JÜLICH
FORSCHUNGSZENTRUM

Performance Evaluation of Scientific Applications on POWER8

2014 Nov 16

Andrew V. Adinetz[1], **Paul F. Baumeister**[1],
Hans Böttiger[3], Thorsten Hater[1], Thilo Maurer[3],
Dirk Pleiter[1], Wolfram Schenck[4], and S. Fabio Schifano[2]

[1] Jülich Supercomputing Centre, Forschungszentrum Jülich, 52425 Jülich, Germany
[2] Dip. di Matematica e Informatica, Università di Ferrara and INFN, Ferrara, Italy
[3] IBM Deutschland Research & Development GmbH, 71032 Böblingen, Germany
[4] SimLab Neuroscience, IAS, Forschungszentrum Jülich, 52425 Jülich, Germany

Outline

- POWER8 architecture intro
- Micro benchmarks
    - STREAM memory bandwidth
    - OpenMP overheads
- Scientific HPC applications
    - LBM Lattice Boltzmann method for fluid dynamics
    - NEST Neuroscience network simulator
    - MAFIA subspace clustering data analysis tool

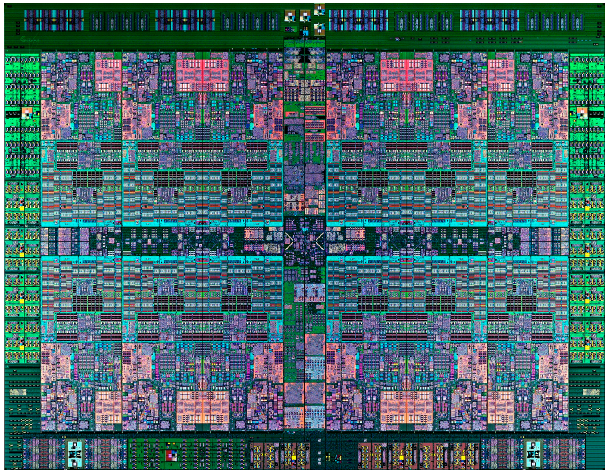
POWER8 chip architecture

3.42 GHz core frequency

10 cores per socket (max 12)

core architecture:

- POWER ISA v2.07
- 8-way SMT
- 2xFXU,
- 2xLSU, 2xLU
- 2xVMX, 4xFPU



Memory hierarchy of POWER8

L1$: 64k /core

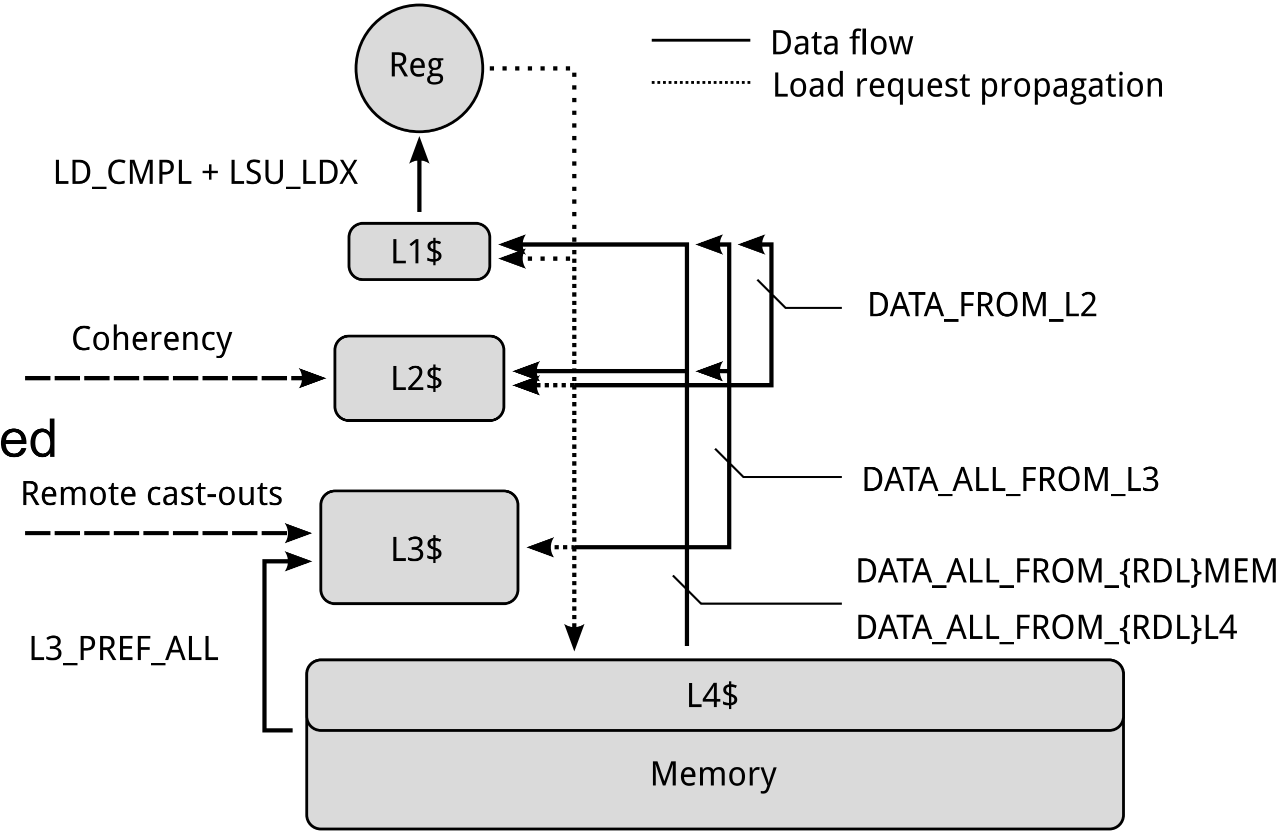
L2$: 512k /core

L3$: 8M /core shared

Centaur:

L4$: max. 128M

Mem: 1T max.



STREAM benchmark

- with 2 streams

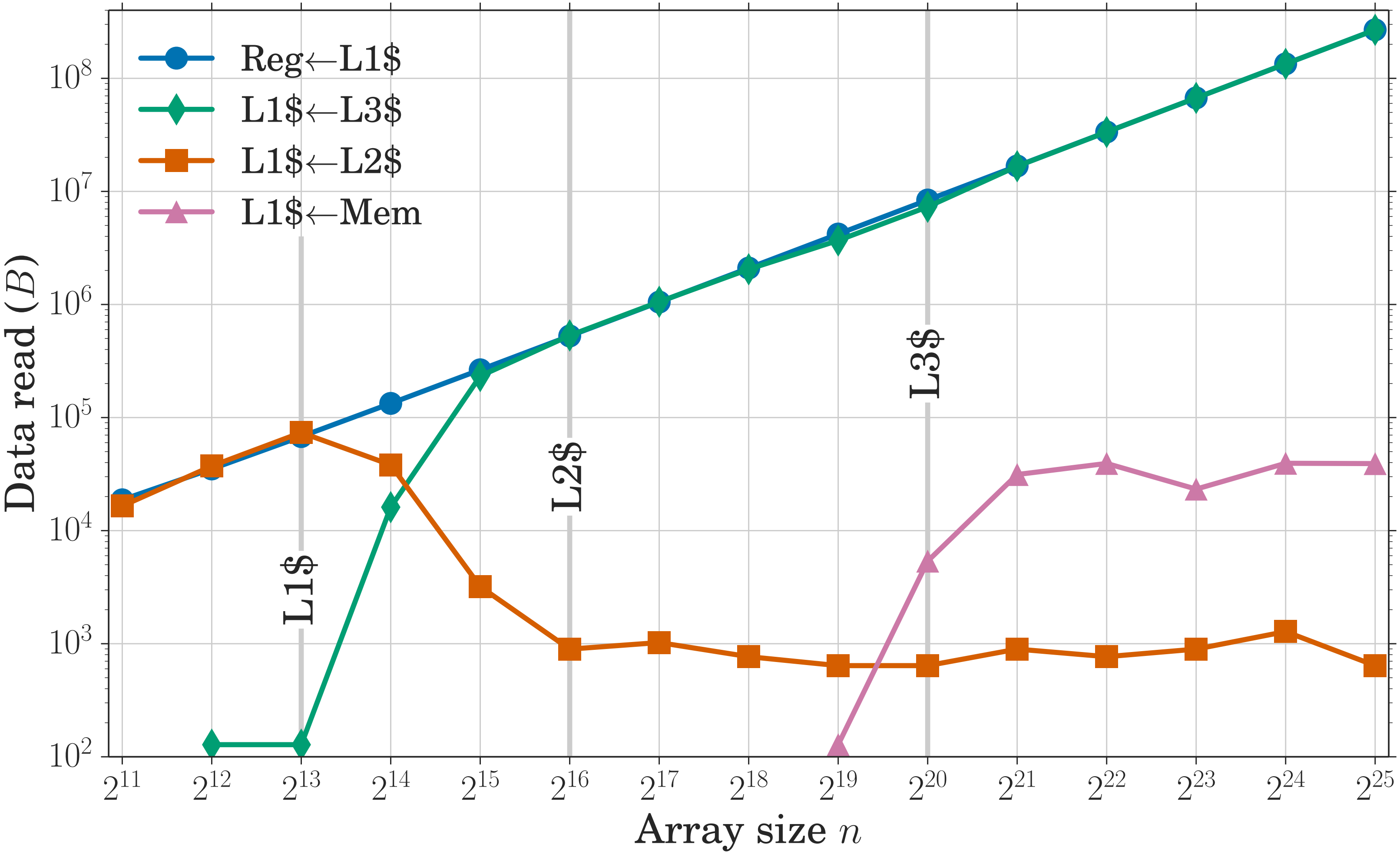
```
copy        c[i] = a[i];
scale       b[i] = s*c[i];
```

- with 3 streams

```
sum         a[i] = b[i]+c[i];
triad       a[i] = s*b[i]+c[i];
```

- efficient prefetching expected

STREAM memory traffic - copy



Mitglied der Helmholtz-Gemeinschaft

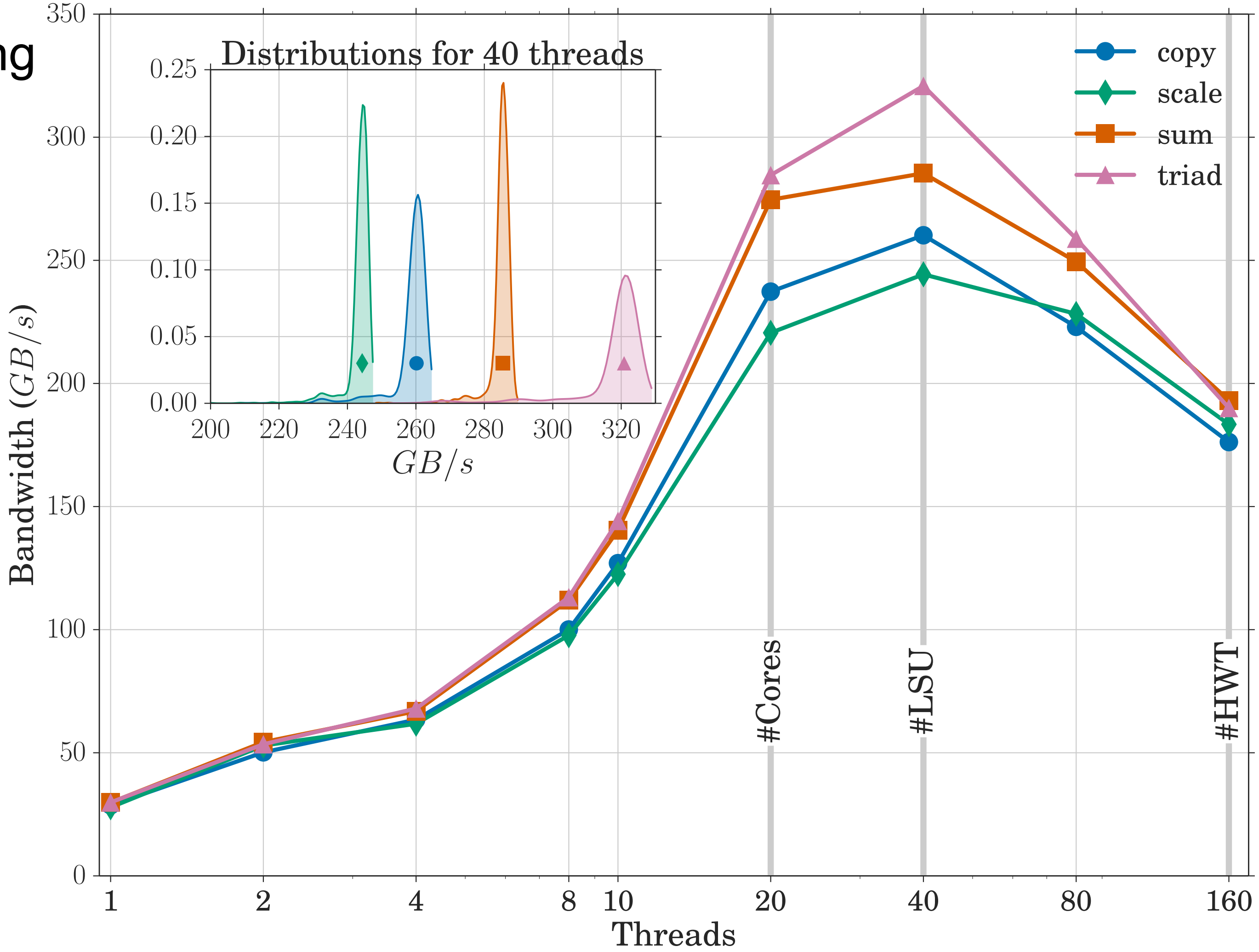
STREAM performance

sampling

1000x

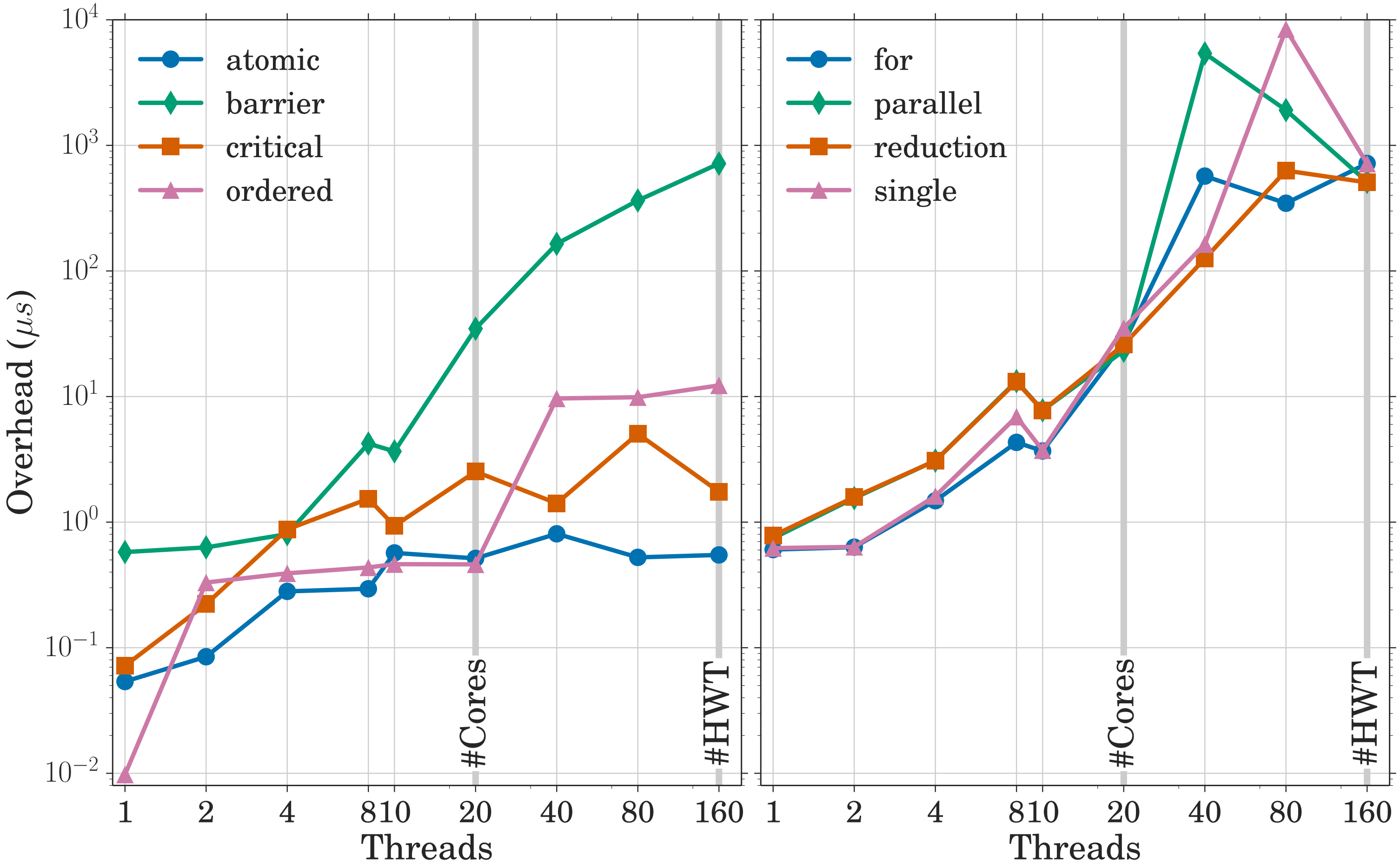
median

values



OpenMP overheads

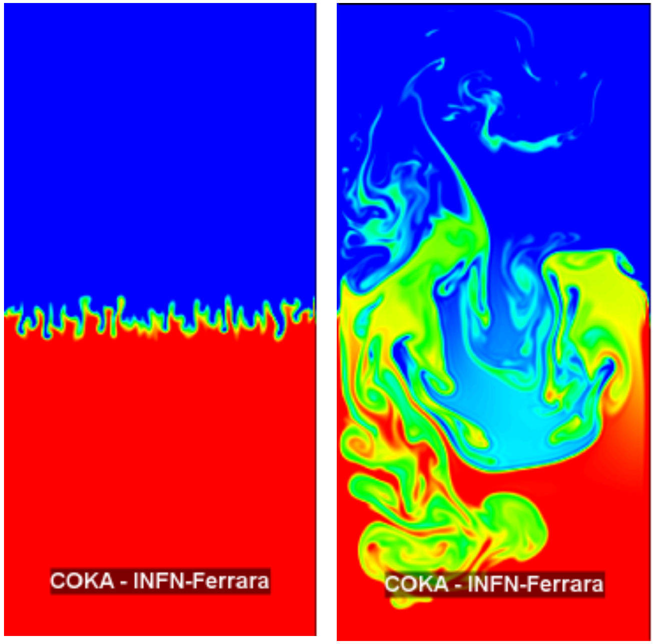
Explicit vs. implicit constructs: overhead = $t(n) - t_{\mathrm{serial}}/n$



Lattice Boltzmann method

Fluid dynamics

- discrete sampling of positions and velocities
- complex and irregular structures
- multi-phase flow possible



Example: Rayleigh-Taylor instability

Lattice Boltzmann method – D2Q37

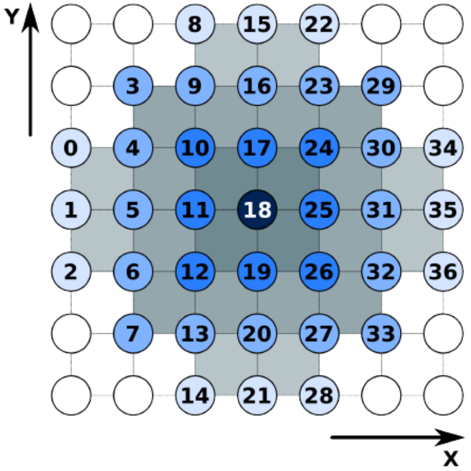
- two kernels:

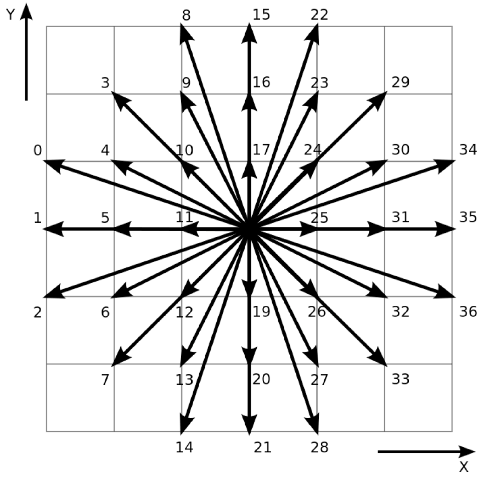
  `collide` (6200 FP/lattice site)

  `propagate` (BW limited)

  on most platforms $t_{\text{collide}} \gg t_{\text{propagate}}$

- high degree of parallelism
- high arithmetic intensity





Mitglied der Helmholtz-Gemeinschaft

LBM D2Q37 – performance

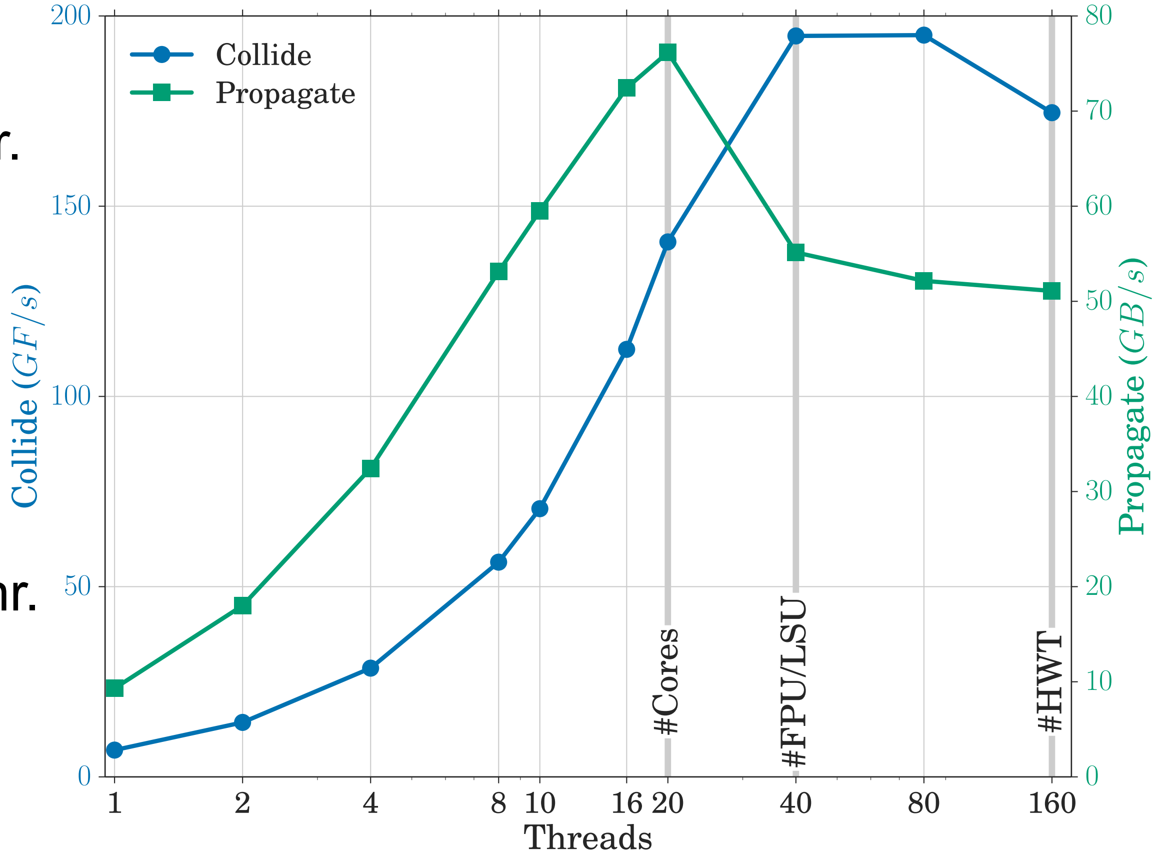
best results ~200 GF/s

for collide at 40 or 80 thr.

best results ~75 GB/s

for propagate at 20 thr.

shortest runtime at 40 thr.



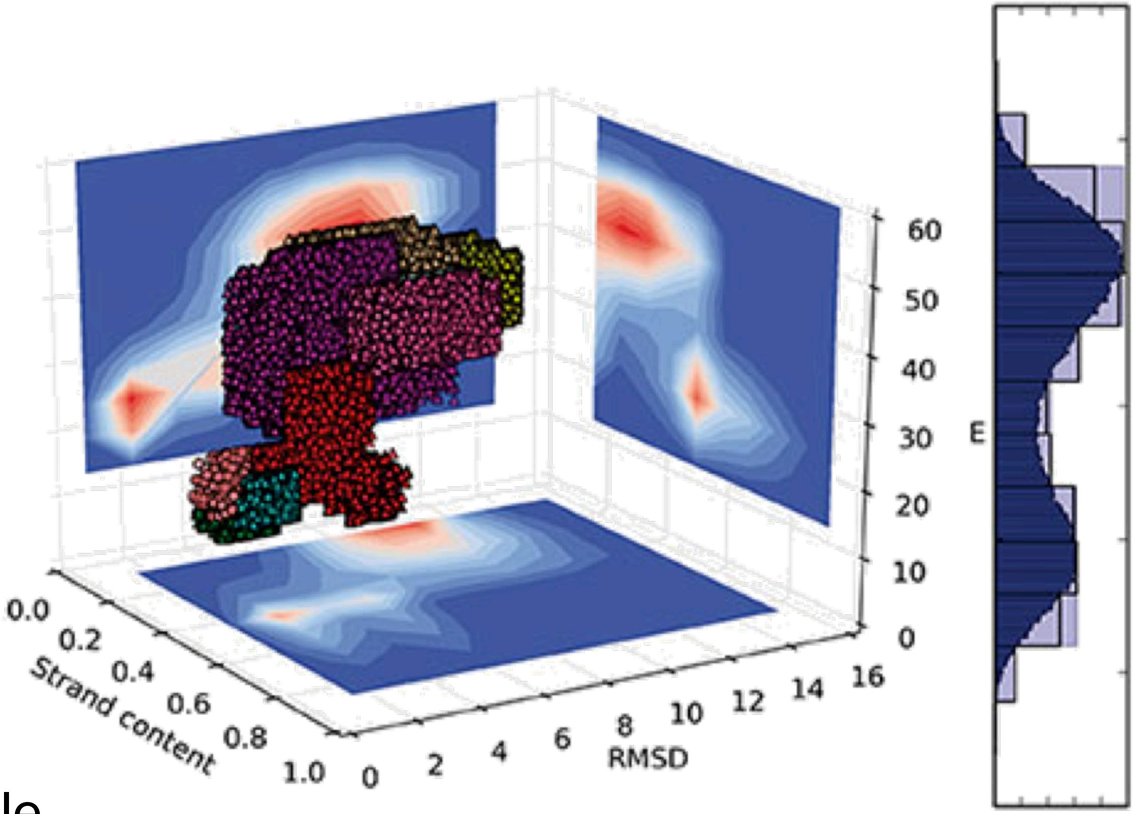
MAFIA

a parallelized algorithm to identify subspace clusters in high dimensional spaces

kernel: `pcount`

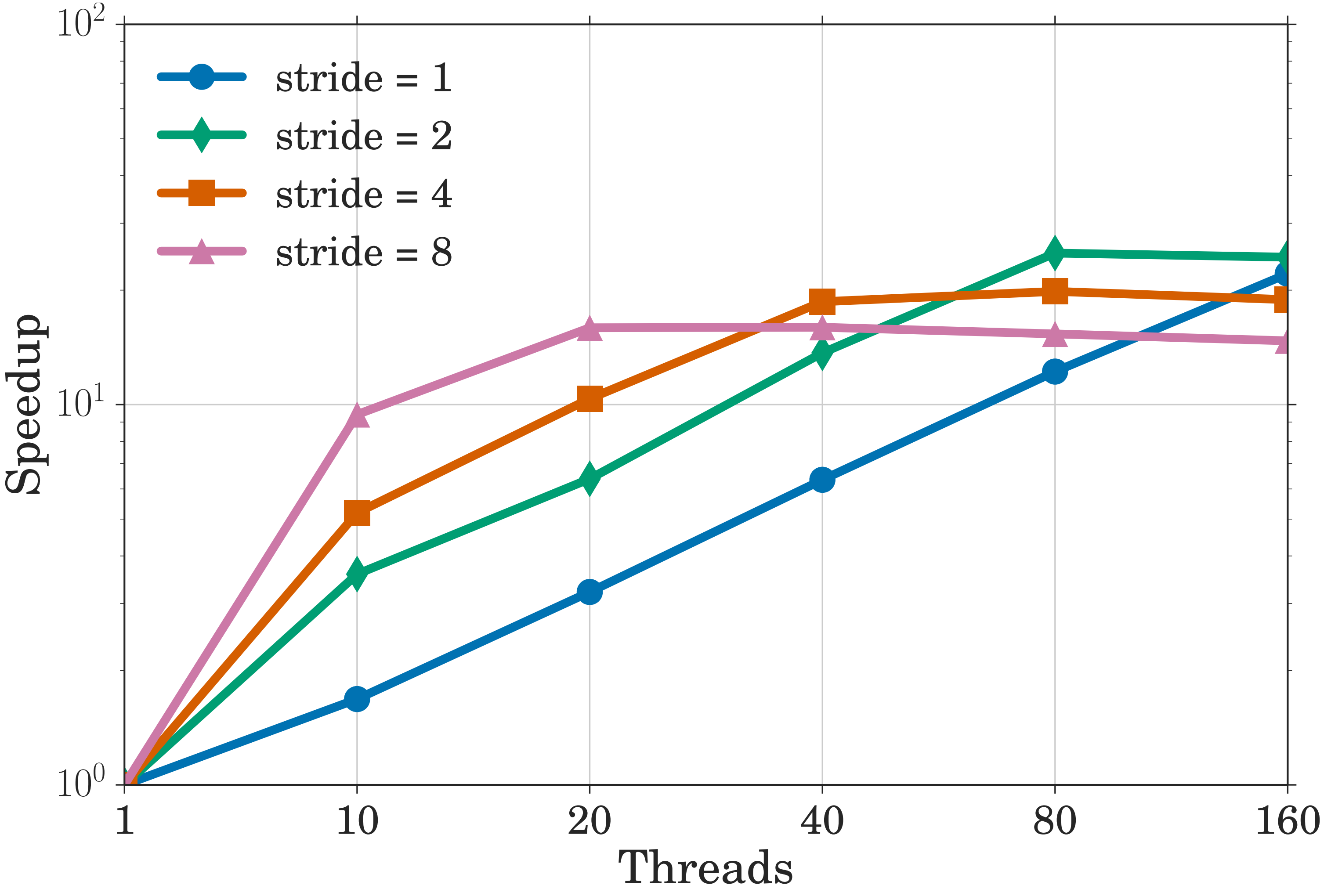
bitwise &

GPU acceleration possible



Adinetz *et al.*, Euro-Par'13 Proceedings, pp 838-849

MAFIA – performance

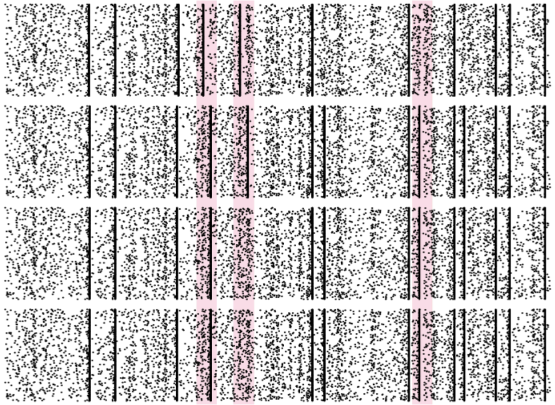


NEST – NEural Simulation Tool

nest::
simulated()

nest-initiative.org

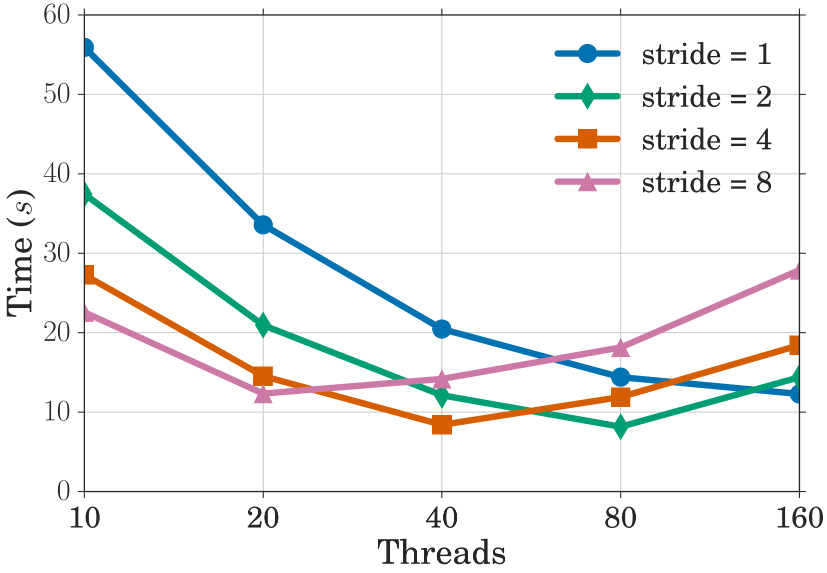
- discrete event simulator on an a distributed graph
- uses domains-specific interpreter language (SLI)
- stochastic input
- hybrid parallelization with *M*×*T* virtual processes
- *n*=9375 neurons per MPI process
- dry-run mode for *M* processes
- kernels:
  - neuron update (some FP-ops)
  - spike delivery (dominant)

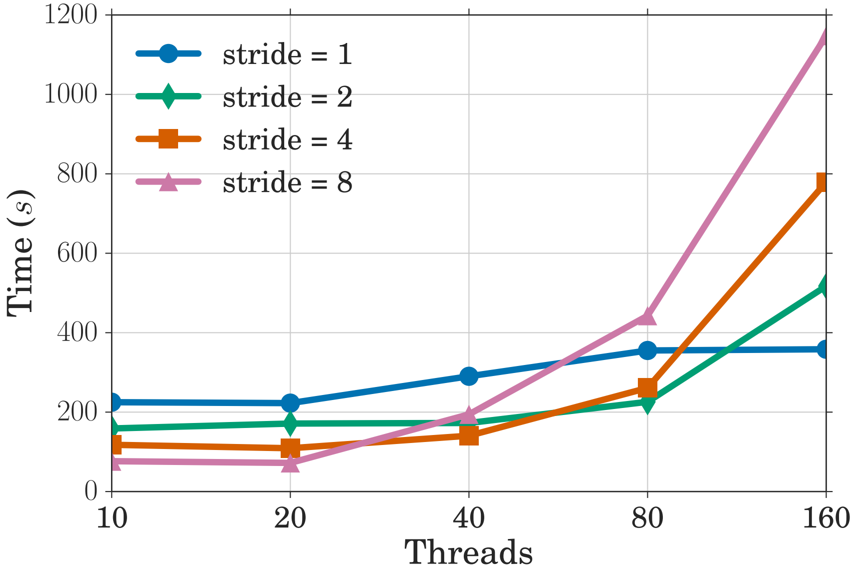


Brette *et al.* J Comp. Neuroscience Vol 23, 3 (2007)

NEST – total runtime

- control the binding of threads to cores
- dry-run mode with M=512 and M=16,384





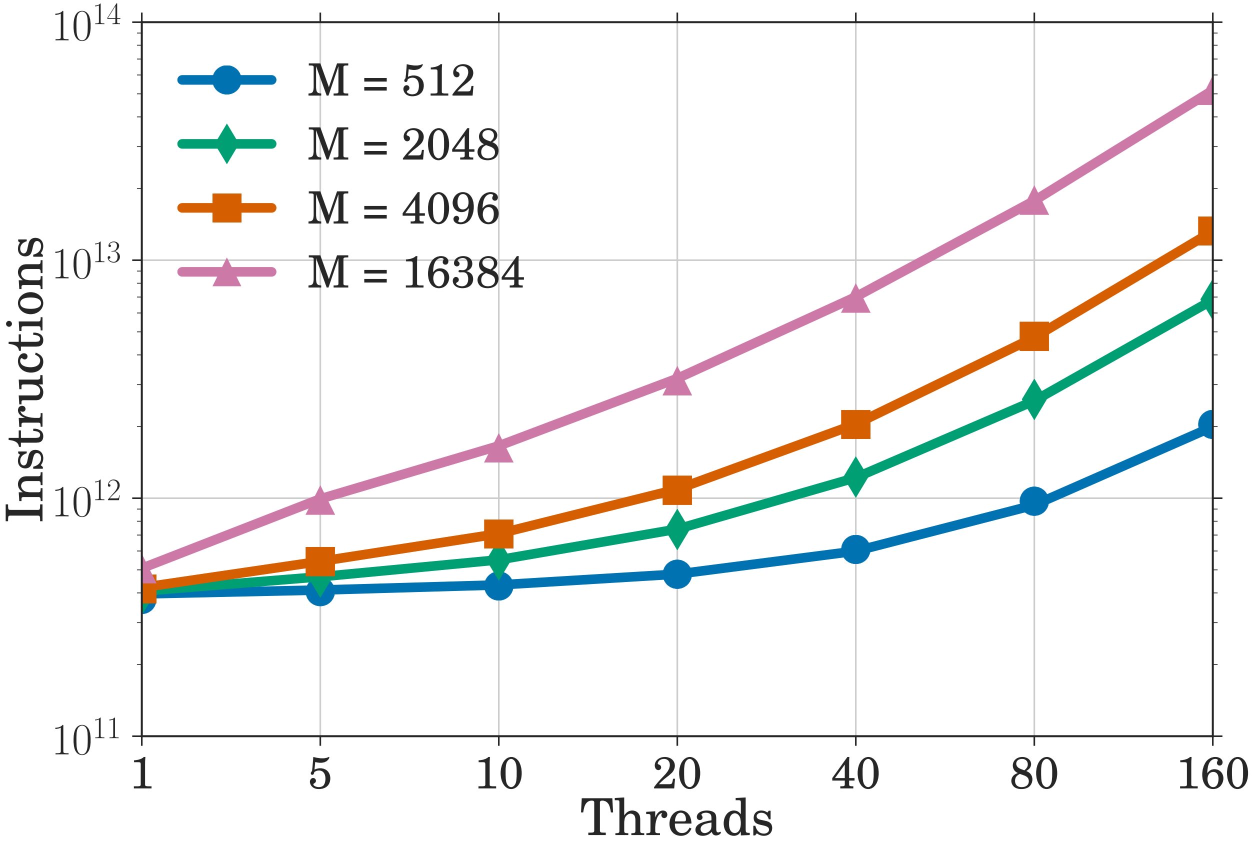
- use 160/T as stride

NEST – spike delivery

total amount of work fitted to $C = c_0 + c_1\, MT + c_2\, MT^2$

points: measured

lines: fit



NEST – potential for improvement

- Each of T threads runs through MT buffer elements in the spike delivery kernel, avoiding the $c_2\ MT^2$ component would improve thread-scalability
- Instructions per cycle throughput benefits from SMT8
- Needs deeper understanding

Summary

- LBM and MAFIA benefit from vectorization and instruction-level parallelism
- NEST benefits from SMT8, needs restructuring
- POWER8 suitable as host CPU in GPU-accelerated system

Outlook

POWER8 with GPUs

installed – porting in progress