Parallelism on CPU What's hidden from the eyes?

Kumbrasev Pavel
Kochin Ivan
Arutyunyan Ruslan

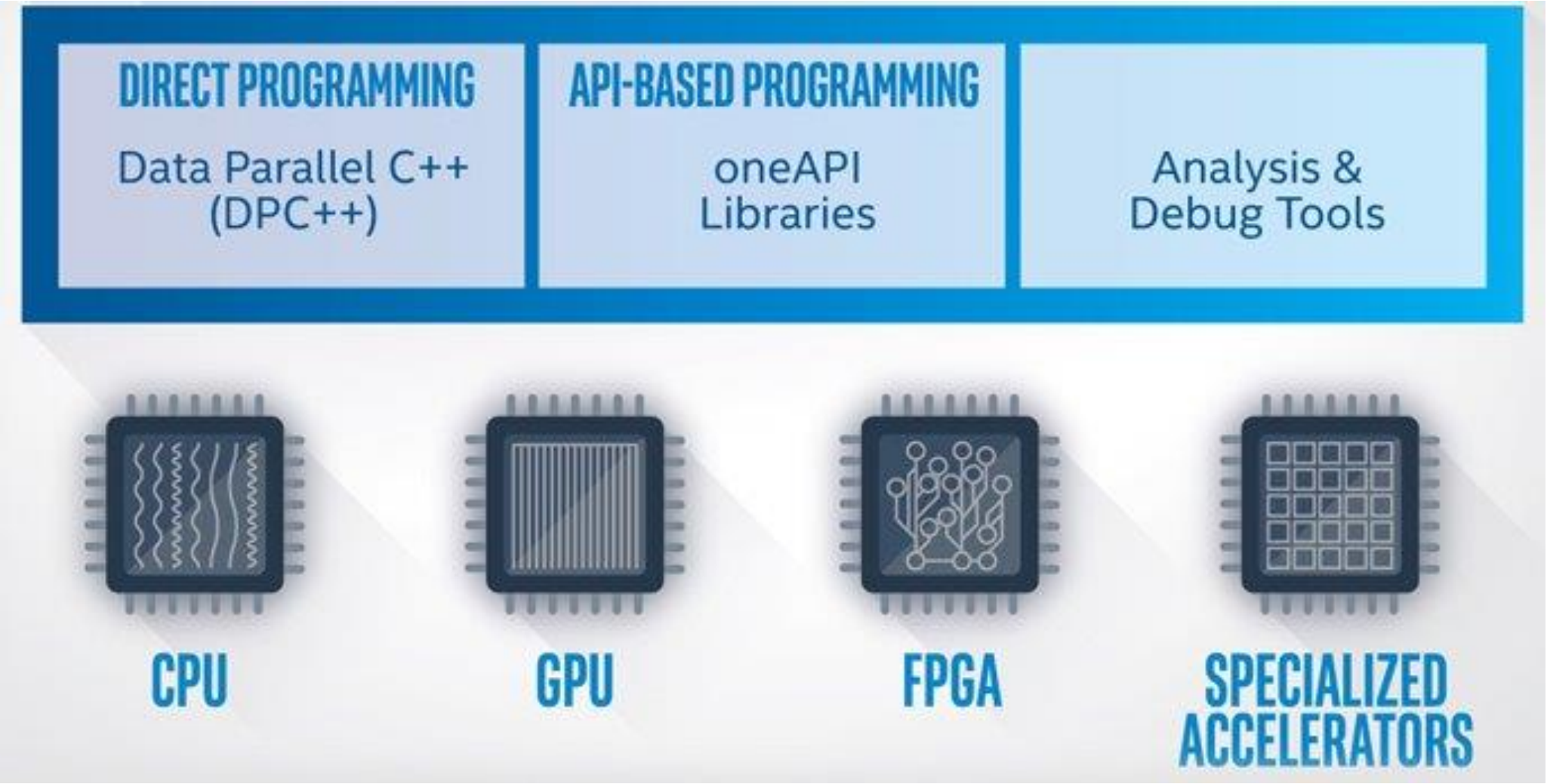
intel®

Agenda

- Intel® oneAPI overview
- Intel® oneAPI Threading Building Blocks (oneTBB) overview
- Problem statement
- Parallel solution with oneTBB
- Self-made task group
- Synchronization mechanisms
- Possible improvements

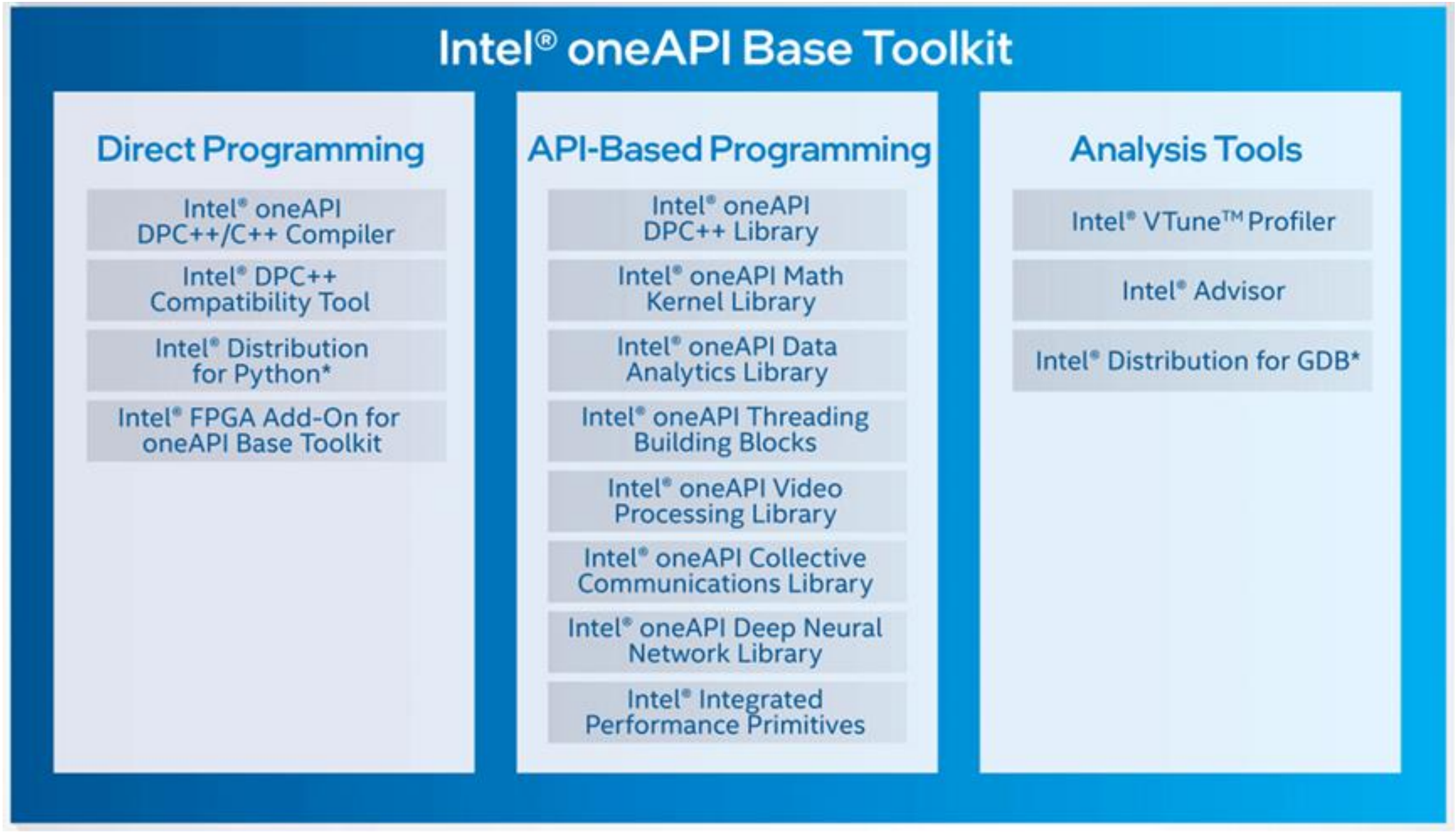
Intel® oneAPI overview

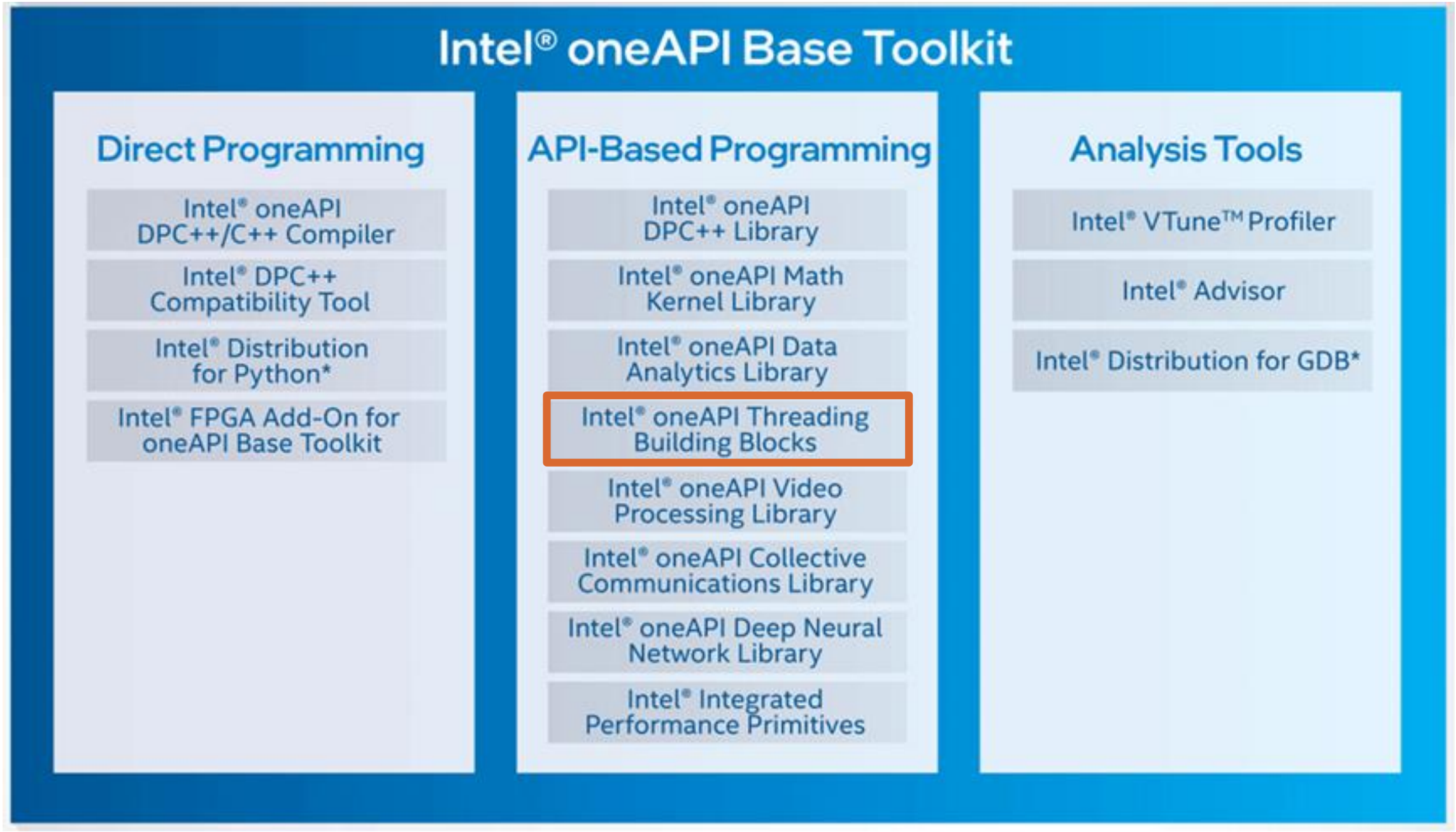
oneAPI: Unified cross-architecture programming



Why oneAPI?

- Unified cross-architecture programming
- Standards-based (C++, DPC++)
- Stable and open API
- Backward compatibility


Intel® oneAPI Base Toolkit
Direct Programming
Intel® oneAPI DPC++/C++ Compiler
Intel® DPC++ Compatibility Tool
Intel® Distribution for Python*
Intel® FPGA Add-On for oneAPI Base Toolkit
API-Based Programming
Intel® oneAPI DPC++ Library
Intel® oneAPI Math Kernel Library
Intel® oneAPI Data Analytics Library
Intel® oneAPI Threading Building Blocks
Intel® oneAPI Video Processing Library
Intel® oneAPI Collective Communications Library
Intel® oneAPI Deep Neural Network Library
Intel® Integrated Performance Primitives
Analysis Tools
Intel® VTune™ Profiler
Intel® Advisor
Intel® Distribution for GDB*


Intel® oneAPI Base Toolkit
Direct Programming
Intel® oneAPI DPC++/C++ Compiler
Intel® DPC++ Compatibility Tool
Intel® Distribution for Python*
Intel® FPGA Add-On for oneAPI Base Toolkit
API-Based Programming
Intel® oneAPI DPC++ Library
Intel® oneAPI Math Kernel Library
Intel® oneAPI Data Analytics Library
Intel® oneAPI Threading Building Blocks
Intel® oneAPI Video Processing Library
Intel® oneAPI Collective Communications Library
Intel® oneAPI Deep Neural Network Library
Intel® Integrated Performance Primitives
Analysis Tools
Intel® VTune™ Profiler
Intel® Advisor
Intel® Distribution for GDB*

oneTBB overview

oneTBB overview

Parallel algorithms and data structures

Threads and synchronization

Memory allocation and task scheduling

Generic Parallel Algorithms

Efficient scalable way to exploit the power of multi-core without having start to scratch

Flow Graph

A set of classes to express parallelism as a graph of compute dependencies and/or data flow

Concurrent Containers

Concurrent access and a scalable alternative to serial containers with external locking

Synchronization Primitives

A variety of mutexes with different properties

Task Scheduler

Sophisticated work scheduling engine that empowers parallel algorithms and flow graph

Thread Local Storage

Unlimited number of thread local variables

Miscellaneous

Thread-safe timers, system topology traversing

Memory allocation

Scalable memory management, Thread-safe allocator

oneTBB role in oneAPI

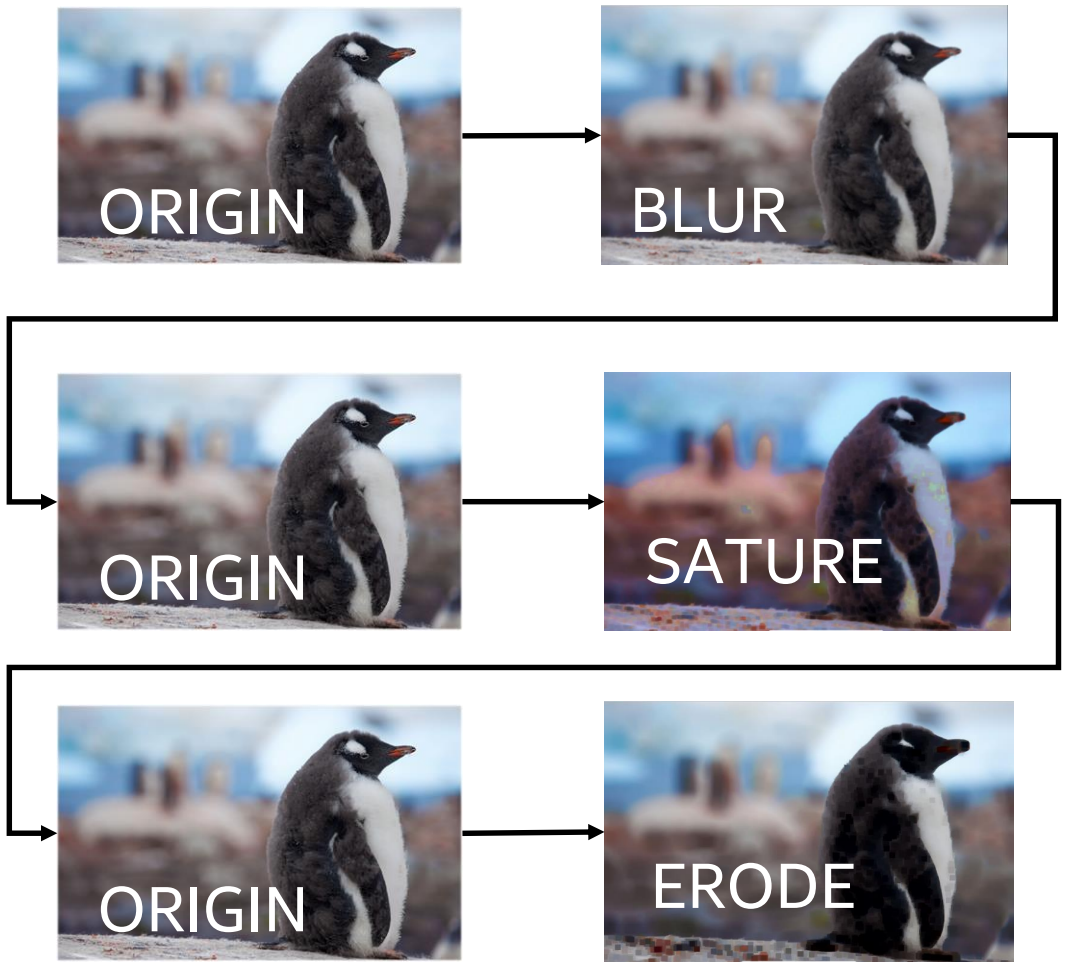
- Public API for CPU parallelism
- Internal layer for parallelism on CPU used by:
  - DPC++/C++ Compiler
  - DPC++ Library
  - Various oneAPI libraries

Problem statement

Problem statement

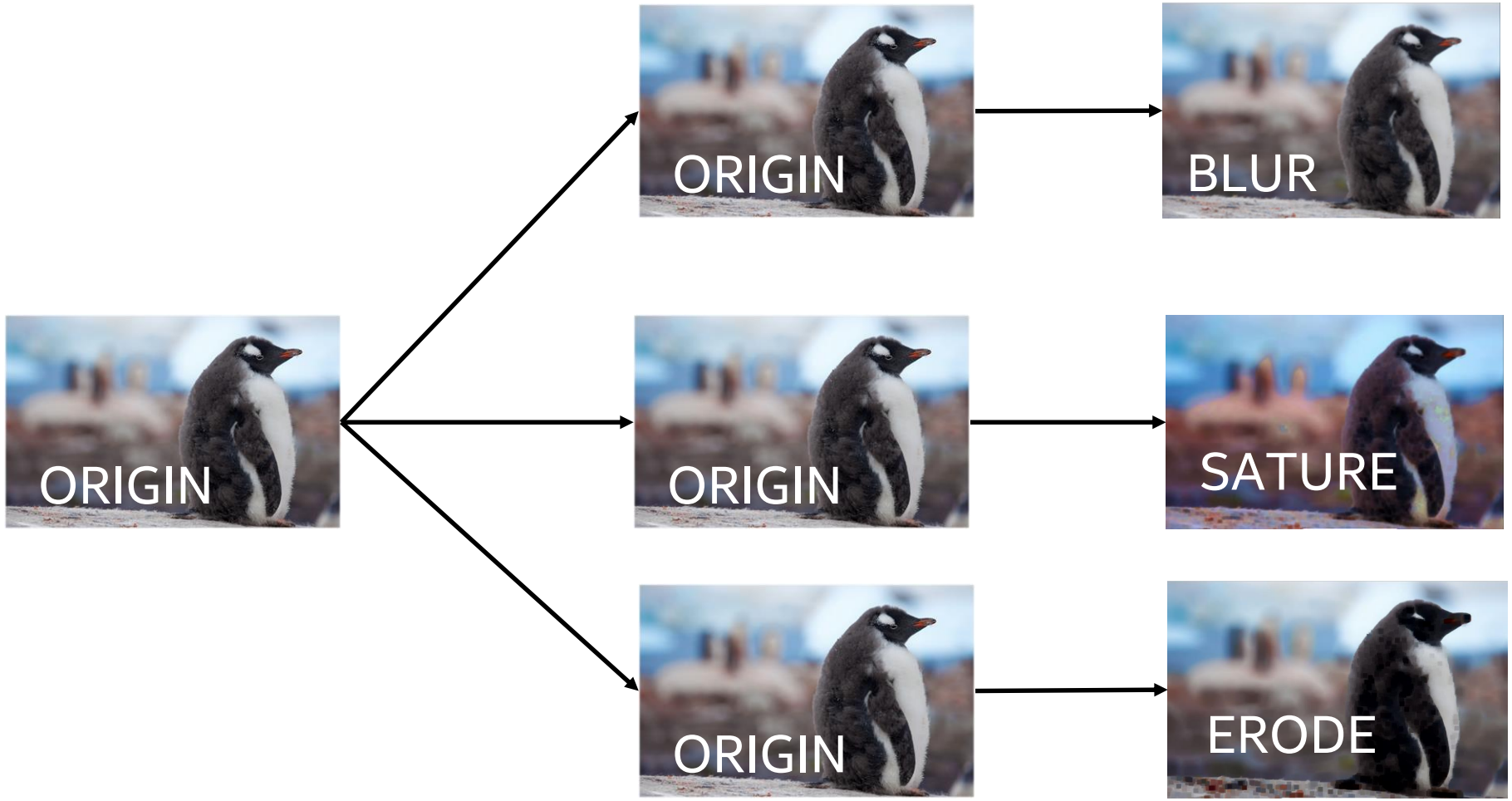
Problem statement (cont.)

Possible solution



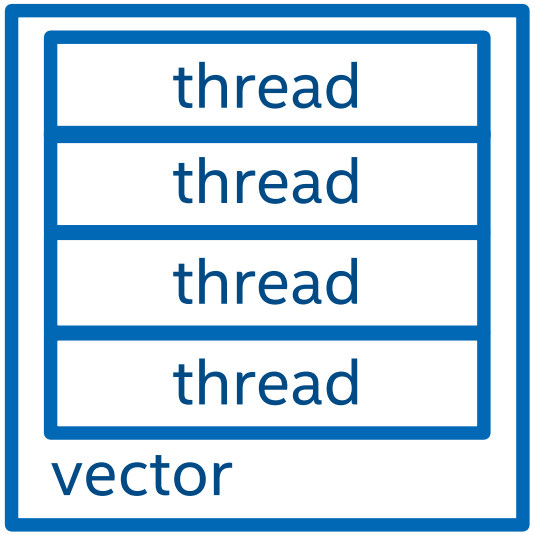
Parallel solution with oneTBB

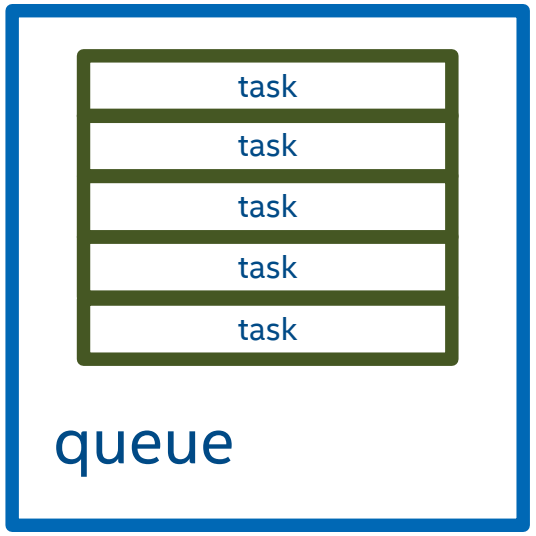
Parallel solution



Self-made task group

Self-made task group

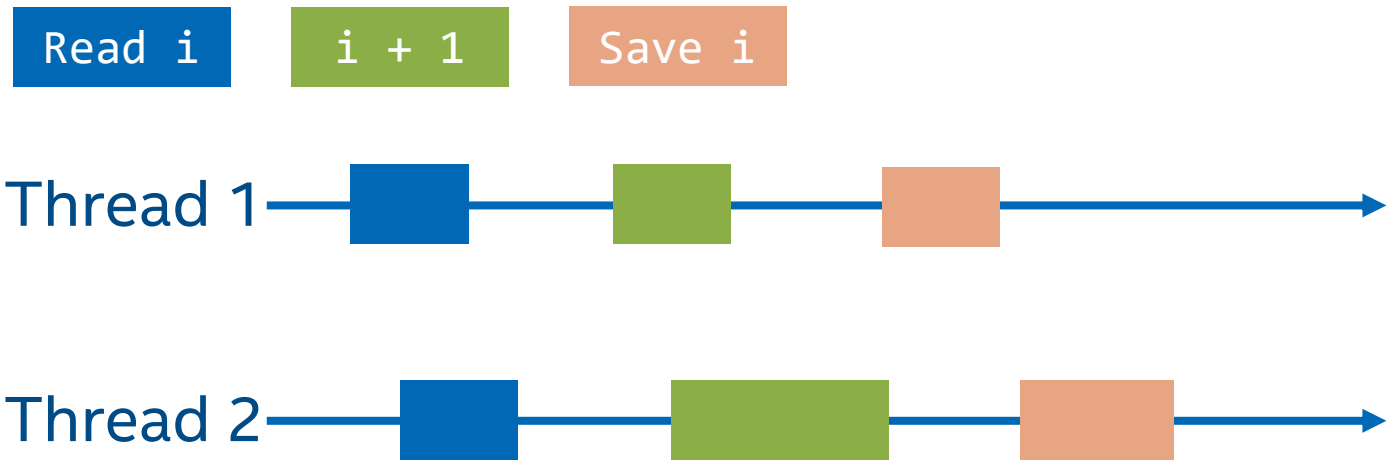




Synchronization mechanisms

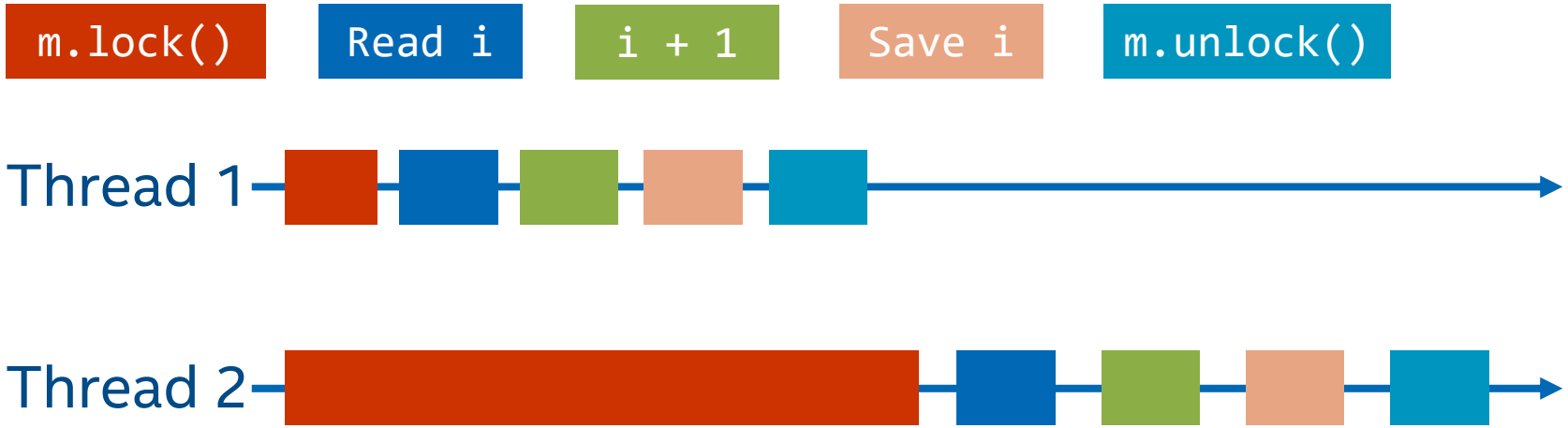
Data race example

```
int i = 0;
++i;
```



Mutex solution

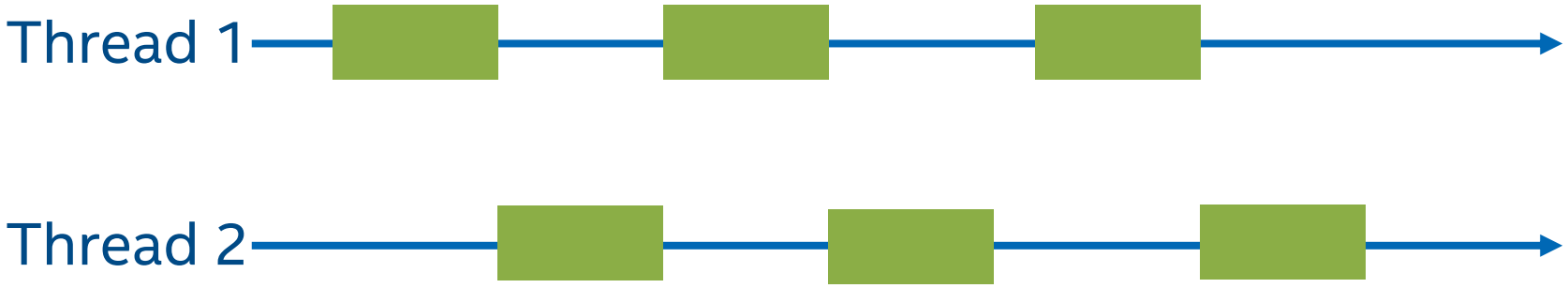
```
int i = 0;
std::mutex m;
++i;
```



Atomic operations

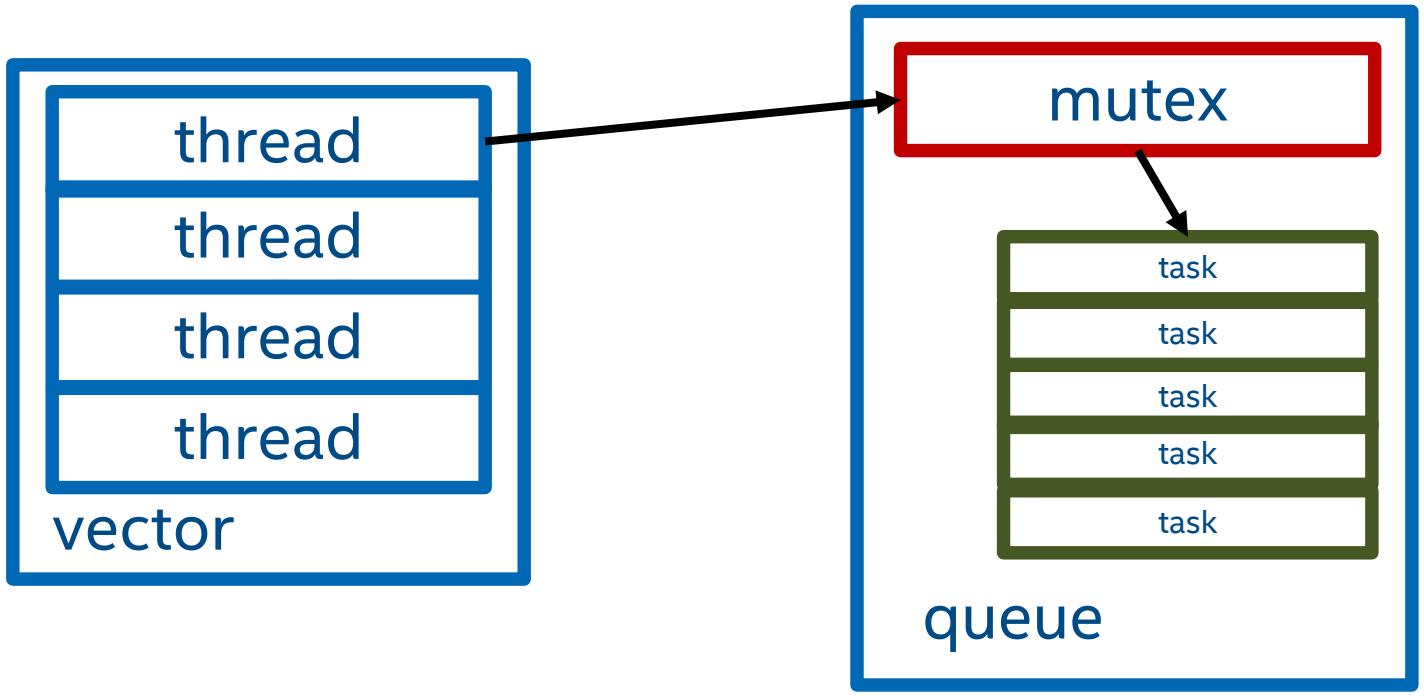
```
std::atomic<int> i(0);
```

```
++i
```

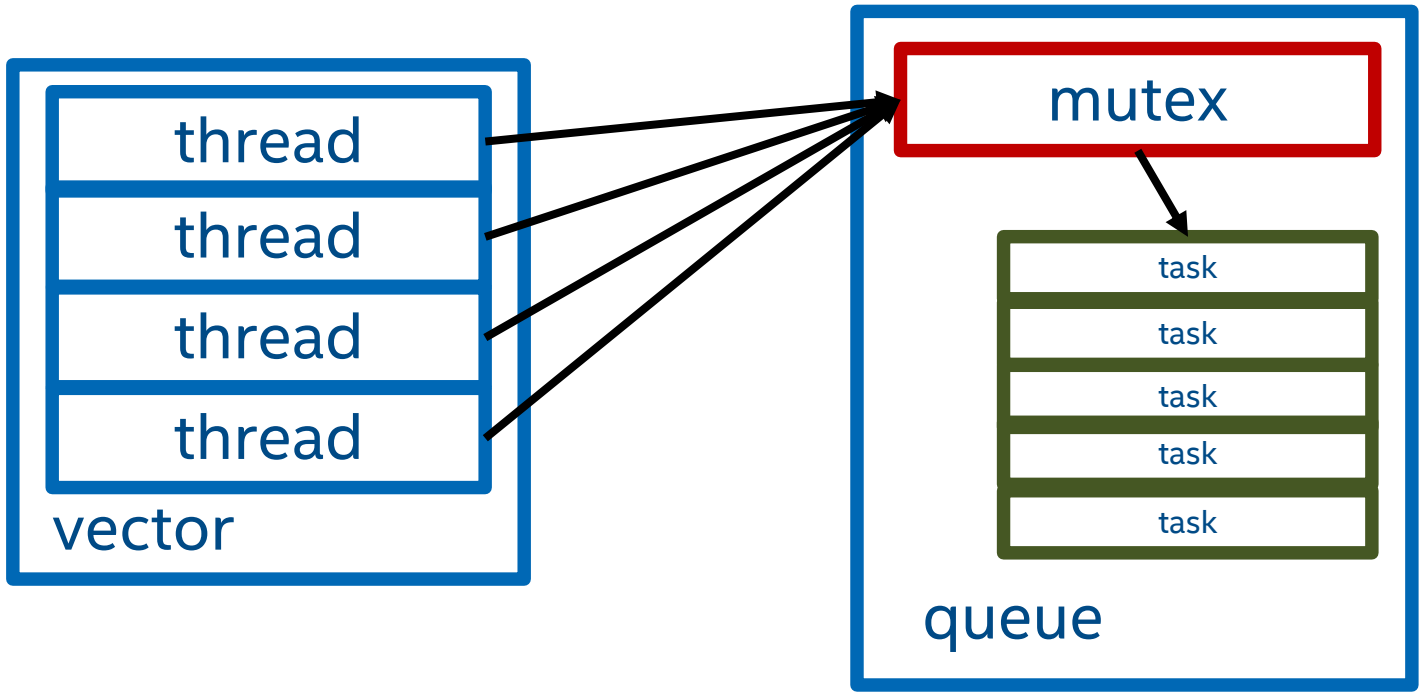


Contention

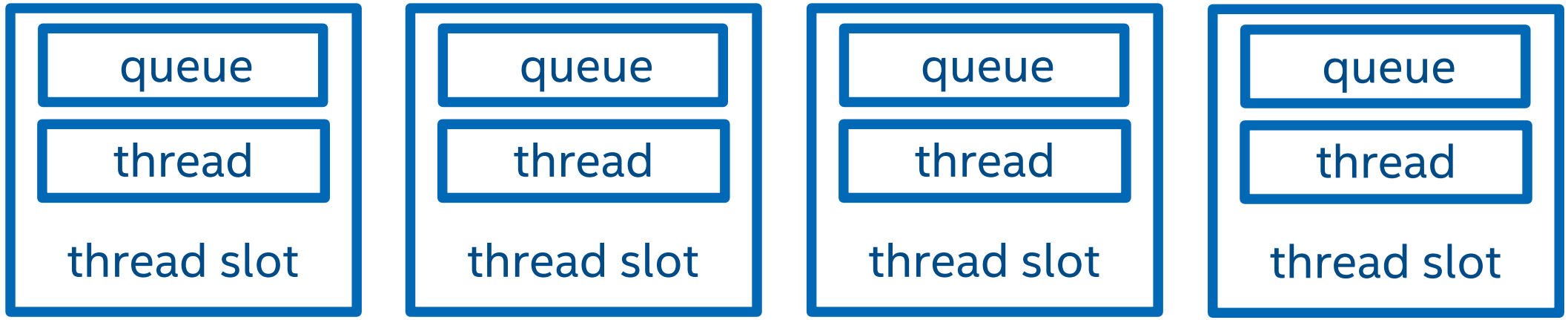
Contention on mutex



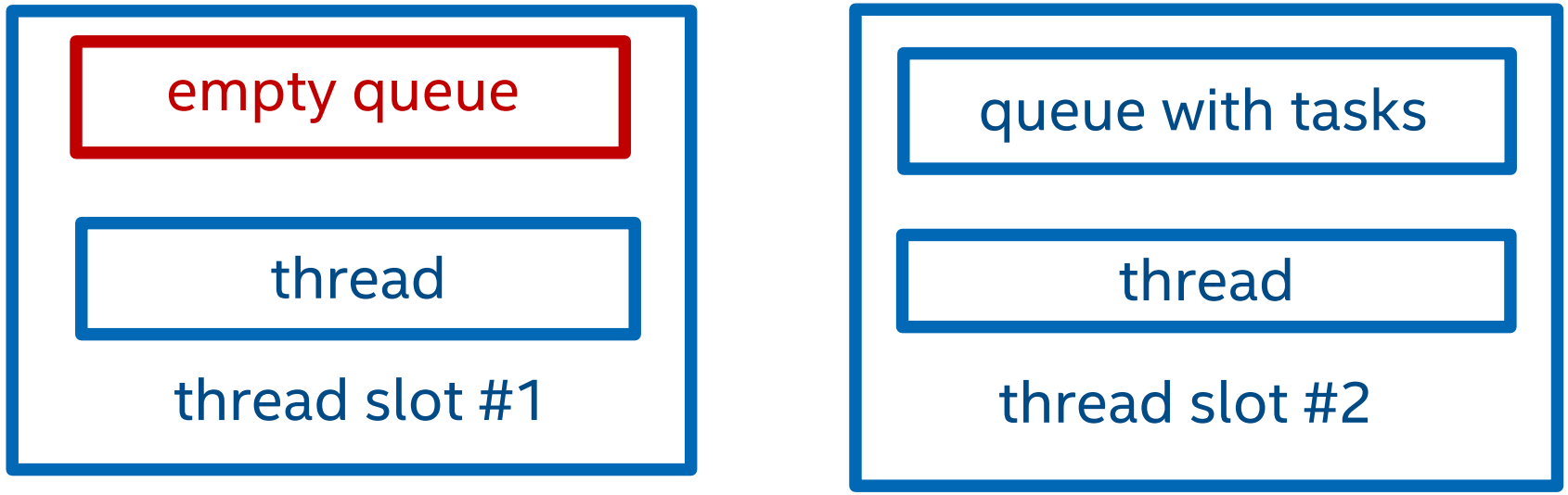
Contention on mutex (cont.)



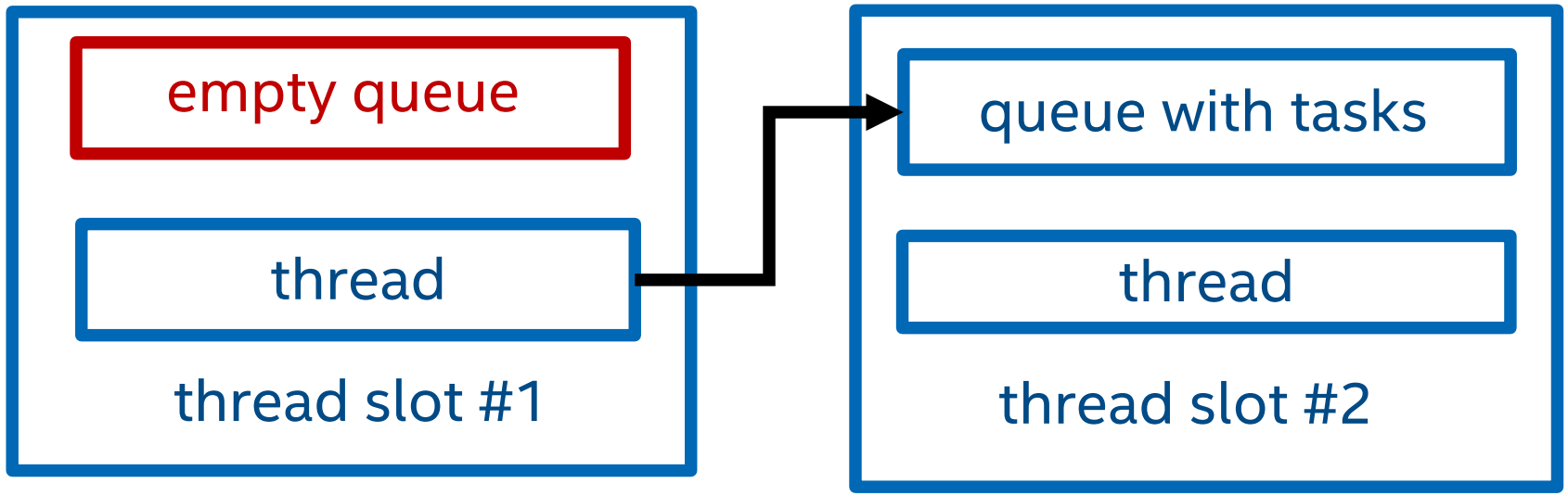
Solution: local task queue for each thread



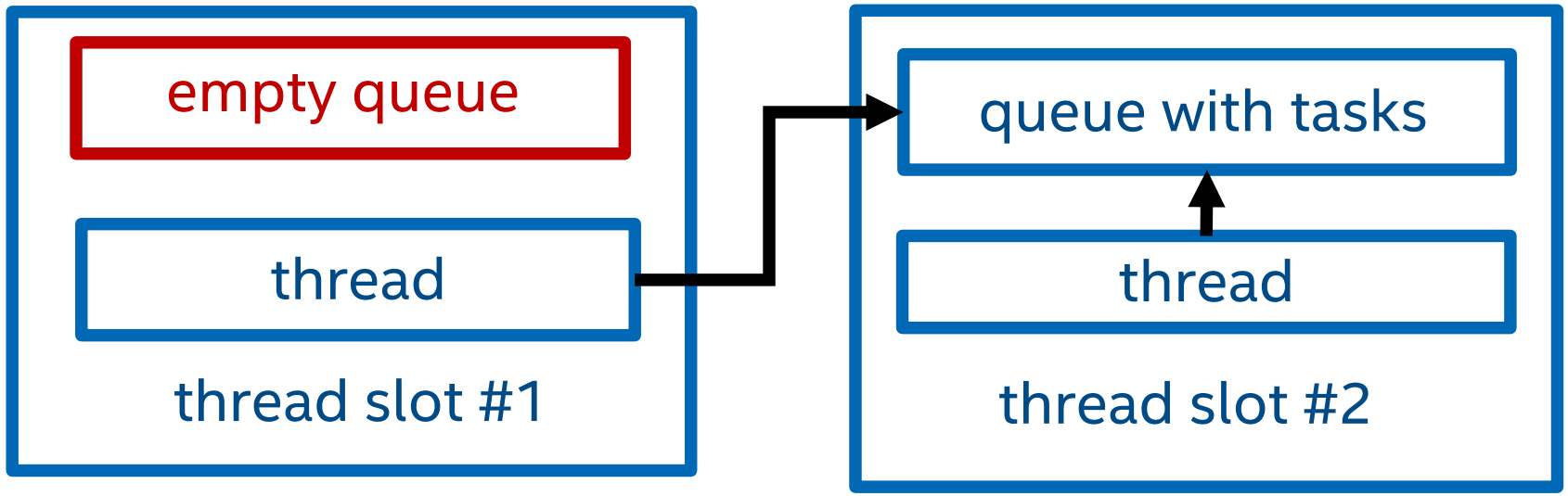
Task stealing



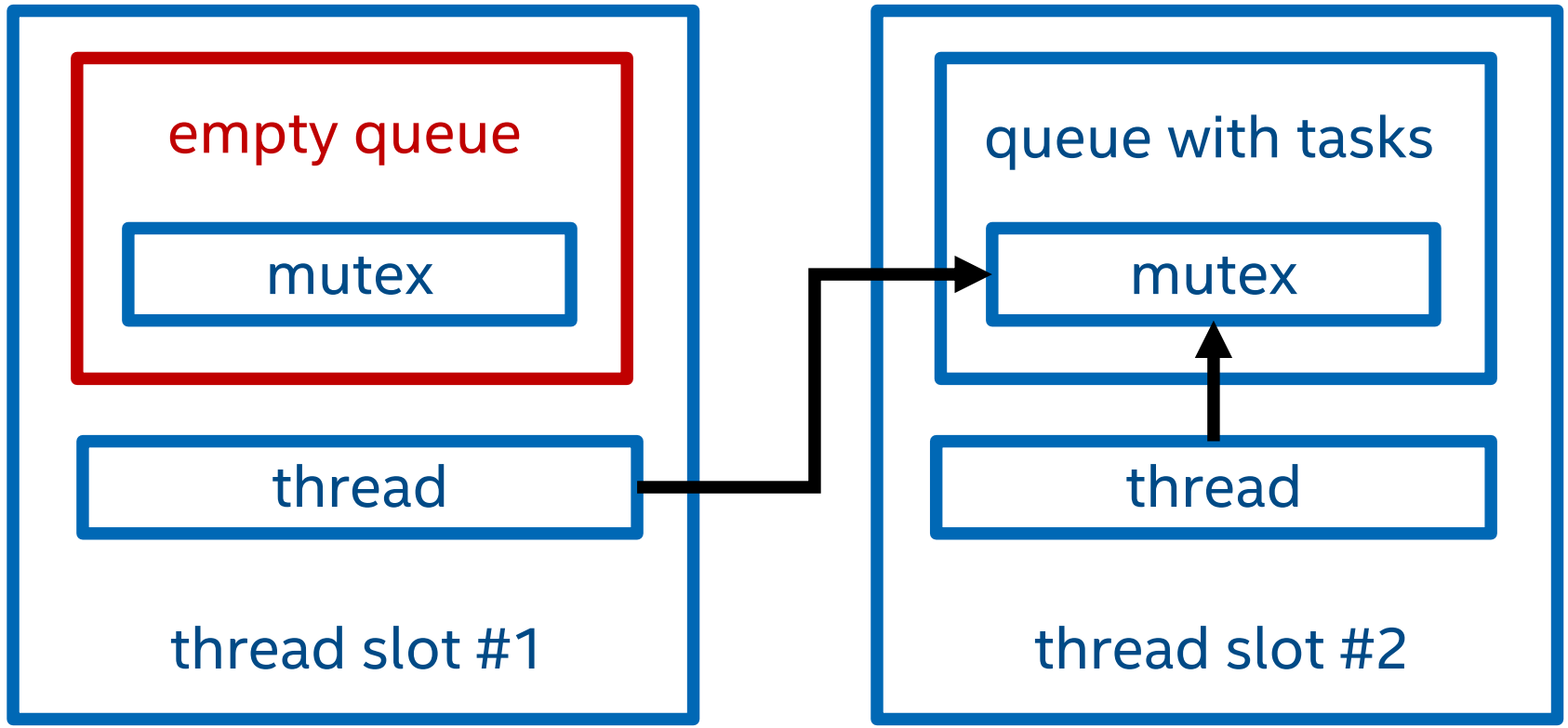
Task stealing (cont.)



Task stealing (cont.)

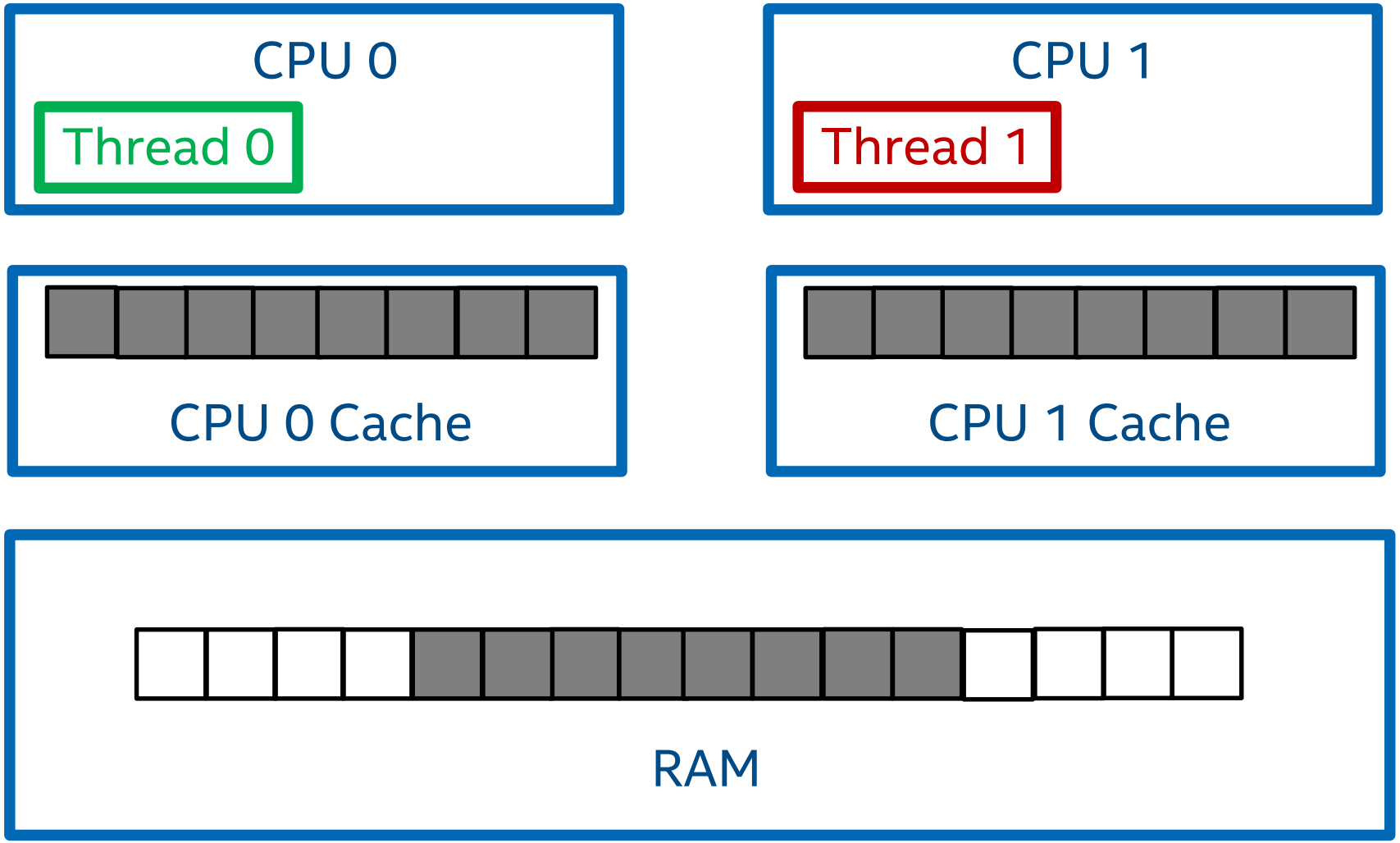


Task stealing (cont.)

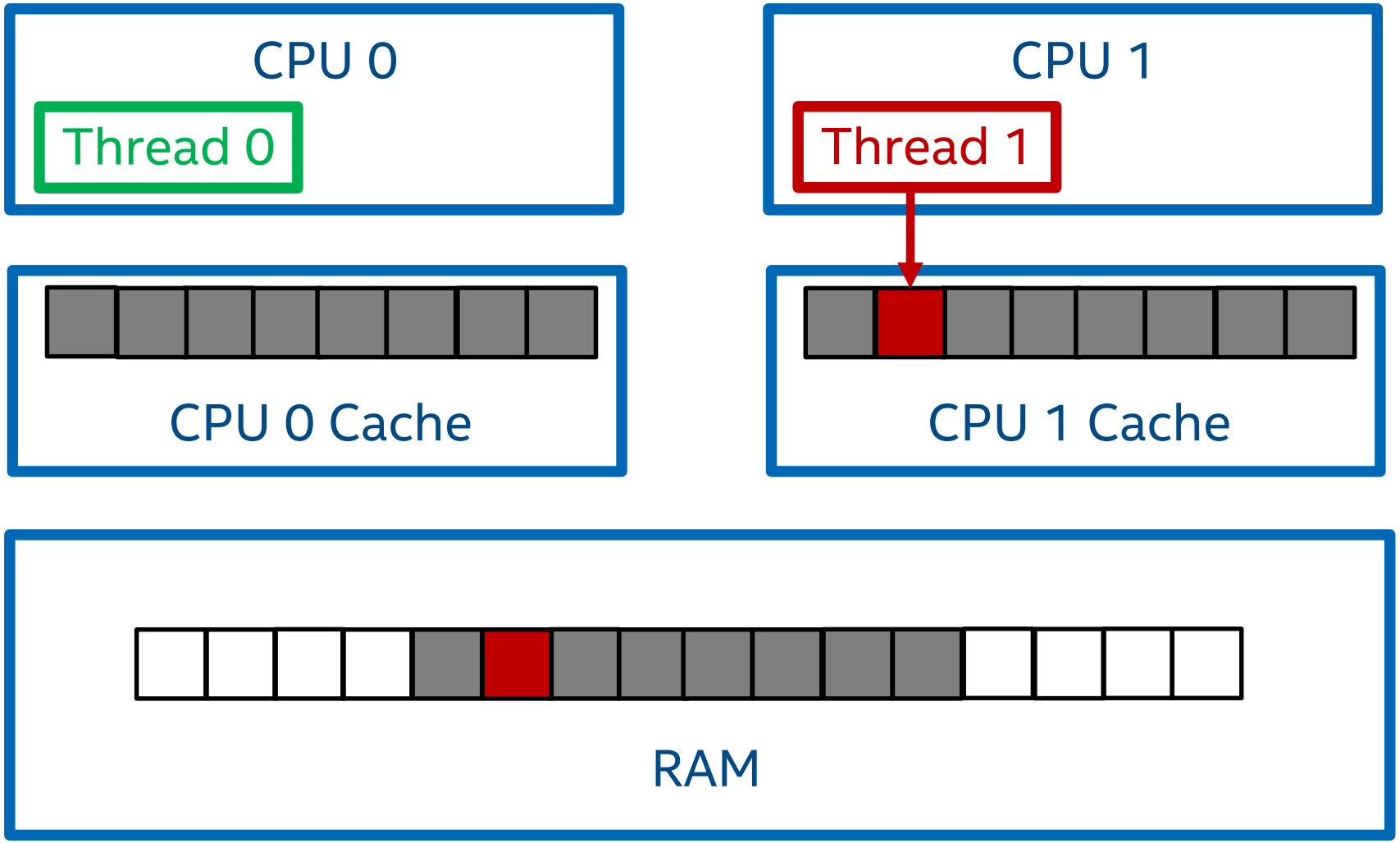


False sharing

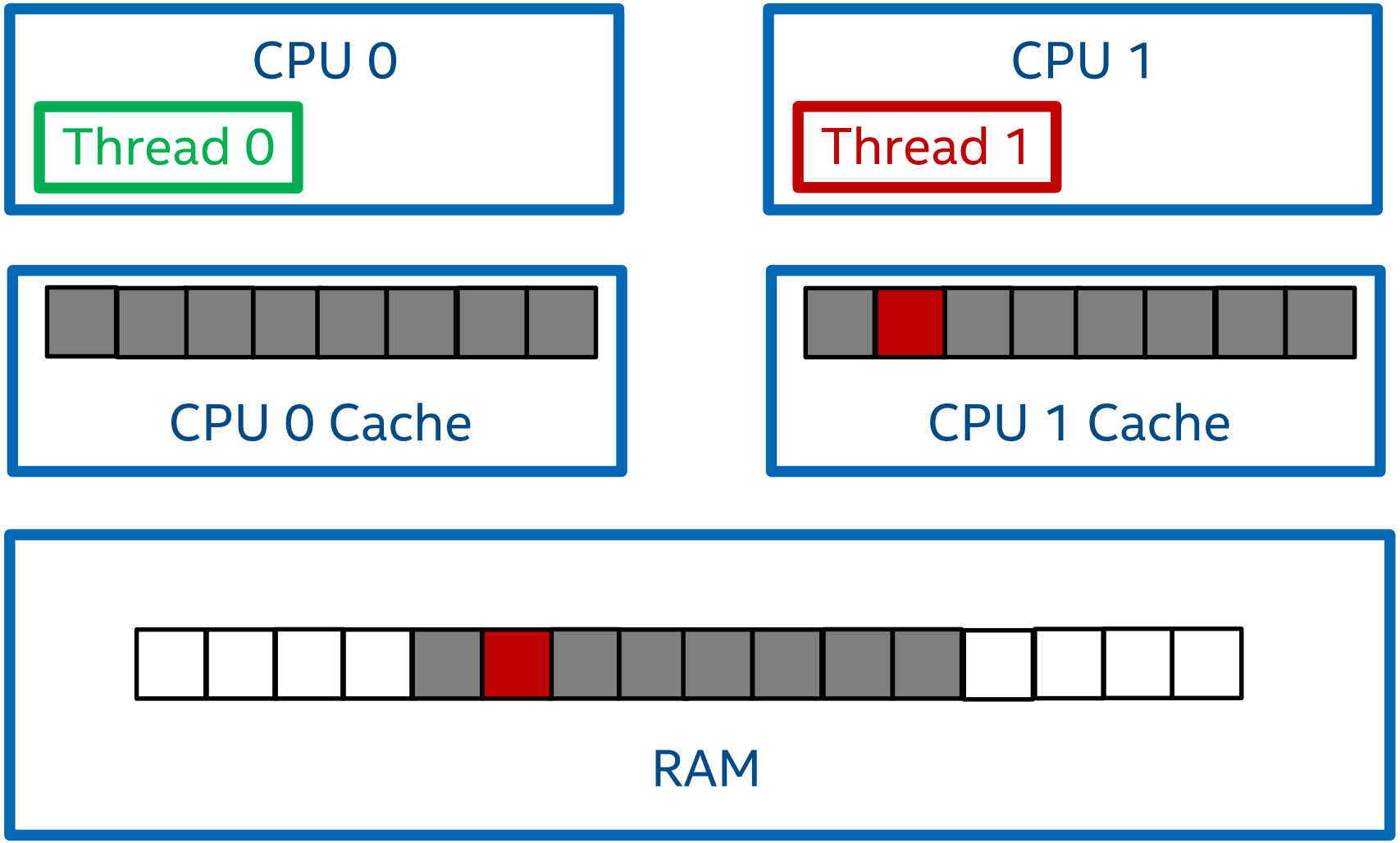
Thread slot alignment (false sharing)



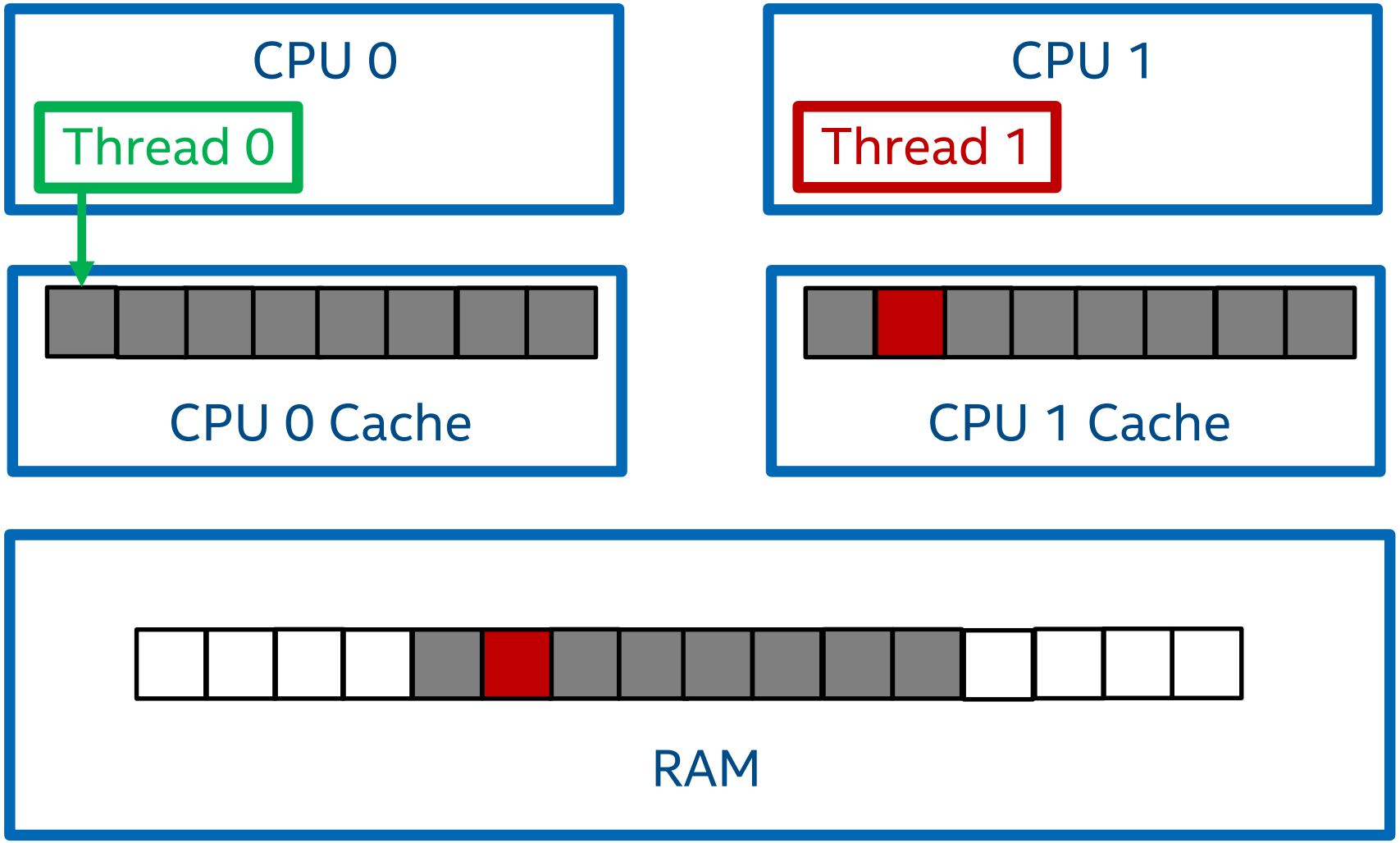
Thread slot alignment (false sharing)



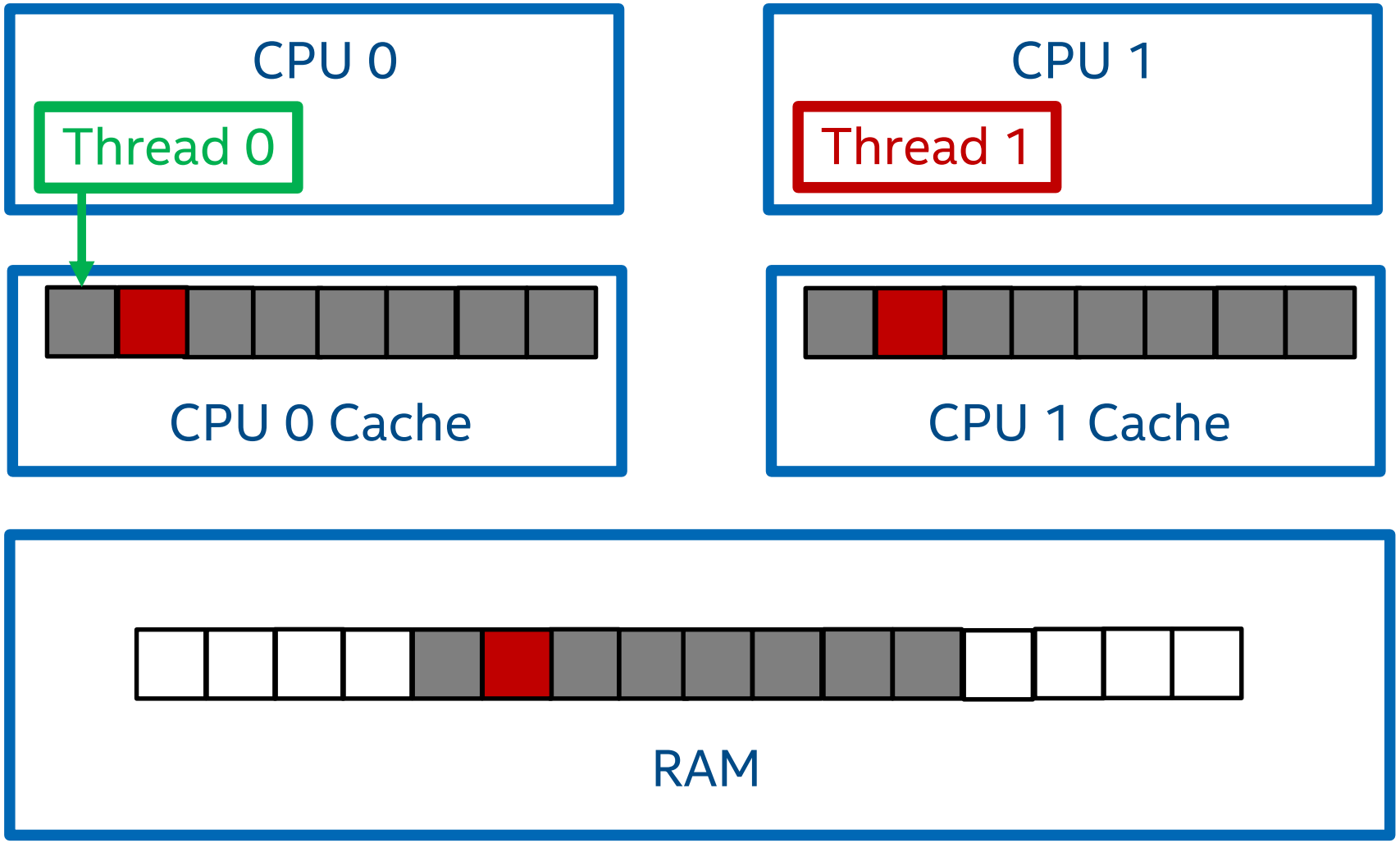
Thread slot alignment (false sharing)



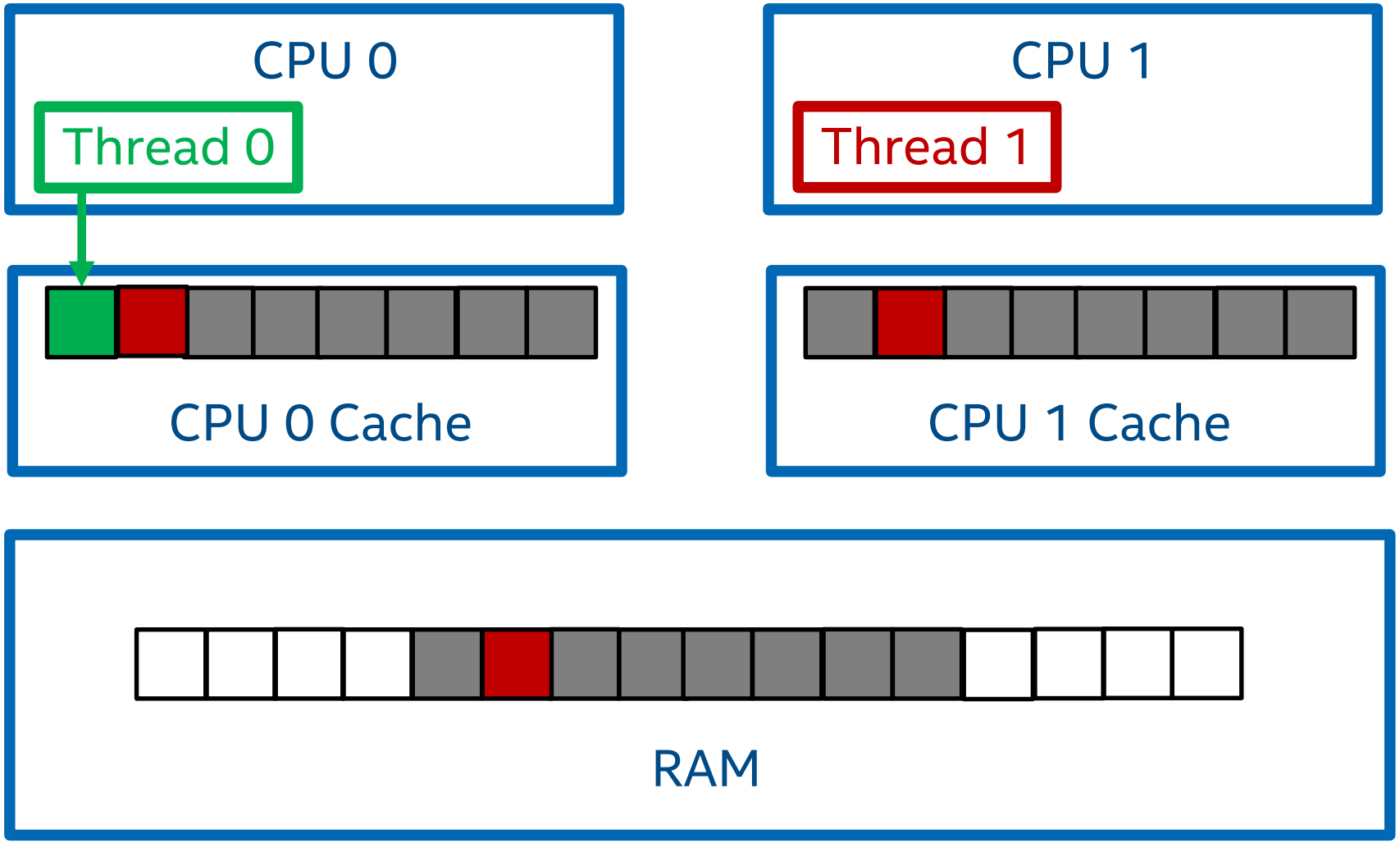
Thread slot alignment (false sharing)



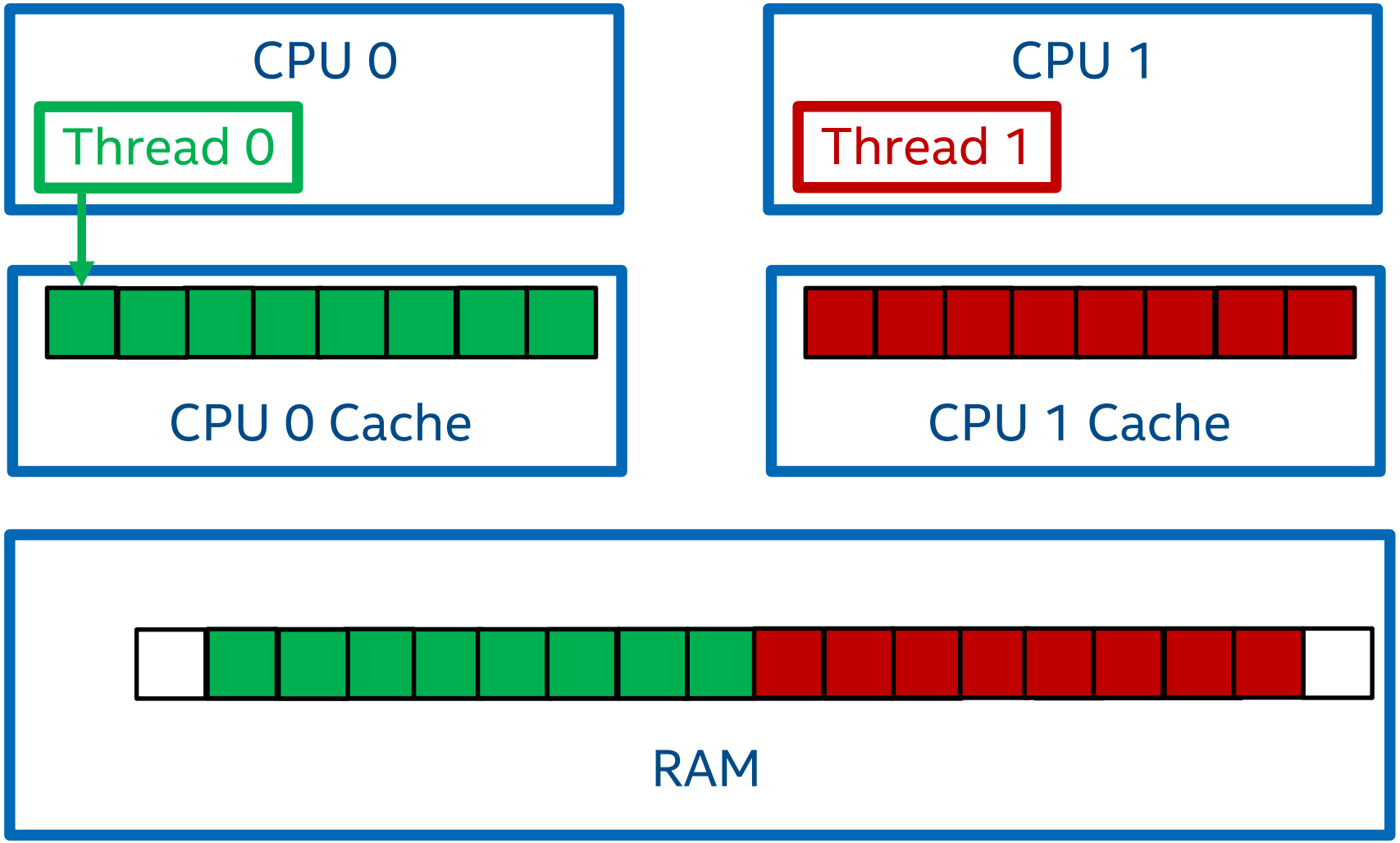
Thread slot alignment (false sharing)



Thread slot alignment (false sharing)



Thread slot alignment (false sharing)



Possible improvements

- Cache-line aligned tasks
- Condition variable synchronization