Otter: An OMPT Tool for Tracing and Visualising OpenMP Tasks

Adam Tuft *(MSc Scientific Computing and Data Analysis programme, Department of Computer Science, Durham University)* · **adam.s.tuft@durham.ac.uk** · **github.com/adamtuft/otter**

Introduction

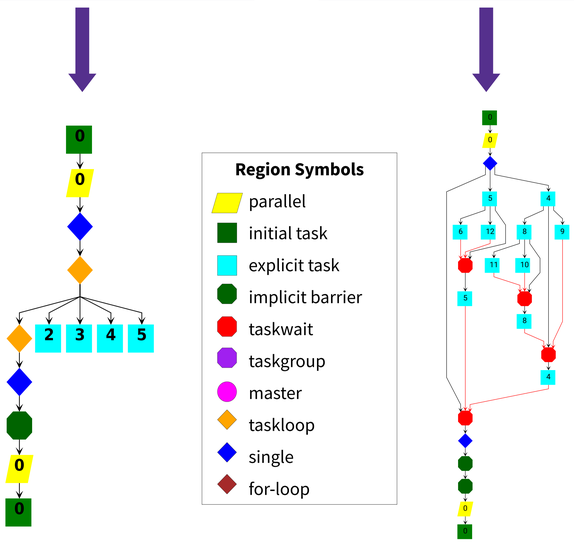
- Understanding performance of task-based code difficult due to additional concurrency of tasks & myriad scheduling possibilities.
- Thread-centric analysis tools obscure underlying task-graph structure by showing particular scheduling of tasks onto threads.
- Opportunity: portable performance analysis tool for measuring & visualising task graph structure of task-based code.
- Case study reveals the task-based structure of a PDE solver produced with ExaHyPE & illustrates performance bottlenecks identified in LLVM's OpenMP implementation.

Solution Overview

- **Otter:** event-driven, callback-based tool for tracing and visualising structure of *parallel-for*- & *task*-based OpenMP programs.
- OpenMP Tools (OMPT) interface is a non-invasive, portable alternative to direct instrumentation.
- Otter traces runtime event data from OMPT interface in OTF2 format.
- Trace data transformed into a directed acyclic graph (DAG) visualising *parallel-for*- and *task*-based structure.

```
#pragma omp parallel
{
  #pragma omp single nowait
  #pragma omp taskloop nogroup
  for (int j=0; j<4; j++)
  {
  // do work
  }
}
```

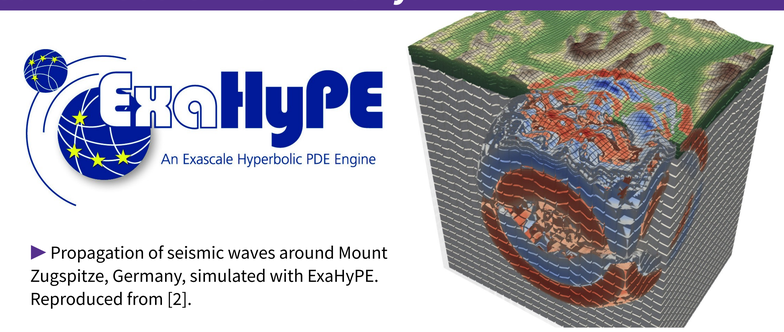
```
int fib(int n) {
  int i, j;
  if (n<2) return n;
  #pragma omp task
    i = fib(n-1);
  #pragma omp task
    j = fib(n-2);
  #pragma omp taskwait
  return i+j;
}
```



▲ OpenMP constructs are represented as nodes. Edges represent execution flow and task creation & synchronisation.

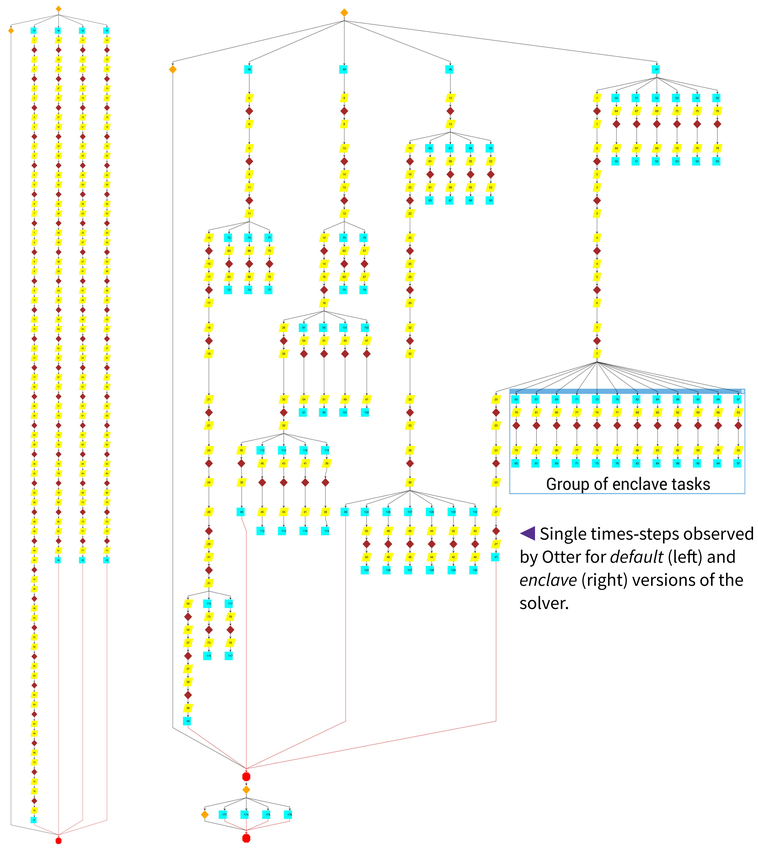
Case Study: Visualising the Task Graph of a Task-Based PDE Solver from ExaHyPE

- **ExaHyPE**: engine for solving first-order hyperbolic PDEs.
- Uses adaptive spatial grids of **Peano-4** to serialise domain cells along space-filling curve (SFC). May spawn OpenMP task on each cell.
- Case study target: Solver from [1], using **default** and **enclave** task generation modes.
- In a single time-step threads traverse partitions of the SFC to update the cells of the grid.
- **Default:** per-thread grid traversals mapped onto a set of synchronised OpenMP tasks.
- **Enclave**: non-critical cell updates packaged in *enclave* tasks to allow overlap with communication & reduce time-to-solution.



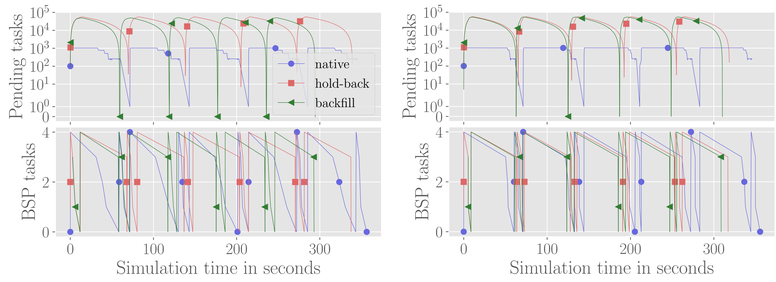
▶ Propagation of seismic waves around Mount Zugspitze, Germany, simulated with ExaHyPE. Reproduced from [2].

Results



◀ Single times-steps observed by Otter for *default* (left) and *enclave* (right) versions of the solver.

- Single time-steps observed in **default** and **enclave** tasking modes shown above.
- **Default** (left): cells updated sequentially in *parallel-for* blocks during synchronised domain traversal tasks.
- **Enclave** (right): synchronised tasks spawn unsynchronised non-critical enclave tasks which may be overlapped with later communication phase (not shown).
- Both time-steps show 81 parallel-for regions corresponding to cells of 9x9 grid, with graph structure determined by tasking mode.
- Otter illustrates inefficiencies of LLVM OpenMP implementation observed in [1] – native scheduler not able to take advantage of concurrency exposed by enclave tasks.



▲ Tasking inefficiencies observed in LLVM OpenMP implementation for ill-balanced (left) and well-balanced (right) loads. Native task limit and greedy consumption of ready tasks negates intended benefit of enclave tasks. Reproduced from [1].

Limitations & Future Work

Limitations:

- Insensitive to non-OpenMP events.
- Doesn't support `depend` clause or `distribute` & `workshare` constructs.
- Can't attribute nodes to target source.
- No means of filtering events.

Focus of future work:

- API & analysis workflow for data-driven taskification of serial code.
- Quantitative performance measurements.
- Support for other tasking runtimes e.g. Intel oneAPI toolchain.

Acknowledgements

The ExCALIBUR programme is supported by the UKRI Strategic Priorities Fund. The programme is led by the Met Office and the Engineering and Physical Sciences Research Council (EPSRC) along with the Public Sector Research Establishment, the UK Atomic Energy Authority (UKAEA) and UK Research and Innovation (UKRI) research councils, including the Natural Environment Research Council (NERC), the Medical Research Council (MRC) and the Science and Technologies Facilities Council (STFC).

This work is supported by ExCALIBUR's cross-cutting tasking theme (grant ESA 10 CDEL).

References

[1] Schulz, Holger, et al. (2021) *Task inefficiency patterns for a wave equation solver.* arXiv preprint arXiv:2105.12739

[2] Reinarz, Anne, et al. (2020) *ExaHyPE: An engine for parallel dynamically adaptive simulations of wave problems*. Computer Physics Communications 254