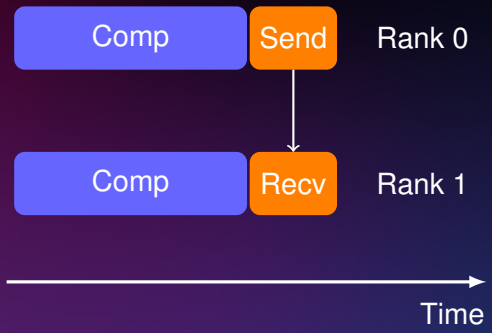
Towards Efficient Oversubscription: On the Cost and Benefit of Event-Based Communication in MPI

Jan Bierbaum, Maksym Planeta, Hermann Härtig

ROSS, 2022-11-13

Classical MPI Programming

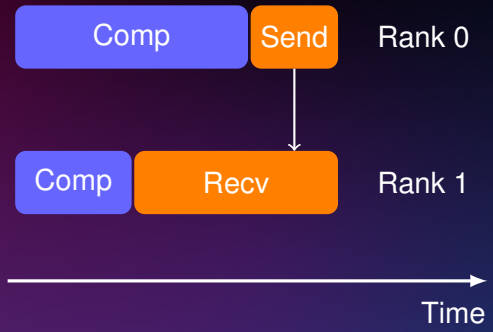
- Bulk synchronous parallel
- One rank/process per CPU core
- OS-bypass communication
- Polling for completion



Classical MPI Programming

- Bulk synchronous parallel
- One rank/process per CPU core
- OS-bypass communication
- Polling for completion

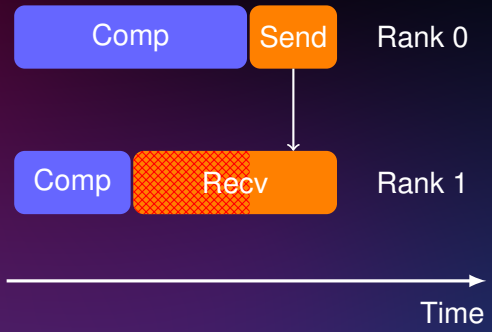
Imbalance



Classical MPI Programming

- Bulk synchronous parallel
- One rank/process per CPU core
- OS-bypass communication
- Polling for completion

Imbalance → cycles/energy wasted



Countermeasures

- Load balancing

Countermeasures

- Load balancing (effort, complexity)

Countermeasures

- Load balancing (effort, complexity)
- Interleaving of computation & communication

Countermeasures

- Load balancing (effort, complexity)
- Interleaving of computation & communication (effort, complexity)

Countermeasures

- Load balancing (effort, complexity)
- Interleaving of computation & communication (effort, complexity)
- Higher-level runtimes or MPI extensions

Countermeasures

- Load balancing (effort, complexity)
- Interleaving of computation & communication (effort, complexity)
- Higher-level runtimes or MPI extensions (effort, expertise)

Countermeasures

- Load balancing (effort, complexity)
- Interleaving of computation & communication (effort, complexity)
- Higher-level runtimes or MPI extensions (effort, expertise)
- Oversubscription

Countermeasures

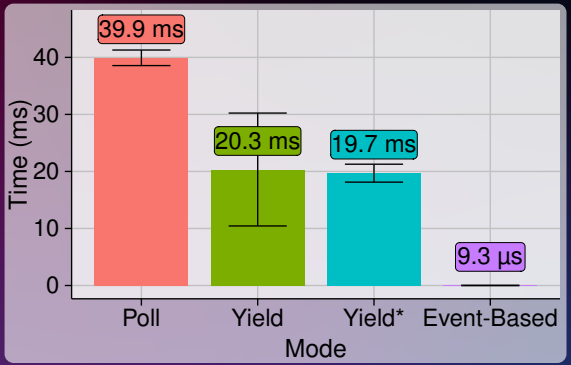
- Load balancing (effort, complexity)
- Interleaving of computation & communication (effort, complexity)
- Higher-level runtimes or MPI extensions (effort, expertise)
- Oversubscription (efficient implementation?)

Oversubscription

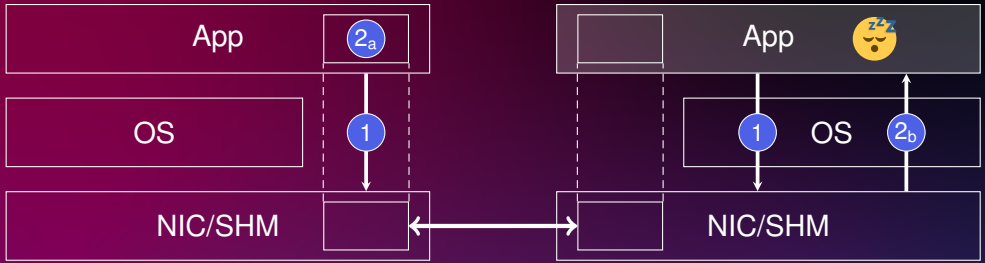
- Ping-pong micro benchmark
- 2 ranks sharing 1 CPU

Oversubscription

- Ping-pong micro benchmark
- 2 ranks sharing 1 CPU
- Polling $\rightarrow$ massive overhead
- Yield = `sched_yield`
- Yield* = “legacy” variant of `sched_yield`

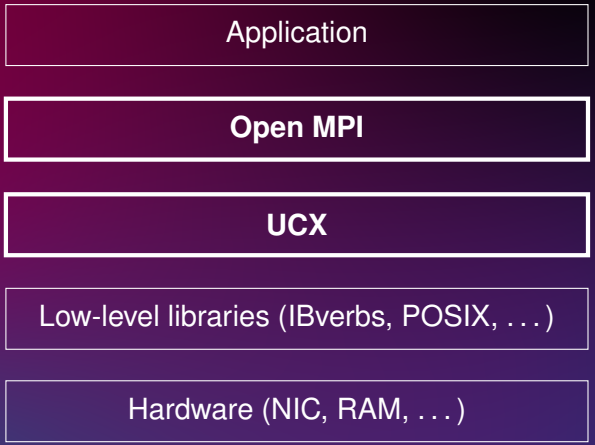


Event-Based Communication



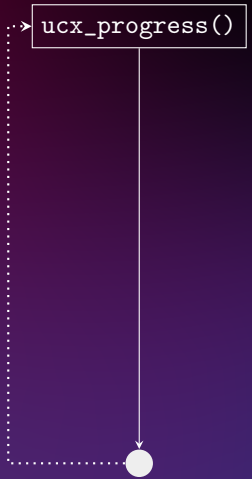
1. Application/OS sets up communication operation
2a. Application polls memory for completion
2b. OS resumes application on completion

Open MPI & UCX


Application
Open MPI
UCX
Low-level libraries (IBverbs, POSIX, . . .)
Hardware (NIC, RAM, . . .)

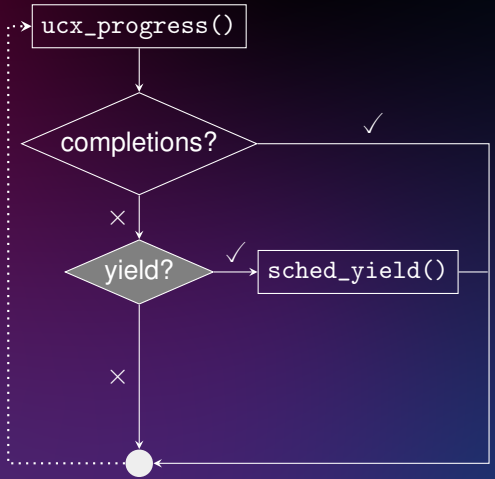
Event-Based Communication in Open MPI

- UCX as standard backend for InfiniBand



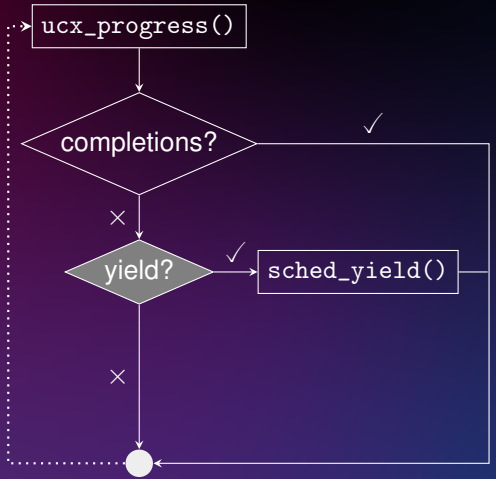
Event-Based Communication in Open MPI

- UCX as standard backend for InfiniBand
- Open MPI uses `sched_yield` when oversubscribed



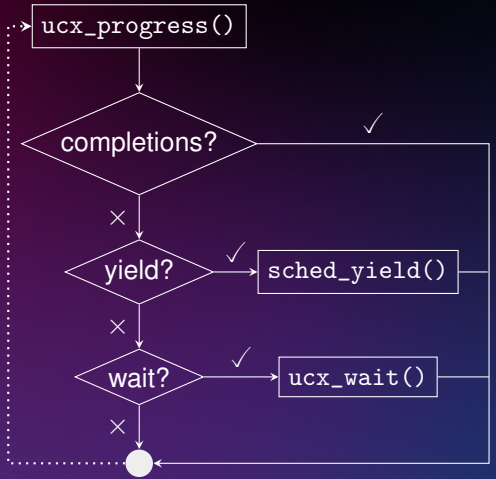
Event-Based Communication in Open MPI

- UCX as standard backend for InfiniBand
- Open MPI uses `sched_yield` when oversubscribed
- UCX backend supports event-based communication



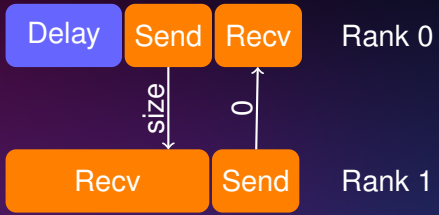
Event-Based Communication in Open MPI

- UCX as standard backend for InfiniBand
- Open MPI uses `sched_yield` when oversubscribed
- UCX backend supports event-based communication
- Extension to Open MPI for *adaptive waiting*



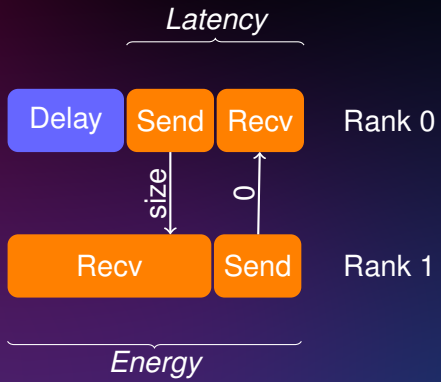
LIBRA, an MPI Micro-Benchmark

- P2P ping-pong using blocking MPI operations
- Configurable sender delay and message size



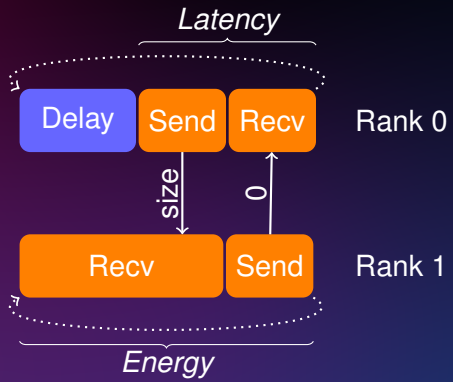
LIBRA, an MPI Micro-Benchmark

- P2P ping-pong using blocking MPI operations
- Configurable sender delay and message size
- Measure communication latency and overall energy consumption



LIBRA, an MPI Micro-Benchmark

- P2P ping-pong using blocking MPI operations
- Configurable sender delay and message size
- Measure communication latency and overall energy consumption



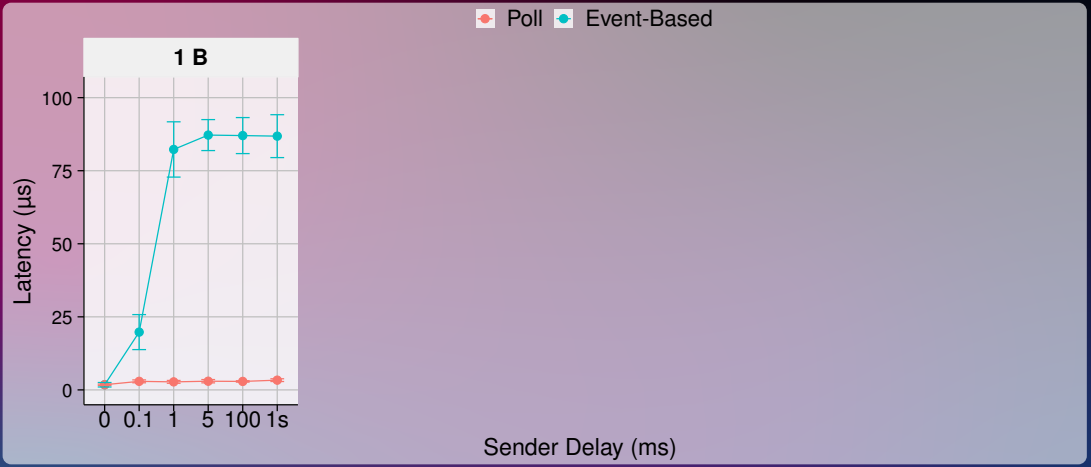
Evaluation: Setup

- High Performance Computing and Storage Complex (“Taurus”) at TU Dresden:
  - 2×12-core Intel Xeon E5-2680 v3 @ 2.50 GHz
  - Mellanox Connect-IB
  - “High Definition Energy Efficiency Monitoring” (HDEEM)
  - Exclusively allocated node

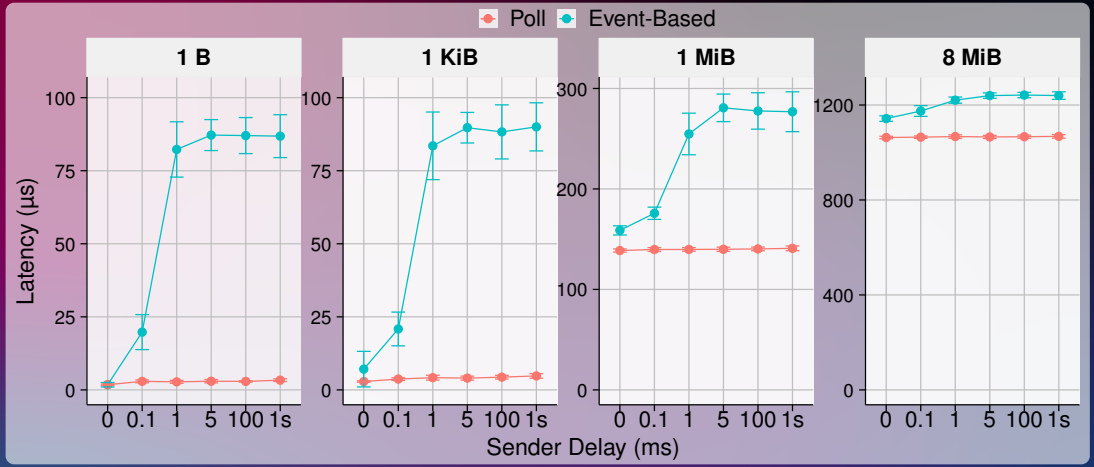
Evaluation: Setup

- High Performance Computing and Storage Complex (“Taurus”) at TU Dresden:
  - $2 \times$ 12-core Intel Xeon E5-2680 v3 @ 2.50 GHz
  - Mellanox Connect-IB
  - “High Definition Energy Efficiency Monitoring” (HDEEM)
  - Exclusively allocated node
- LIBRA with ranks pinned to dedicated CPU
  - Rank 0: polling mode
  - Rank 1: polling / event-based mode

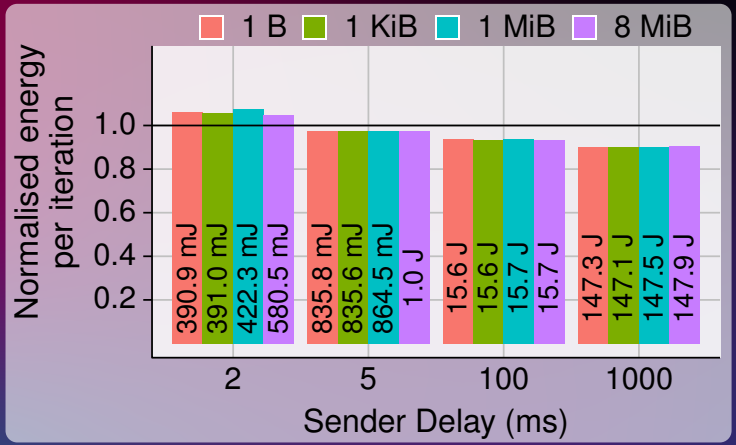
Evaluation: Latency (InfiniBand)



Evaluation: Latency (InfiniBand)



Evaluation: Energy of the Node (InfiniBand)



Conclusion

Summary

- `sched_yield` suboptimal for oversubscription
- Event-based communication in Open MPI with minimal code changes
- LIBRA micro-benchmark

Conclusion

Summary

- `sched_yield` suboptimal for oversubscription
- Event-based communication in Open MPI with minimal code changes
- LIBRA micro-benchmark
- Latency: Overhead of $\approx 90\,\mu s$ for small messages
- CPU energy: Savings of >10% for longer sender delays

Conclusion

Summary

- `sched_yield` suboptimal for oversubscription
- Event-based communication in Open MPI with minimal code changes
- LIBRA micro-benchmark
- Latency: Overhead of $\approx 90\,\mu s$ for small messages
- CPU energy: Savings of >10% for longer sender delays

Outlook

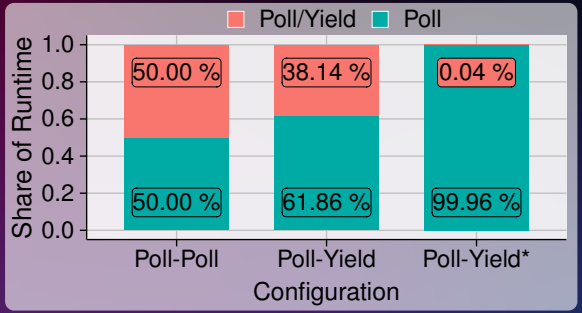
- Identify and mitigate sources of latency overhead
- Apply event-based communication to oversubscription of applications

`sched_yield` in Linux

- CFS (Completely Fair Scheduler) based on "virtual runtime"
- `sched_yield` well defined only for RT schedulers
- Implementation change in Linux 3.0

`sched_yield` in Linux

- CFS (Completely Fair Scheduler) based on “virtual runtime”
- `sched_yield` well defined only for RT schedulers
- Implementation change in Linux 3.0
- Busy loop micro-benchmark: fixed runtime



Evaluation: Energy of CPU 1 (InfiniBand)