scope, namespace-scope, or static block-scope variables that do not have incomplete types.

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

declare reduction

Declares a reduction-identifier that can be used in a reduction clause.

#pragma omp declare reduction(reduction-identifier :typename-list : combiner) [initializer-clause]

reduction-identifier: A base language identifier or one of the following operators: +, -, *, &, |, ^, && and ||In C++, this may also be an operator-function-id

typename-list: A list of type names

combiner: An expression

initializer-clause: **initializer** (**omp_priv** = initializer | function-name (argument-list))

https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

ICV (Internal Control Variable)

An OpenMP implementation must act
as if there are internal control variables (ICVs)
that control the behavior of an OpenMP program.

These ICVs store information such as
- the number of threads to use for future parallel regions,
- the schedule to use for worksharing loops and
- whether nested parallelism is enabled or not.

https://www.openmp.org/wp-content/uploads/OpenMP4.0.0.pdf

ICV (Internal Control Variable)

The ICVs are given values at various times
during the execution of the program.

They are initialized by the implementation itself
and may be given values
through OpenMP environment variables and
through calls to OpenMP API routines.

The program can retrieve the values of these ICVs
only through OpenMP API routines.

https://www.openmp.org/wp-content/uploads/OpenMP4.0.0.pdf

ICV (Internal Control Variable)

dyn-var - controls whether dynamic adjustment of the number of threads is enabled for encountered parallel regions. There is one copy of this ICV per data environment.

- nest-var - controls whether nested parallelism is enabled for encountered parallelregions. There is one copy of this ICV per data environment.

- nthreads-var - controls the number of threads requested for encountered parallelregions. There is one copy of this ICV per data environment.

- thread-limit-var - controls the maximum number of threads participating in the contention group. There is one copy of this ICV per data environment.

https://www.openmp.org/wp-content/uploads/OpenMP4.0.0.pdf

ICV (Internal Control Variable)

•max-active-levels-var - controls the maximum number of nested active parallelregions. There is one copy of this ICV per device.

•place-partition-var – controls the place partition available to the execution environment for encountered parallel regions. There is one copy of this ICV per implicit task.

•active-levels-var - the number of nested, active parallel regions enclosing the current task such that all of the parallel regions are enclosed by the outermost initial task region on the current device. There is one copy of this ICV per data environment.

https://www.openmp.org/wp-content/uploads/OpenMP4.0.0.pdf

ICV (Internal Control Variable)

- levels-var - the number of nested parallel regions enclosing the current task such that all of the parallel regions are enclosed by the outermost initial task region on the current device. There is one copy of this ICV per data environment.

- bind-var - controls the binding of OpenMP threads to places. When binding is requested, the variable indicates that the execution environment is advised not to move threads between places. The variable can also provide default thread affinity policies. There is one copy of this ICV per data environment. The following ICVs store values that affect the operation of loop regions.

https://www.openmp.org/wp-content/uploads/OpenMP4.0.0.pdf

ICV (Internal Control Variable)

- run-sched-var - controls the schedule that the runtime schedule clause uses for loop regions. There is one copy of this ICV per data environment.

- def-sched-var - controls the implementation defined default scheduling of loop regions. There is one copy of this ICV per device.

https://www.openmp.org/wp-content/uploads/OpenMP4.0.0.pdf

ICV (Internal Control Variable)

The following ICVs store values that affect the program execution.

- stacksize-var - controls the stack size for threads that the OpenMP implementation creates. There is one copy of this ICV per device.

- wait-policy-var - controls the desired behavior of waiting threads. There is one copy of this ICV per device.

- cancel-var - controls the desired behavior of the cancel construct and cancellation points. There is one copy of the ICV for the whole program (the scope is global).

default-device-var - controls the default target device. There is one copy of this ICV per data environment.

https://www.openmp.org/wp-content/uploads/OpenMP4.0.0.pdf

ICV Initialization

ICV	Environment Variable	Initial value
dyn-var	OMP_DYNAMIC	See comments below
nest-var	OMP_NESTED	false
nthreads-var	OMP_NUM_THREADS	Implementation defined
run-sched-var	OMP_SCHEDULE	Implementation defined
def-sched-var	(none)	Implementation defined
bind-var	OMP_PROC_BIND	Implementation defined
stacksize-var	OMP_STACKSIZE	Implementation defined
wait-policy-var	OMP_WAIT_POLICY I	mplementation defined
thread-limit-var	OMP_THREAD_LIMIT	Implementation defined
max-active-levels-var	OMP_MAX_ACTIVE_LEVELS	See comments below
active-levels-var	(none)	zero
levels-var	(none)	zero
place-partition-var	OMP_PLACES	Implementation defined
cancel-var	OMP_CANCELLATION	false
default-device-var	OMP_DEFAULT_DEVICE	Implementation defined

https://www.openmp.org/wp-content/uploads/OpenMP4.0.0.pdf

ICV Initialization

ICV	Ways to modify value	Way to retrieve value
dyn-var	omp_set_dynamic()	omp_get_dynamic()
nest-var	omp_set_nested()	omp_get_nested()
nthreads-var	omp_set_num_threads()	omp_get_max_threads()
run-sched-var	omp_set_schedule()	omp_get_schedule()
def-sched-var	(none)	(none)
bind-var	(none)	omp_get_proc_bind()
stacksize-var	(none)	(none)
Wait-policy-var	(none)	(none)
thread-limit-var	thread_limit clause	omp_get_thread_limit()
max-active-levels-var	omp_set_max_active_levels()	omp_get_max_active_levels()
active-levels-var	(none)	omp_get_active_levels()
levels-var	(none)	omp_get_level()
place-partition-var	(none)	(none)
cancel-var	(none)	omp_get_cancellation()
default-device-var	omp_set_default_device()	omp_get_default_device()

https://www.openmp.org/wp-content/uploads/OpenMP4.0.0.pdf

References

[1] ftp://ftp.geoinfo.tuwien.ac.at/navratil/HaskellTutorial.pdf

[2] https://www.umiacs.umd.edu/~hal/docs/daume02yaht.pdf