A place consists of one or more processors. Pinning on the level of places.
Free migration of the threads on a place between the processors of that place.

- `setenv OMP_PLACES threads`  
  → Each place corresponds to the single processor of a single hardware thread (hyper-thread)
- `setenv OMP_PLACES cores`  
  → Each place corresponds to the processors (one or more hardware threads) of a single core
- `setenv OMP_PLACES sockets`  
  → Each place corresponds to the processors of a single socket (consisting of all hardware threads of one or more cores)
- `setenv OMP_PLACES abstract_name(num_places)`  
  → In general, the number of places may be explicitly defined
  - Or with explicit numbering, e.g. 8 places, each consisting of 4 processors:
    - `setenv OMP_PLACES "{0,1,2,3},{4,5,6,7},{8,9,10,11}, ... {28,29,30,31}"`
    - `setenv OMP_PLACES "{0:4},{4:4},{8:4}, ... {28:4}"`
    - `setenv OMP_PLACES "{0:4}:8:4"`

CAUTION:
The numbers highly depend on hardware and operating system, e.g.,
{0,1} = hyper-threads of 1st core of 1st socket, or
{0,1} = 1st hyper-thread of 1st core of 1st and 2nd socket, or …
OMP_PROC_BIND variable / proc_bind() clause

- Determines how places are used for pinning:

<table>
<thead>
<tr>
<th>OMP_PROC_BIND</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FALSE</td>
<td>Affinity disabled</td>
</tr>
<tr>
<td>TRUE</td>
<td>Affinity enabled, implementation defined strategy</td>
</tr>
<tr>
<td>CLOSE</td>
<td>Threads bind to consecutive places</td>
</tr>
<tr>
<td>SPREAD</td>
<td>Threads are evenly scattered among places</td>
</tr>
<tr>
<td>MASTER</td>
<td>Threads bind to the same place as the master thread that was running before the parallel region was entered</td>
</tr>
</tbody>
</table>

- If there are more threads than places, consecutive threads are put into individual places ("balanced")
Some simple OMP_PLACES examples

- Intel Xeon w/ SMT, 2x10 cores, 1 thread per physical core, fill 1 socket
  
  ```
  OMP_NUM_THREADS=10
  OMP_PLACES=cores
  OMP_PROC_BIND=close
  ```

- Intel Xeon Phi with 72 cores,
  
  32 cores to be used, 2 threads per physical core
  
  ```
  OMP_NUM_THREADS=64
  OMP_PLACES=cores(32)
  OMP_PROC_BIND=close  # spread will also do
  ```

- Intel Xeon, 2 sockets, 4 threads per socket (no binding within socket!)
  
  ```
  OMP_NUM_THREADS=8
  OMP_PLACES=sockets
  OMP_PROC_BIND=close  # spread will also do
  ```

- Intel Xeon, 2 sockets, 4 threads per socket, binding to cores
  
  ```
  OMP_NUM_THREADS=8
  OMP_PLACES=cores
  OMP_PROC_BIND=spread
  ```

Always prefer abstract places instead of HW thread IDs!