Programming OpenMP

Task Affinity

Christian Terboven

Jannis Klinkenberg

RWTH AACHEN UNIVERSITY

Improving Tasking Performance: Task Affinity

Motivation

- Techniques for process binding & thread pinning available
  - → OpenMP thread level: `OMP_PLACES & OMP_PROC_BIND`
  - → OS functionality: `taskset -c`

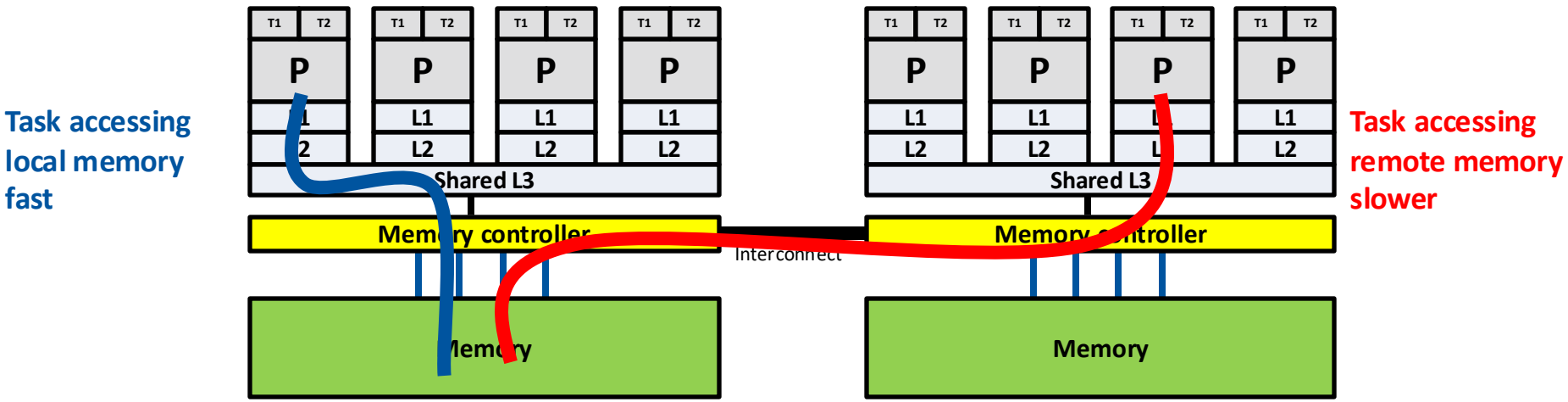
OpenMP Tasking:

- In general: Tasks may be executed by any thread in the team
  - → Missing task-to-data affinity may have detrimental effect on performance

OpenMP 5.0:

- `affinity` clause to express affinity to data

Problem / Challenge



- Might result in
  - → Unpredictable remote memory accesses & execution times
  - → High runtime variability
- Data locality crucial to sustain performance

`affinity` clause

- New clause: `#pragma omp task affinity (list)`
  - → Hint to the runtime to execute task closely to physical data location
  - → Clear separation between dependencies and affinity
- Expectations:
  - → Improve data locality / reduce remote memory accesses
  - → Decrease runtime variability
- Still expect task stealing
  - → In particular, if a thread is under-utilized

Code Example

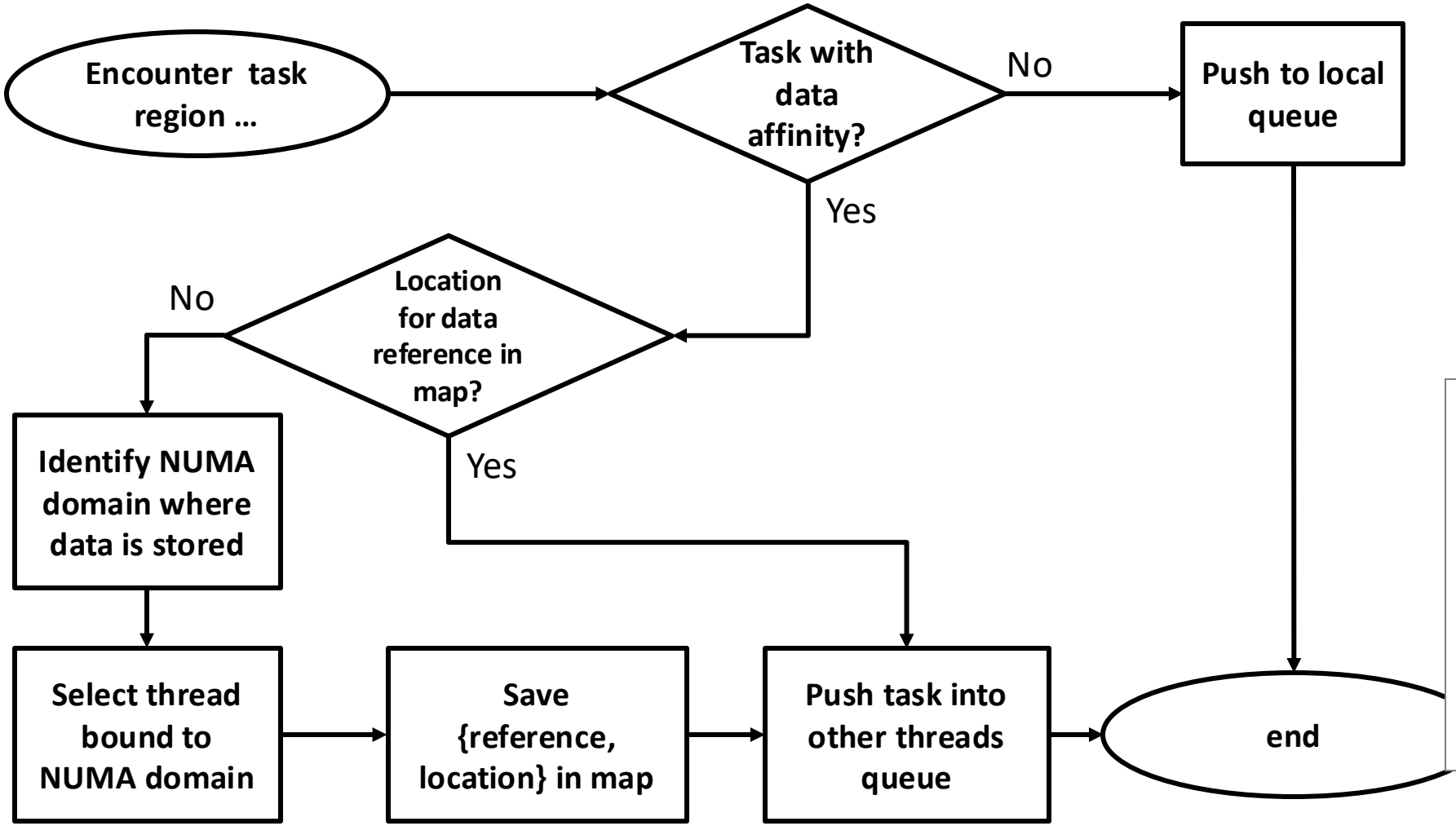
- Excerpt from task-parallel STREAM

```
1   #pragma omp task \
2       firstprivate(tmp_idx_start, tmp_idx_end) \
3       affinity( a[tmp_idx_start] )
4   {
5      int i;
6      for(i = tmp_idx_start; i <= tmp_idx_end; i++)
7           a[i] = b[i] + scalar * c[i];
8   }
```

→Loops have been blocked manually (see `tmp_idx_start/end`)

→Assumption: initialization and computation have same blocking and same affinity

Selected LLVM implementation details

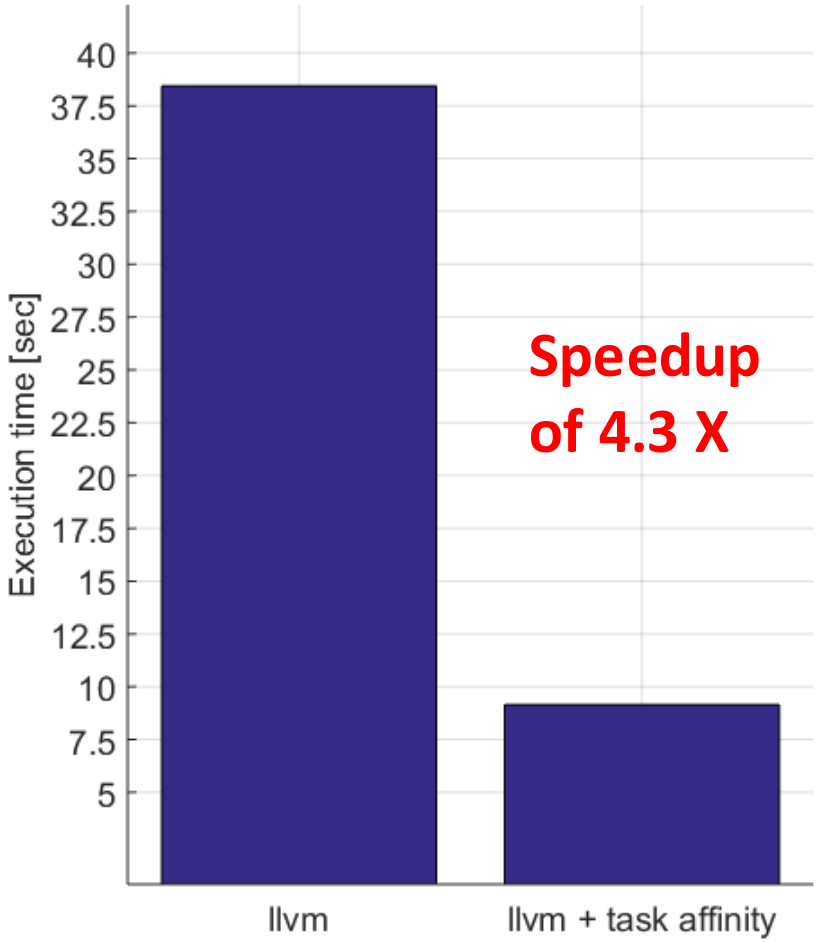


A map is introduced to store location information of data that was previously used

Jannis Klinkenberg, Philipp Samfass, Christian Terboven, Alejandro Duran, Michael Klemm, Xavier Teruel, Sergi Mateo, Stephen L. Olivier, and Matthias S. Müller. **Assessing Task-to-Data Affinity in the LLVM OpenMP Runtime**. Proceedings of the 14th International Workshop on OpenMP, IWOMP 2018. September 26-28, 2018, Barcelona, Spain.

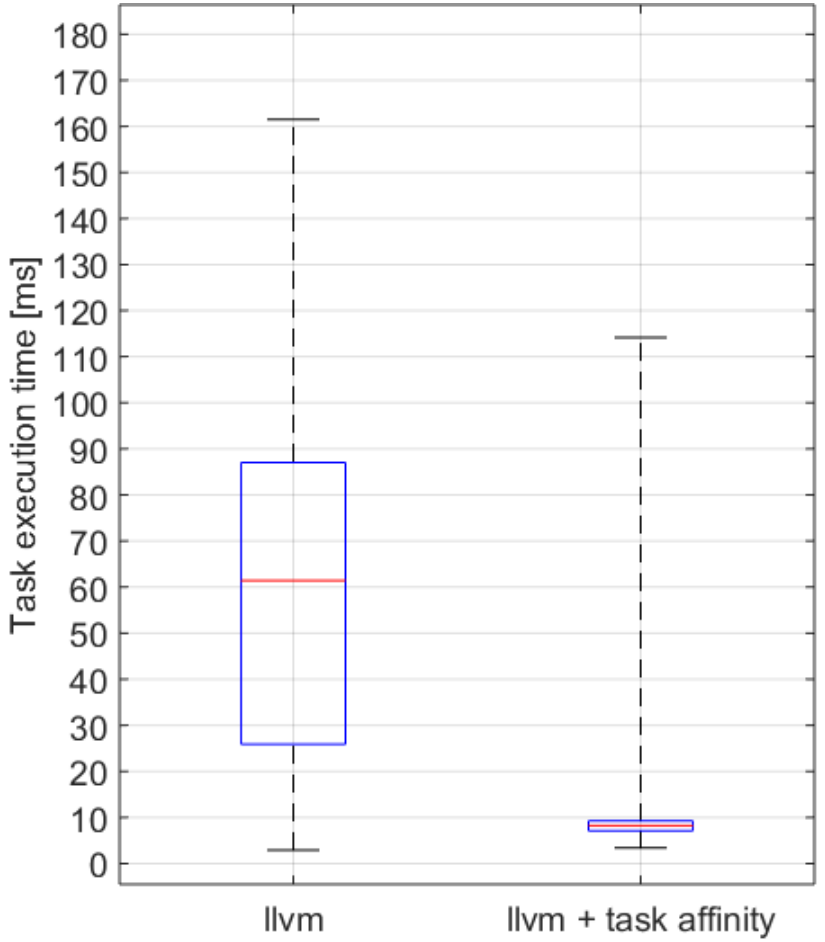
Evaluation (8-socket system)

Program runtime
Median of 10 runs



Speedup of 4.3 X

Distribution of single task execution times



LIKWID: reduction of remote data volume from 69% to 13%

Summary

- Requirement for this feature: thread affinity enabled

- The `affinity` clause helps, if
  - → tasks access data heavily
  - → single task creator scenario, or task not created with data affinity
  - → high load imbalance among the tasks

- Different from thread binding: task stealing is absolutely allowed