OPTIMISING PARALLEL PROGRAMS ON XEON PHI

Adrian Jackson

adrianj@epcc.ed.ac.uk

@adrianjhpc

epcc

THE UNIVERSITY OF EDINBURGH

Specialised Optimisations

- Some optimisation are specific to Xeon Phi only
  - Offloading
  - MPI performance
  - Thread and process placement
  - Filesystems

Offload memory

- By default memory allocated for all data before offload and deallocated on completion of offload
- Can use offload_transfer directive to explicitly manage data

```
#pragma offload_transfer target(mic:1) in(a)
!dir$ offload_transfer target(mic:1) in(a)
```

- Can specify allocation and free status for device memory

```
!dir$ offload target(mic:0)  in(p : alloc_if(.true.) free_if(.false.))
#pragma offload target(mic) out(p : alloc_if(1) free_if(0))
```

  - Can be combined with length attribute (`length(0)` would specify no transfer)
- Also possible to send data asynchronously using `signal` and `wait` attributes/directives
- Can get information on data transfer

```
export OFFLOAD_REPORT=2
```

epcc | THE UNIVERSITY OF EDINBURGH

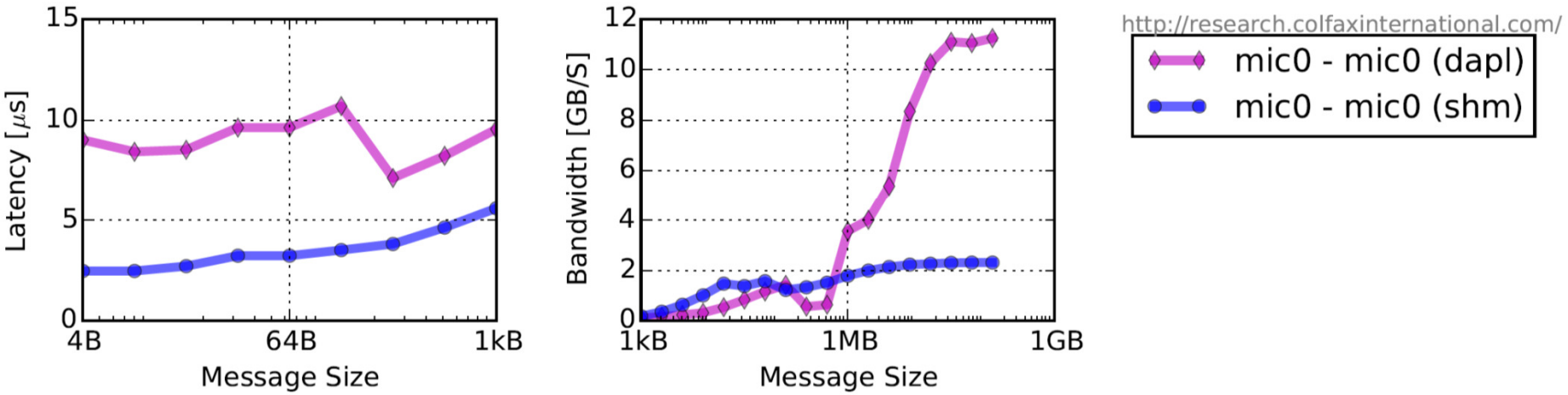
MPI fabric choice

- Intel MPI can choose different mechanisms for sending data:
  - shm: Shared-memory
  - dapl: DAPL-capable network fabric (Infiniband etc…)
  - ofa: OFA-capable network fabric (Infiniband etc…)
  - tcp: TCP/IP-capable network fabrics (Ethernet etc…)
- Can specify what fabric to use:

```
export I_MPI_FABRICS=shm:dapl
```

MPI fabric choice

- By default inside single Phi:
  - If dapl is installed (or infiniband card installed)
  - shm:dapl



- May be beneficial in some circumstances to select a specific one

epcc | THE UNIVERSITY OF EDINBURGH

Thread placement

- `KMP_AFFINITY` variable controls thread placement

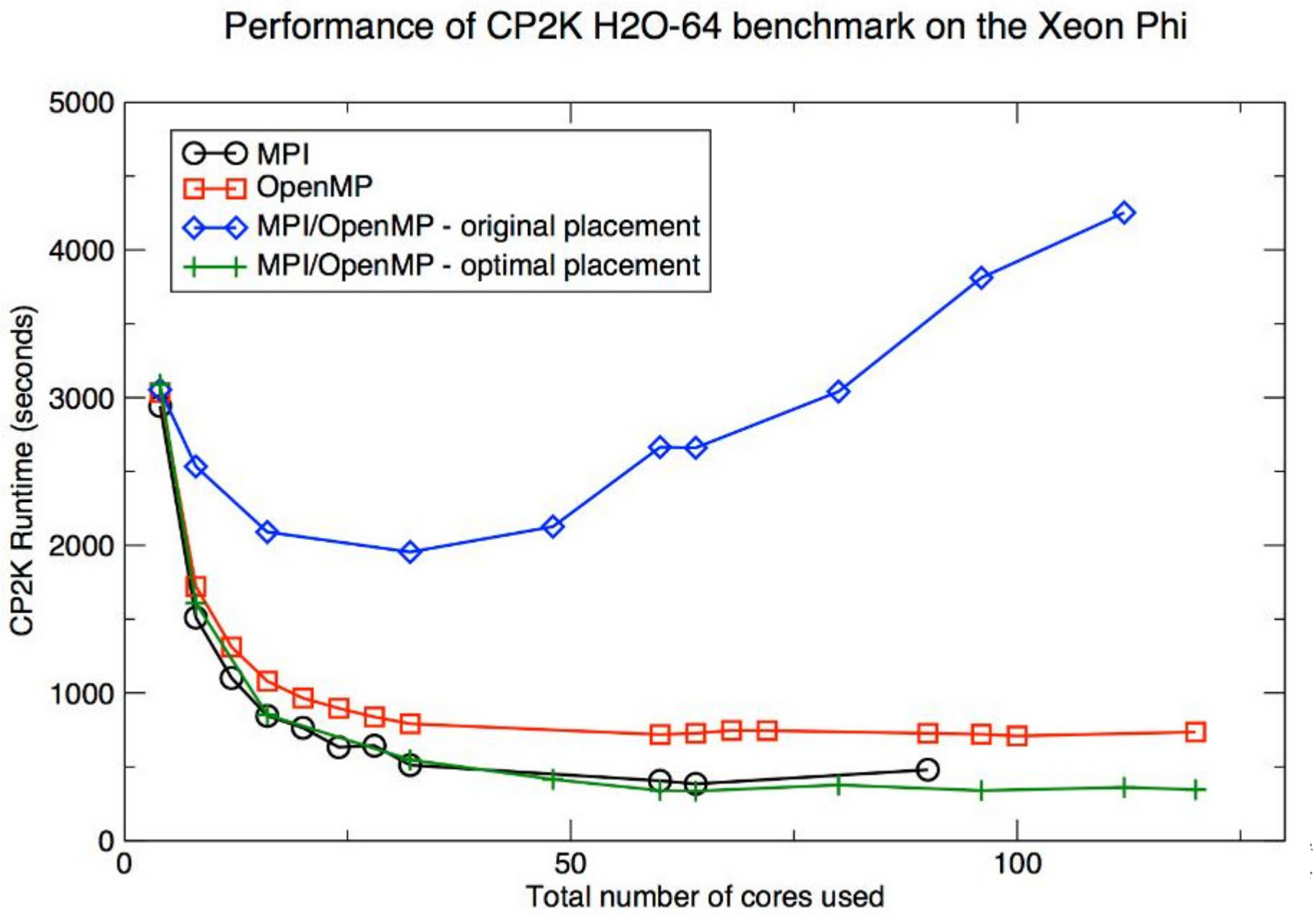
  `export KMP_AFFINITY=`[attribute]

- Attribute can be:
  - `compact`, `scatter`, `balanced`, or `explicit`
  - Can specify `granularity` as well
    - `fine`, `thread`, and `core` (default)

```
export KMP_AFFINITY=compact,granularity=fine
export KMP_AFFINITY=scatter
```

- Compute bound application:
  - compact (2 or more threads per core)
- Bandwidth-bound application:
  - scatter (1 thread per core)

epcc | THE UNIVERSITY OF EDINBURGH


Performance of CP2K H2O-64 benchmark on the Xeon Phi
MPI
OpenMP
MPI/OpenMP - original placement
MPI/OpenMP - optimal placement
CP2K Runtime (seconds)
Total number of cores used
0
1000
2000
3000
4000
5000
0
50
100

File systems

- RAM file system
  - Stored in memory
  - Fastest
  - Volatile
- Local host drives
  - Mount disk from host on Xeon Phi
  - Persistent, not as fast as RAM file system
- Network storage
  - Gives access to larger data systems
  - Even slower

Conclusions

- Setup of hardware and software on Phi can make performance difference
  - Communication hardware or libraries
  - Filesystems
- Placement of threads critical for performance
- If offloading, looking at data persistence is a good optimization option