

2.5

parallel Construct	
32 2.5.1 Loop Construct	
. 38 2.5.2 sections Construct	
39 2.5.3 single Construct	
42 2.5.4 workshare Construct	
46 2.6.1 Parallel loop construct	
47 2.6.2 parallel sections Construct	
48 2.6.3 parallel workshare Construct	
51 2.7.1 master Construct	
51 2.7.2 critical Construct	
52 2.7.3 barrier Construct	
54 2.7.4 atomic Construct	
55 2.7.5 flush Construct	
58 2.7.6 ordered Construct	
65 2.8.2 threadprivate Directive	
70 2.8.3.1 default clause	
71 2.8.3.2 shared clause	
72 2.8.3.3 private clause	
73 2.8.3.4 firstprivate clause	
75 2.8.3.5 lastprivate clause	
77 2.8.3.6 reduction clause	
83 2.8.4.1 copyin clause	
84 2.8.4.2 copyprivate clause	
91 3.2.1 omp_set_num_threads	
91 3.2.2 omp_get_num_threads	
93 3.2.3 omp_get_max_threads	
94 3.2.4 omp_get_thread_num	
95 3.2.5 omp_get_num_procs	
96 3.2.6 omp_in_parallel	
96 3.2.7 omp_set_dynamic	

97	3.2.8	omp_get_dynamic
99	3.2.9	omp_set_nested
100	3.2.10	omp_get_nested
102	3.3.1	omp_init_lock and omp_init_nest_lock
104	3.3.2	omp_destroy_lock and omp_destroy_nest_lock
105	3.3.3	omp_set_lock and omp_set_nest_lock
105	3.3.4	omp_unset_lock and omp_unset_nest_lock
106	3.3.5	omp_test_lock and omp_test_nest_lock
108	3.4.1	omp_get_wtime
109	3.4.2	omp_get_wtick
113	4.1	OMP_SCHEDULE
114	4.2	OMP_NUM_THREADS
4.3		OMP_DYNAMIC
116	4.4	OMP_NESTED

3.0

.4 parallel Construct	
37 2.5.1 Loop Construct	
45 2.5.2 sections Construct	
47 2.5.3 single Construct	
49 2.5.4 workshare Construct	
54 2.6.1 Parallel Loop construct	
54 2.6.2 parallel sections Construct	
56 2.6.3 parallel workshare Construct	
58 2.7 task Construct	
63 2.8.1 master Construct	
63 2.8.2 critical Construct	
65 2.8.3 barrier Construct	
66 2.8.4 taskwait Construct	
68 2.8.5 atomic Construct	
69 2.8.6 flush Construct	
72 2.8.7 ordered Construct	
80 2.9.2 threadprivate Directive	
85 2.9.3.1 default clause	
86 2.9.3.2 shared clause	
88 2.9.3.3 private clause	
89 2.9.3.4 firstprivate clause	
92 2.9.3.5 lastprivate clause	
94 2.9.3.6 reduction clause	
100 2.9.4.1 copyin clause	
101 2.9.4.2 copyprivate clause	
109 3.2.1 omp_set_num_threads	
110 3.2.2 omp_get_num_threads	
111 3.2.3 omp_get_max_threads	
112 3.2.4 omp_get_thread_num	
113 3.2.5 omp_get_num_procs	

115	3.2.6	omp_in_parallel	
116	3.2.7	omp_set_dynamic	
117	3.2.8	omp_get_dynamic	
118	3.2.9	omp_set_nested	
119	3.2.10	omp_get_nested	
120	3.2.11	omp_set_schedule	
121	3.2.12	omp_get_schedule	
123	3.2.13	omp_get_thread_limit	
125	3.2.14	omp_set_max_active_levels	
126	3.2.15	omp_get_max_active_levels	
3.2.16		omp_get_level	
129	3.2.17	omp_get_ancestor_thread_num	
130	3.2.18	omp_get_team_size	
131	3.2.19	omp_get_active_level	
133	3.3	Lock Routines	
134	3.3.1	omp_init_lock and omp_init_nest_lock	
136	3.3.2	omp_destroy_lock and omp_destroy_nest_lock	...	
137	3.3.3	omp_set_lock and omp_set_nest_lock	
138	3.3.4	omp_unset_lock and omp_unset_nest_lock	
140	3.3.5	omp_test_lock and omp_test_nest_lock	
141	3.4	Timing Routines	
142	3.4.1	omp_get_wtime	
142	3.4.2	omp_get_wtick	
145	4.1	OMP_SCHEDULE	
146	4.2	OMP_NUM_THREADS	
147	4.3	OMP_DYNAMIC	
148	4.4	OMP_NESTED	
148	4.5	OMP_STACKSIZE	
149	4.6	OMP_WAIT_POLICY	150
		OMP_MAX_ACTIVE_LEVELS	4.7
150	4.8	OMP_THREAD_LIMIT		

3.1.

2.4 parallel Construct	
38 2.5.1 Loop Construct	
39 2.5.2 sections Construct	
48 2.5.3 single Construct	
50 2.5.4 workshare Construct	
55 2.6.1 Parallel Loop Construct	
56 2.6.2 parallel sections Construct	
57 2.6.3 parallel workshare Construct	
61 2.7.1 task Construct	
61 2.7.2 taskyield Construct	
67 2.8.1 master Construct	
67 2.8.2 critical Construct	
68 2.8.3 barrier Construct	
70 2.8.4 taskwait Construct	
72 2.8.5 atomic Construct	
73 2.8.6 flush Construct	
78 2.8.7 ordered Construct	
84 2.9.2 threadprivate Directive	
115 3.2.1 omp_set_num_threads	
116 3.2.2 omp_get_num_threads	
117 3.2.3 omp_get_max_threads	
118 3.2.4 omp_get_thread_num	
119 3.2.5 omp_get_num_procs	
121 3.2.6 omp_in_parallel	
122 3.2.7 omp_set_dynamic	
123 3.2.8 omp_get_dynamic	
124 3.2.9 omp_set_nested	
125 3.2.10 omp_get_nested	
126 3.2.11 omp_set_schedule	
128 3.2.12 omp_get_schedule	

130	3.2.13	omp_get_thread_limit
131	3.2.14	omp_set_max_active_levels
132	3.2.15	omp_get_max_active_levels
134	3.2.16	omp_get_level
135	3.2.17	omp_get_ancestor_thread_num
136	3.2.18	omp_get_team_size
137	3.2.19	omp_get_active_level
139	3.2.20	omp_in_final
140	3.3	Lock Routines
141	3.3.1	omp_init_lock and omp_init_nest_lock
143	3.3.2	omp_destroy_lock and omp_destroy_nest_lock
144	3.3.3	omp_set_lock and omp_set_nest_lock
3.3.4		omp_unset_lock and omp_unset_nest_lock
146	3.3.5	omp_test_lock and omp_test_nest_lock
148	3.4.1	omp_get_wtime
148	3.4.2	omp_get_wtick
4.1		OMP_SCHEDULE
154	4.2	OMP_NUM_THREADS
155	4.3	OMP_DYNAMIC
156	4.4	OMP_PROC_BIND
156	4.5	OMP_NESTED
157	4.6	OMP_STACKSIZE
157	4.7	OMP_WAIT_POLICY
158	4.8	OMP_MAX_ACTIVE_LEVELS
159	4.9	OMP_THREAD_LIMIT

4.0.

2.5 parallel Construct	
53 2.7.1 Loop Construct	
53 2.7.2 sections Construct	
60 2.7.3 single Construct	
63 2.7.4 workshare Construct	
68 2.8.1 simd construct	
68 2.8.2 declare simd construct	
72 2.8.3 Loop SIMD construct	
77 2.9.1 target data Construct	
77 2.9.2 target Construct	
79 2.9.3 target update Construct	
81 2.9.4 declare target Directive	
83 2.9.5 teams Construct	
86 2.9.6 distribute Construct	
88 2.9.7 distribute simd Construct	
91 iii 2.9.8 Distribute Parallel Loop Construct	
92 2.9.9 Distribute Parallel Loop SIMD Construct	
95 2.10.1 Parallel Loop Construct	
95 2.10.2 parallel sections Construct	
97 2.10.3 parallel workshare Construct	
99 2.10.4 Parallel Loop SIMD Construct	
100 2.10.5 target teams construct	
101 2.10.6 teams distribute Construct	
102 2.10.7 teams distribute simd Construct	
104 2.10.8 target teams distribute Construct	
105 2.10.9 target teams distribute simd Construct	
106 2.10.10 Teams Distribute Parallel Loop Construct	
107 2.10.11 Target Teams Distribute Parallel Loop Construct	
109 2.10.12 Teams Distribute Parallel Loop SIMD Construct	
110 2.10.13 Target Teams Distribute Parallel Loop SIMD Construct	

113	2.11.1 task Construct
113	2.11.2 taskyield Construct
120	2.12.1 master Construct
120	2.12.2 critical Construct
122	2.12.3 barrier Construct
123	2.12.4 taskwait Construct
125	2.12.5 taskgroup Construct
126	2.12.6 atomic Construct
127	2.12.7 flush Construct
134	2.12.8 ordered Construct
	2.13.1 cancel Construct
140	2.13.2 cancellation point Construct
146	2.14.2 threadprivate Directive
173	2.14.5 map Clause
177	2.15 declare reduction Directive
189	3.2.1 omp_set_num_threads
189	3.2.2 omp_get_num_threads
191	3.2.3 omp_get_max_threads
192	3.2.4 omp_get_thread_num
193	3.2.5 omp_get_num_procs
195	3.2.6 omp_in_parallel
196	3.2.7 omp_set_dynamic
197	3.2.8 omp_get_dynamic
198	3.2.9 omp_get_cancellation
199	3.2.10 omp_set_nested
200	3.2.11 omp_get_nested
201	3.2.12 omp_set_schedule
203	3.2.13 omp_get_schedule
205	3.2.14 omp_get_thread_limit
206	3.2.15 omp_set_max_active_levels
207	3.2.16 omp_get_max_active_levels

209	v 3.2.17	omp_get_level
210	3.2.18	omp_get_ancestor_thread_num
211	3.2.19	omp_get_team_size
212	3.2.20	omp_get_active_level
214	3.2.21	omp_in_final
215	3.2.22	omp_get_proc_bind
216	3.2.23	omp_set_default_device
218	3.2.24	omp_get_default_device
219	3.2.25	omp_get_num_devices
220	3.2.26	omp_get_num_teams
221	3.2.27	omp_get_team_num
222	3.2.28	omp_is_initial_device 2
224	3.3.1	omp_init_lock and omp_init_nest_lock
226	3.3.2	omp_destroy_lock and omp_destroy_nest_lock	...
227	3.3.3	omp_set_lock and omp_set_nest_lock
228	3.3.4	omp_unset_lock and omp_unset_nest_lock
229	3.3.5	omp_test_lock and omp_test_nest_lock
231	3.4	Timing Routines
233	3.4.1	omp_get_wtime
233	3.4.2	omp_get_wtick
237	4.1	OMP_SCHEDULE
238	4.2	OMP_NUM_THREADS
239	4.3	OMP_DYNAMIC
240	4.4	OMP_PROC_BIND
241	4.5	OMP_PLACES
241	4.6	OMP_NESTED
243	4.7	OMP_STACKSIZE
4.8		OMP_WAIT_POLICY
245	4.9	OMP_MAX_ACTIVE_LEVELS
245	4.10	OMP_THREAD_LIMIT
246	4.11	OMP_CANCELLATION

246 4.12 OMP_DISPLAY_ENV

247 4.13 OMP_DEFAULT_DEVICE

4.5.

2.5 parallel Construct	
56 2.7.1 Loop Construct	
64 2.7.2 sections Construct	
65 2.7.3 single Construct	
67 2.7.4 workshare Construct	
. 72 2.8.1 simd Construct	
72 2.8.2 declare simd Construct	
76 2.8.3 Loop SIMD Construct	
83 2.9.1 task Construct	
83 2.9.2 taskloop Construct	
87 2.9.3 taskloop simd Construct	
91 2.9.4 taskyield Construct	
95 2.10.1 target data Construct	
95 2.10.2 target enter data Construct	
97 2.10.3 target exit data Construct	
103 2.10.5 target update Construct	
107 2.10.6 declare target Directive	
110 2.10.7 teams Construct	
114 2.10.8 distribute Construct	
117 2.10.9 distribute simd Construct	
119 2.10.10 Distribute Parallel Loop Construct	
121 2.10.11 Distribute Parallel Loop SIMD Construct	
124 2.11.1 Parallel Loop Construct	
124 2.11.2 parallel sections Construct	
125 2.11.3 parallel workshare Construct	
127 2.11.4 Parallel Loop SIMD Construct	
128 2.11.5 target parallel Construct	
129 2.11.6 Target Parallel Loop Construct	
131 2.11.7 Target Parallel Loop SIMD Construct	
132 2.11.8 target simd Construct	

134	2.11.9 target teams Construct
135	2.11.10 teams distribute Construct
136	2.11.11 teams distribute simd Construct
137	2.11.12 target teams distribute Construct
139	2.11.13 target teams distribute simd Construct
140	2.11.14 Teams Distribute Parallel Loop Construct
141	2.11.15 Target Teams Distribute Parallel Loop Construct
142	2.11.16 Teams Distribute Parallel Loop SIMD Construct
144	2.11.17 Target Teams Distribute Parallel Loop SIMD Construct
145	2.12 if Clause
148	2.13.1 master Construct
148	2.13.2 critical Construct
149	2.13.3 barrier Construct
151	2.13.4 taskwait Construct
153	2.13.5 taskgroup Construct
155	2.13.7 flush Construct
162	2.13.8 ordered Construct
166	2.13.9 depend Clause
172	2.14.1 cancel Construct
172	2.14.2 cancellation point Construct
183	2.15.2 threadprivate Directive
	2.15.3.1 default Clause
189	2.15.3.2 shared Clause
190	2.15.3.3 private Clause
192	2.15.3.4 firstprivate Clause
196	2.15.3.5 lastprivate Clause
199	2.15.3.6 reduction Clause
201	2.15.3.7 linear Clause
211	2.15.4.1 copyin Clause
211	2.15.4.2 copyprivate Clause
215	2.15.5.1 map Clause

216	2.15.5.2 defaultmap Clause
219	2.16 declare reduction Directive
231	3.2.1 omp_set_num_threads
232	3.2.3 omp_get_max_threads
233	3.2.4 omp_get_thread_num
235	3.2.5 omp_get_num_procs
236	3.2.6 omp_in_parallel
236	3.2.7 omp_set_dynamic
237	3.2.8 omp_get_dynamic
239	3.2.9 omp_get_cancellation
240	3.2.10 omp_set_nested
240	3.2.11 omp_get_nested
242	3.2.12 omp_set_schedule
243	3.2.13 omp_get_schedule
245	3.2.14 omp_get_thread_limit
246	3.2.15 omp_set_max_active_levels
246	3.2.16 omp_get_max_active_levels
248	3.2.17 omp_get_level
249	3.2.18 omp_get_ancestor_thread_num
250	3.2.19 omp_get_team_size
251	3.2.20 omp_get_active_level
252	3.2.21 omp_in_final
253	3.2.22 omp_get_proc_bind
254	3.2.23 omp_get_num_places
256	3.2.24 omp_get_place_num_procs
257	3.2.25 omp_get_place_proc_ids
258	3.2.26 omp_get_place_num
259	3.2.27 omp_get_partition_num_places
260	3.2.28 omp_get_partition_place_nums
261	3.2.29 omp_set_default_device
262	3.2.30 omp_get_default_device

263	3.2.31	omp_get_num_devices
264	3.2.32	omp_get_num_teams
264	3.2.33	omp_get_team_num
266	3.2.34	omp_is_initial_device
267	Contents v 3.2.35	omp_get_initial_device
267	3.2.36	omp_get_max_task_priority
268	3.3	Lock Routines
270	3.3.1	omp_init_lock and omp_init_nest_lock
272	3.3.2	omp_init_lock_with_hint and omp_init_nest_lock_with_hint
273	3.3.3	omp_destroy_lock and omp_destroy_nest_lock
275	3.3.4	omp_set_lock and omp_set_nest_lock
276	3.3.5	omp_unset_lock and omp_unset_nest_lock
277	3.3.6	omp_test_lock and omp_test_nest_lock
278	3.4	Timing Routines
279	3.4.1	omp_get_wtime
279	3.4.2	omp_get_wtick
281	3.5	Device Memory Routines
282	3.5.1	omp_target_alloc
282	3.5.2	omp_target_free
283	3.5.3	omp_target_is_present
284	3.5.4	omp_target_memcpy
285	3.5.5	omp_target_memcpy_rect
286	3.5.6	omp_target_associate_ptr
287	3.5.7	omp_target_disassociate_ptr
290	4.1	OMP_SCHEDULE
292	4.2	OMP_NUM_THREADS
293	4.3	OMP_DYNAMIC
294	4.4	OMP_PROC_BIND
294	4.5	OMP_PLACES
295	4.6	OMP_NESTED
297	4.7	OMP_STACKSIZE

298	4.8	OMP_WAIT_POLICY
299	4.9	OMP_MAX_ACTIVE_LEVELS
300	4.10	OMP_THREAD_LIMIT
300	4.11	OMP_CANCELLATION
	4.12	OMP_DISPLAY_ENV
301	4.13	OMP_DEFAULT_DEVICE
302	4.14	OMP_MAX_TASK_PRIORITY

5.0.

2.3.5 declare variant Directive	
58 2.4 requires Directive	
. 72 2.6 parallel Construct	
. 80 2.7 teams Construct	
. 86 2.8.1 sections Construct	
86 2.8.2 single Construct	
89 2.8.3 workshare Construct	
95 2.9.1 Canonical Loop Form	
95 2.9.2 Worksharing-Loop Construct	
110 2.9.3.1 simd Construct	
110 2.9.3.2 Worksharing-Loop SIMD Construct	
114 2.9.3.3 declare simd Directive	
120 2.9.4.1 distribute Construct	
120 2.9.4.2 distribute simd Construct	
123 2.9.4.3 Distribute Parallel Worksharing-Loop Construct	
125 2.9.4.4 Distribute Parallel Worksharing-Loop SIMD Construct	
126 2.9.5 loop Construct	
128 2.9.6 scan Directive	
135 2.10.1 task Construct	
135 2.10.2 taskloop Construct	
140 2.10.3 taskloop simd Construct	
146 2.10.4 taskyield Construct	
147 2.10.5 Initial Task	
152 2.11.3 allocate Directive	
156 2.11.4 allocate Clause	
2.12.2 target data Construct	
161 2.12.3 target enter data Construct	
164 2.12.4 target exit data Construct	
166 2.12.5 target Construct	
170 2.12.6 target update Construct	

176	2.12.7	declare target Directive
185	2.13.1	Parallel Worksharing-Loop Construct
185	2.13.2	parallel loop Construct
186	2.13.3	parallel sections Construct
188	2.13.4	parallel workshare Construct
189	2.13.5	Parallel Worksharing-Loop SIMD Construct
190	2.13.6	parallel master Construct
191	2.13.7	master taskloop Construct
192	2.13.8	master taskloop simd Construct
194	2.13.9	parallel master taskloop Construct
195	2.13.10	parallel master taskloop simd Construct
196	2.13.11	teams distribute Construct
197	2.13.12	teams distribute simd Construct
198	2.13.13	Teams Distribute Parallel Worksharing-Loop Construct
200	2.13.14	Teams Distribute Parallel Worksharing-Loop SIMD Construct
201	2.13.15	teams loop Construct
202	2.13.16	target parallel Construct
203	2.13.17	Target Parallel Worksharing-Loop Construct
205	2.13.18	Target Parallel Worksharing-Loop SIMD Construct
206	2.13.19	target parallel loop Construct
208	2.13.20	target simd Construct
209	2.13.21	target teams Construct
210	2.13.22	target teams distribute Construct
211	2.13.23	target teams distribute simd Construct
213	2.13.24	target teams loop Construct
214	2.13.25	Target Teams Distribute Parallel Worksharing-Loop Construct
215	2.13.26	Target Teams Distribute Parallel Worksharing-Loop SIMD Construct
218	2.15	if Clause
220	2.16	master Construct
223	2.17.1	critical Construct
223	2.17.2	barrier Construct

230	2.17.5 taskwait Construct
230	2.17.6 taskgroup Construct
232	2.17.7 atomic Construct
234	2.17.8 flush Construct
246	2.17.9 ordered Construct
. 254	2.17.10.1 depobj Construct
254	2.17.11 depend Clause
263	2.18.1 cancel Construct
263	2.18.2 cancellation point Construct
273	2.19.2 threadprivate Directive
282	2.19.4.1 default Clause
282	2.19.4.2 shared Clause
283	2.19.4.3 private Clause
285	2.19.4.4 firstprivate Clause
286	2.19.4.5 lastprivate Clause
300	2.19.5.4 reduction Clause
300	2.19.5.5 task_reduction Clause
303	2.19.5.6 in_reduction Clause
303	2.19.5.7 declare reduction Directive
309	2.19.6.1 copyin Clause
310	2.19.6.2 copyprivate Clause
314	2.19.7.1 map Clause
315	2.19.7.2 defaultmap Clause
324	2.19.7.3 declare mapper Directive
334	3.2.1 omp_set_num_threads
334	3.2.2 omp_get_num_threads
335	3.2.3 omp_get_max_threads
336	3.2.4 omp_get_thread_num
337	3.2.5 omp_get_num_procs
338	3.2.6 omp_in_parallel
339	3.2.7 omp_set_dynamic

340	3.2.8	omp_get_dynamic
341	3.2.9	omp_get_cancellation
342	3.2.10	omp_set_nested
343	3.2.11	omp_get_nested
. 344	3.2.12	omp_set_schedule
. 345	3.2.13	omp_get_schedule
3.2.14		omp_get_thread_limit
348	3.2.15	omp_get_supported_active_levels
349	3.2.16	omp_set_max_active_levels
350	3.2.17	omp_get_max_active_levels
351	3.2.18	omp_get_level
352	3.2.19	omp_get_ancestor_thread_num
353	3.2.20	omp_get_team_size
354	3.2.21	omp_get_active_level
355	3.2.22	omp_in_final
356	3.2.23	omp_get_proc_bind
357	3.2.24	omp_get_num_places
358	3.2.25	omp_get_place_num_procs
359	3.2.26	omp_get_place_proc_ids
360	3.2.27	omp_get_place_num
362	3.2.28	omp_get_partition_num_places
362	3.2.29	omp_get_partition_place_nums
. 363	3.2.30	omp_set_affinity_format
364	3.2.31	omp_get_affinity_format
366	3.2.32	omp_display_affinity	
.. 367	3.2.33	omp_capture_affinity	
.. 368	3.2.34	omp_set_default_device	
369	3.2.35	omp_get_default_device	
370	3.2.36	omp_get_num_devices	
... 371	3.2.37	omp_get_device_num	
... 372	3.2.38	omp_get_num_teams	

....	373	3.2.39	omp_get_team_num	
.....	374	3.2.40	omp_is_initial_device	
.	375	3.2.41	omp_get_initial_device	
	376	3.2.42	omp_get_max_task_priority
	377	3.2.43	omp_pause_resource	
...	378	3.2.44	omp_pause_resource_all	
	381	3.3.1	omp_init_lock and omp_init_nest_lock
ii		3.3.2	omp_init_lock_with_hint and omp_init_nest_lock_with_hint
	385	3.3.3	omp_destroy_lock and omp_destroy_nest_lock
	387	3.3.4	omp_set_lock and omp_set_nest_lock
	388	3.3.5	omp_unset_lock and omp_unset_nest_lock
.	390	3.3.6	omp_test_lock and omp_test_nest_lock
	394	3.4.1	omp_get_wtime	
.....	394	3.4.2	omp_get_wtick	
.....	396	3.5.1	omp_fulfill_event	
.....	397	3.6.1	omp_target_alloc	
.....	397	3.6.2	omp_target_free	
.....	399	3.6.3	omp_target_is_present	
.	400	3.6.4	omp_target_memcpy	
....	400	3.6.5	omp_target_memcpy_rect	
	402	3.6.6	omp_target_associate_ptr
	403	3.6.7	omp_target_disassociate_ptr
..	406	3.7.2	omp_init_allocator	
...	409	3.7.3	omp_destroy_allocator	
.	410	3.7.4	omp_set_default_allocator
	411	3.7.5	omp_get_default_allocator
	412	3.7.6	omp_alloc	
.....	413	3.7.7	omp_free	
.....	601	6.2	OMP_NUM_THREADS	
.....	602	6.3	OMP_DYNAMIC	
.....	603	6.4	OMP_PROC_BIND	

..... 604 6.5 OMP_PLACES
..... 605 6.6 OMP_STACKSIZE
..... 607 6.7 OMP_WAIT_POLICY
..... 608 6.8 OMP_MAX_ACTIVE_LEVELS
.... 608 6.9 OMP_NESTED
..... 609 6.10 OMP_THREAD_LIMIT
..... 610 6.11 OMP_CANCELLATION
..... 610 6.12 OMP_DISPLAY_ENV
..... 611 6.13 OMP_DISPLAY_AFFINITY
.... 612 6.14 OMP_AFFINITY_FORMAT
..... 613 6.15 OMP_DEFAULT_DEVICE
..... 615 6.16 OMP_MAX_TASK_PRIORITY
.... 615 6.17 OMP_TARGET_OFFLOAD
..... 615 6.18 OMP_TOOL
..... 616 6.19 OMP_TOOL_LIBRARIES
..... 617 6.20 OMP_DEBUG
..... 617 6.21 OMP_ALLOCATOR
..... 618

5.1.

2.3.6 dispatch Construct

.. 83 2.5.1 requires Directive

..... 83 2.5.2 Assume Directive

..... 86 2.5.3 nothing Directive

..... 89 2.5.4 error Directive

..... 90 2.6 parallel Construct

2020 2.7 teams Construct

..... 100 2.8 masked Construct

..... 104 2.9 scope Construct

..... 108 2.10.1 sections Construct

..... 109 2.10.2 single Construct

..... 112 2.10.3 workshare Construct

... 125 2.11.3 order Clause

..... 125 2.11.4 Worksharing-Loop Construct

134 2.11.5.1 simd Construct

..... 134 2.11.5.2 Worksharing-Loop SIMD Construct

138 2.11.5.3 declare simd Directive

. 143 2.11.6.1 distribute Construct

.. 143 2.11.6.2 distribute simd Construct

147 2.11.6.3 Distribute Parallel Worksharing-Loop Construct

148 2.11.6.4 Distribute Parallel Worksharing-Loop SIMD Construct

149 2.11.7 loop Construct

..... 151 2.11.8 scan Directive

157 2.11.9.1 tile Construct

..... 158 2.11.9.2 unroll Construct

..... 161 2.12.1 task Construct

..... 161 2.12.2 taskloop Construct

..... 166 2.12.3 taskloop simd Construct

.. 171 2.12.4 taskyield Construct

..... 178 2.13.3 allocate Directive

.....	181	2.13.4	allocate Clause
.....	186	2.14.2	target data Construct
....	187	2.14.3	target enter data Construct
191	2.14.4	target exit data Construct	
193	2.14.5	target Construct	
.....	197	2.14.6	target update Construct
..	205	2.14.7	Declare Target Directive
.....	216	2.15.1	interop Construct
.....	221	2.16.1	Parallel Worksharing-Loop Construct
221	2.16.2	parallel loop Construct	
..	222	2.16.3	parallel sections Construct
223	2.16.4	parallel workshare Construct	
224	2.16.5	Parallel Worksharing-Loop SIMD Construct	
225	2.16.6	parallel masked Construct	
..	226	2.16.7	masked taskloop Construct
..	228	2.16.8	masked taskloop simd Construct
..	229	2.16.9	parallel masked taskloop Construct
230	2.16.10	parallel masked taskloop simd Construct	
231	2.16.11	teams distribute Construct	
233	2.16.12	teams distribute simd Construct	
234	2.16.13	Teams Distribute Parallel Worksharing-Loop Construct	
235	2.16.14	Teams Distribute Parallel Worksharing-Loop SIMD Construct	
236	2.16.15	teams loop Construct	
2020	2.16.16	target parallel Construct	
..	238	2.16.17	Target Parallel Worksharing-Loop Construct
239	2.16.18	Target Parallel Worksharing-Loop SIMD Construct	
241	2.16.19	target parallel loop Construct	
242	2.16.20	target simd Construct	
....	244	2.16.21	target teams Construct
...	245	2.16.22	target teams distribute Construct
246	2.16.23	target teams distribute simd Construct	

247	2.16.24	target teams loop Construct
248	2.16.25	Target Teams Distribute Parallel Worksharing-Loop Construct
249	2.16.26	Target Teams Distribute Parallel Worksharing-Loop SIMD Construct
252	2.18	if Clause
. 255	2.19.1	critical Construct
.	255 2.19.2	barrier Construct
261	2.19.5	taskwait Construct
.	261 2.19.6	taskgroup Construct
.	264 2.19.7	atomic Construct
.	266 2.19.8	flush Construct
.	279 2.19.9	ordered Construct
.	283 2.19.10	Depend Objects
.	286 2.19.10.1	depobj Construct
.	287 2.19.11	depend Clause
.	295 2.20.1	cancel Construct
.	295 2.20.2	cancellation point Construct
306	2.21.2	threadprivate Directive
. 315	2.21.4.1	default Clause
.	315 2.21.4.2	shared Clause
.	316 2.21.4.3	private Clause
.	318 2.21.4.4	firstprivate Clause
. . 318	2.21.4.5	lastprivate Clause
. . . 321	2.21.4.6	linear Clause
. 332	2.21.5.4	reduction Clause
. . . 332	2.21.5.5	task_reduction Clause
335	2.21.5.6	in_reduction Clause
. . 335	2.21.5.7	declare reduction Directive
.	341 2.21.6.1	copyin Clause
.	342 2.21.6.2	copyprivate Clause
2.21.7.1		map Clause
356	2.21.7.3	defaultmap Clause

...	357	2.21.7.4	declare mapper Directive
.....	368	3.2.1	omp_set_num_threads
...	368	3.2.2	omp_get_num_threads
...	369	3.2.3	omp_get_max_threads
...	370	3.2.4	omp_get_thread_num
...	371	3.2.5	omp_in_parallel
.....	372	3.2.6	omp_set_dynamic
.....	373	3.2.7	omp_get_dynamic
.....	373	3.2.8	omp_get_cancellation
..	374	3.2.9	omp_set_nested (Deprecated)
	375	3.2.10	omp_get_nested (Deprecated)
	376	3.2.11	omp_set_schedule
.....	376	3.2.12	omp_get_schedule
.....	379	3.2.13	omp_get_thread_limit
..	380	3.2.14	omp_get_supported_active_levels
	380	3.2.15	omp_set_max_active_levels
..	381	3.2.16	omp_get_max_active_levels
	382	3.2.17	omp_get_level
.....	383	3.2.18	omp_get_ancestor_thread_num
	384	3.2.19	omp_get_team_size
...	385	3.2.20	omp_get_active_level
..	385	3.3	Thread Affinity Routines
.....	386	3.3.1	omp_get_proc_bind
...	386	3.3.2	omp_get_num_places
...	388	3.3.3	omp_get_place_num_procs
	389	3.3.4	omp_get_place_proc_ids
	389	3.3.5	omp_get_place_num
...	390	3.3.6	omp_get_partition_num_places
	391	3.3.7	omp_get_partition_place_nums
	392	3.3.8	omp_set_affinity_format
	393	3.3.9	omp_get_affinity_format

394 3.3.10 omp_display_affinity
 .. 395 3.3.11 omp_capture_affinity
 397 3.4.1 omp_get_num_teams
 397 3.4.2 omp_get_team_num
 398 3.4.3 omp_set_num_teams
 399 3.4.4 omp_get_max_teams
 400 3.4.5 omp_set_teams_thread_limit
 400 3.4.6 omp_get_teams_thread_limit
 402 3.5.1 omp_get_max_task_priority
 402 3.5.2 omp_in_final
 ... 404 3.6.1 omp_pause_resource
 ... 404 3.6.2 omp_pause_resource_all
 407 3.7.1 omp_get_num_procs
 407 3.7.2 omp_set_default_device
 408 3.7.3 omp_get_default_device
 408 3.7.4 omp_get_num_devices
 ... 409 3.7.5 omp_get_device_num
 ... 410 3.7.6 omp_is_initial_device
 . 411 3.7.7 omp_get_initial_device
 412 3.8.1 omp_target_alloc
 412 3.8.2 omp_target_free
 414 3.8.3 omp_target_is_present
 . 416 3.8.4 omp_target_is_accessible
 417 3.8.5 omp_target_memcpy
 418 3.8.6 omp_target_memcpy_rect
 419 3.8.7 omp_target_memcpy_async
 422 3.8.8 omp_target_memcpy_rect_async
 424 3.8.9 omp_target_associate_ptr
 426 3.8.10 omp_target_disassociate_ptr
 429 3.8.11 omp_get_mapped_ptr
 432 3.9.1 omp_init_lock and omp_init_nest_lock

434	3.9.2	omp_init_lock_with_hint and omp_init_nest_lock_with_hint
. 435	3.9.3	omp_destroy_lock and omp_destroy_nest_lock
436	3.9.4	omp_set_lock and omp_set_nest_lock
437	3.9.5	omp_unset_lock and omp_unset_nest_lock
. 439	3.9.6	omp_test_lock and omp_test_nest_lock
.....	442	3.10.1 omp_get_wtime
.....	442	3.10.2 omp_get_wtick
.....	443	3.11.1 omp_fulfill_event
.....	444	3.12.1 omp_get_num_interop_properties
446	3.12.2	omp_get_interop_int
... 446	3.12.3	omp_get_interop_ptr
... 447	3.12.4	omp_get_interop_str
... 448	3.12.5	omp_get_interop_name
.. 449	3.12.6	omp_get_interop_type_desc
450	3.12.7	omp_get_interop_rc_desc
.. 451	3.13.2	omp_init_allocator
... 454	3.13.3	omp_destroy_allocator
. 455	3.13.4	omp_set_default_allocator
456	3.13.5	omp_get_default_allocator
457	3.13.6	omp_alloc and omp_aligned_alloc
458	3.13.7	omp_free
459	3.13.8	omp_calloc and omp_aligned_calloc
461	3.13.9	omp_realloc
.....	640	6.2 OMP_NUM_THREADS
.....	640	6.3 OMP_DYNAMIC
.....	641	6.4 OMP_PROC_BIND
.....	642	6.5 OMP_PLACES
.....	643	6.6 OMP_STACKSIZE
.....	645	6.7 OMP_WAIT_POLICY
.....	646	6.8 OMP_MAX_ACTIVE_LEVELS

.... 647 6.9 OMP_NESTED (Deprecated)
..... 647 6.10 OMP_THREAD_LIMIT
..... 648 6.11 OMP_CANCELLATION
..... 648 6.12 OMP_DISPLAY_ENV
..... 648 6.13 OMP_DISPLAY_AFFINITY
.... 649 6.14 OMP_AFFINITY_FORMAT
..... 650 6.15 OMP_DEFAULT_DEVICE
..... 652 2020 6.16 OMP_MAX_TASK_PRIORITY
.... 652 6.17 OMP_TARGET_OFFLOAD
..... 652 6.18 OMP_TOOL
..... 653 6.19 OMP_TOOL_LIBRARIES
..... 653 6.20 OMP_TOOL_VERBOSE_INIT
.... 654 6.21 OMP_DEBUG
..... 655 6.22 OMP_ALLOCATOR
..... 655 6.23 OMP_NUM_TEAMS
..... 656 6.24 OMP_TEAMS_THREAD_LIMIT

5.2.

3.4 if Clause

..... 72 3.5 destroy Clause

. 91 4.4.3 collapse Clause

..... 93 4.4.4 ordered Clause

100 5.2 threadprivate Directive

.... 101 5.3 List Item Privatization

.. 108 5.4.1 default Clause

..... 109 5.4.2 shared Clause

..... 110 5.4.3 private Clause

..... 111 5.4.4 firstprivate Clause

.... 112 5.4.5 lastprivate Clause

.... 115 5.4.6 linear Clause

..... 117 5.4.7 is_device_ptr Clause

... 120 5.4.8 use_device_ptr Clause

... 121 5.4.9 has_device_addr Clause

.. 122 5.4.10 use_device_addr Clause

128 5.5.4 initializer Clause

. 134 5.5.8 reduction Clause

..... 134 5.5.9 task_reduction Clause

... 137 5.5.10 in_reduction Clause

.... 138 5.5.11 declare reduction Directive

139 5.6 scan Directive

..... 141 5.6.1 inclusive Clause

..... 143 5.6.2 exclusive Clause

..... 144 5.7.1 copyin Clause

..... 144 5.7.2 copyprivate Clause

149 5.8.3 map Clause

..... 150 5.8.4 enter Clause

..... 158 5.8.5 link Clause

160 5.8.7 defaultmap Clause

.....	161	5.8.8 declare mapper Directive
.....	165	5.9.1 to Clause
.....	166	5.9.2 from Clause
.....	167	5.10 uniform Clause
.....	168	5.11 aligned Clause
.....	172	6.3 align Clause
.....	174	6.4 allocator Clause
.....	175	6.5 allocate Directive
.....	176	6.6 allocate Clause
.....	178	6.7 allocators Construct
.....	180	6.8 uses_allocators Clause
.....	183	7.2 Context Selectors
.....	189	7.4.1 when Clause
.....	191	7.4.3 metadirective
.....	192	7.4.4 begin metadirective
.....	193	7.5.1 match Clause
.....	194	7.5.2 adjust_args Clause
.....	195	7.5.3 append_args Clause
.....	196	7.5.4 declare variant Directive
. 197	7.5.5	begin declare variant Directive
198	7.6	dispatch Construct
.....	200	7.6.1 novariants Clause
.....	201	7.6.2 nocontext Clause
.....	201	7.7 declare simd Directive
.....	204	7.8.1 declare target Directive
.. 206	7.8.2	begin declare target Directive
207	7.8.3	indirect Clause
.....	210	8.2 requires Directive
.....	213	8.3.2 assumes Directive
.....	214	8.3.3 assume Directive
.....	215	8.3.4 begin assumes Directive

... 215 8.4 nothing Directive
..... 216 8.5 error Directive
..... 216 8.5.1 severity Clause
..... 217 8.5.2 message Clause
..... 219 9.1.1 sizes Clause
..... 220 9.2 unroll Construct
..... 220 9.2.1 full Clause
..... 221 9.2.2 partial Clause
..... 221 10.1 parallel Construct
10.1.2 num_threads Clause
228 10.1.4 proc_bind Clause
..... 229 10.2 teams Construct
..... 230 10.2.1 num_teams Clause
..... 233 10.3 order Clause
..... 233 10.4 simd Construct
..... 235 10.4.1 nontemporal Clause
..... 236 10.4.2 safelen Clause
..... 237 10.4.3 simdlen Clause
..... 237 10.5 masked Construct
..... 238 10.5.1 filter Clause
..... 239 240 11.1 single Construct
..... 240 11.2 scope Construct
..... 242 11.3 sections Construct
..... 243 11.3.1 section Directive
..... 244 11.4 workshare Construct
... 247 11.5.1 for Construct
..... 250 11.5.2 do Construct
..... 251 11.5.3 schedule Clause
..... 252 distribute Construct
..... 254 11.6.1 dist_schedule Clause
... 256 11.7 loop Construct

.....	257	11.7.1	bind Clause
.....	260	12.1	untied Clause
.....	260	12.2	mergeable Clause
.....	260	12.3	final Clause
.....	261	12.4	priority Clause
.....	261	12.5	task Construct
.....	262	12.5.1	affinity Clause
.....	264	12.5.2	detach Clause
.....	265	12.6	taskloop Construct
.....	266	12.6.1	grainsize Clause
.....	269	12.6.2	num_tasks Clause
.....	270	12.7	taskyield Construct
.....	272	13 13.1	device_type Clause
.....	275	13.2	device Clause
.....	276	13.3	thread_limit Clause
.....	277	13.4	Device Initialization
.....	278	13.5	target data Construct
....	279	13.6	target enter data Construct
.	280	13.7	target exit data Construct
..	282	13.8	target Construct
.....	284	13.9	target update Construct
291	14.1		interop Construct
293	14.1.2		init Clause
.....	293	14.1.3	use Clause
...	296	15.1.2	hint Clause
.....	299	15.2	critical Construct
.....	301	15.3.1	barrier Construct
304	15.4		taskgroup Construct
.....	304	15.5	taskwait Construct
.....	306	15.6	nowait Clause
.....	308	15.7	nogroup Clause

.....	310	15.8.4	atomic Construct
.....	311	15.8.5	flush Construct
.....	321	15.9.3	update Clause
.....	321	15.9.4	depobj Construct
.....	322	15.9.5	depend Clause
.....	323	15.9.6	doacross Clause
.....	326	15.10	ordered Construct
329	15.10.		ordered Construct
...	332	16.1	cancel Construct
.....	332	16.2	cancellation point Construct
.....	348	18.2.1	omp_set_num_threads
...	348	18.2.2	omp_get_num_threads
...	349	18.2.3	omp_get_max_threads
...	350	18.2.4	omp_get_thread_num
...	350	18.2.5	omp_in_parallel
.....	351	18.2.6	omp_set_dynamic
.....	352	18.2.7	omp_get_dynamic
.....	352	18.2.8	omp_get_cancellation
..	353	18.2.9	omp_set_nested (Deprecated)
353	18.2.10	omp_get_nested (Deprecated)	
354	18.2.11	omp_set_schedule	
.....	355	18.2.12	omp_get_schedule
.....	356	18.2.13	omp_get_thread_limit
..	357	18.2.14	omp_get_supported_active_levels
358	18.2.15	omp_set_max_active_levels	
. 358	18.2.16	omp_get_max_active_levels	
359	18.2.17	omp_get_level	
.....	360	18.2.18	omp_get_ancestor_thread_num
360	18.2.19	omp_get_team_size	
....	361	18.2.20	omp_get_active_level
.....	363	18.3.1	omp_get_proc_bind

- 363 18.3.2 omp_get_num_places
- ... 364 18.3.3 omp_get_place_num_procs
- 365 18.3.4 omp_get_place_proc_ids
- 365 18.3.5 omp_get_place_num
- 366 18.3.6 omp_get_partition_num_places
- 367 18.3.7 omp_get_partition_place_nums
- 368 18.3.8 omp_set_affinity_format
- 368 18.3.9 omp_get_affinity_format
- 369 18.3.10 omp_display_affinity
- .. 370 18.3.11 omp_capture_affinity
- 372 18.4.1 omp_get_num_teams
- 372 18.4.2 omp_get_team_num
- 373 18.4.3 omp_set_num_teams
- 373 18.4.4 omp_get_max_teams
- 374 18.4.5 omp_set_teams_thread_limit
- 375 18.4.6 omp_get_teams_thread_limit
- 377 18.5.1 omp_get_max_task_priority
- 377 18.5.2 omp_in_explicit_task
- .. 377 18.5.3 omp_in_final
- . 378 18.6.1 omp_pause_resource
- ... 378 18.6.2 omp_pause_resource_all
- 381 18.7.1 omp_get_num_procs
- 381 18.7.2 omp_set_default_device
- 382 18.7.3 omp_get_default_device
- 382 18.7.4 omp_get_num_devices
- ... 383 18.7.5 omp_get_device_num
- ... 384 18.7.6 omp_is_initial_device
- . 384 18.7.7 omp_get_initial_device
- 385 18.8.1 omp_target_alloc
- 385 18.8.2 omp_target_free
- 387 18.8.3 omp_target_is_present

. 389	18.8.4	omp_target_is_accessible
390	18.8.5	omp_target_memcpy
. . . . 391	18.8.6	omp_target_memcpy_rect
392	18.8.7	omp_target_memcpy_async
394	18.8.8	omp_target_memcpy_rect_async
396	18.8.9	omp_target_associate_ptr
399	18.8.10	omp_target_disassociate_ptr
401	18.8.11	omp_get_mapped_ptr
. 403	18.9.1	omp_init_lock and omp_init_nest_lock
405	18.9.2	omp_init_lock_with_hint and omp_init_nest_lock_with_hint
406	18.9.3	omp_destroy_lock and omp_destroy_nest_lock
408	18.9.4	omp_set_lock and omp_set_nest_lock
409	18.9.5	omp_unset_lock and omp_unset_nest_lock
410	18.9.6	omp_test_lock and omp_test_nest_lock
. 413	18.10.1	omp_get_wtime
. 413	18.10.2	omp_get_wtick
. 414	18.11	Event Routine
. 414	18.11.1	omp_fulfill_event
. 416	18.12.1	omp_get_num_interop_properties
417	18.12.2	omp_get_interop_int
. . . 417	18.12.3	omp_get_interop_ptr
. . . 418	18.12.4	omp_get_interop_str
. . . 419	18.12.5	omp_get_interop_name
. . 420	18.12.6	omp_get_interop_type_desc
421	18.12.7	omp_get_interop_rc_desc
. . 422	18.13.2	omp_init_allocator
. . . 425	18.13.3	omp_destroy_allocator
. 426	18.13.4	omp_set_default_allocator
427	18.13.5	omp_get_default_allocator
428	18.13.6	omp_alloc and omp_aligned_alloc
428	18.13.7	omp_free

..... 430 18.13.8 omp_malloc and omp_aligned_malloc

431 18.13.9 omp_realloc

600 21.1.1 OMP_DYNAMIC

..... 600 21.1.2 OMP_NUM_THREADS

..... 600 21.1.3 OMP_THREAD_LIMIT

..... 601 21.1.4 OMP_MAX_ACTIVE_LEVELS

. 601 21.1.5 OMP_NESTED (Deprecated)

... 602 21.1.6 OMP_PLACES

..... 602 21.1.7 OMP_PROC_BIND

21.2.1 OMP_SCHEDULE

..... 605 21.2.2 OMP_STACKSIZE

..... 606 21.2.3 OMP_WAIT_POLICY

..... 607 21.2.4 OMP_DISPLAY_AFFINITY

.. 607 21.2.5 OMP_AFFINITY_FORMAT

... 608 21.2.6 OMP_CANCELLATION

..... 610 21.2.7 OMP_DEFAULT_DEVICE

... 610 21.2.8 OMP_TARGET_OFFLOAD

... 610 21.2.9 OMP_MAX_TASK_PRIORITY

... 611 21.3.1 OMP_TOOL

..... 611 21.3.2 OMP_TOOL_LIBRARIES

... 612 21.3.3 OMP_TOOL_VERBOSE_INIT

... 613 21.4.1 OMP_DEBUG

614 21.5.1 OMP_ALLOCATOR

... 614 21.6.1 OMP_NUM_TEAMS

..... 615 21.6.2 OMP_TEAMS_THREAD_LIMIT

615 21.7 OMP_DISPLAY_ENV ...