STATUS OF HUMAN BRAIN PROJECT APPLICATIONS ON ARM

Dirk Pleiter, A. Herten, S. Nassyr, B. Brank | Arm HPC Workshop, Frankfurt | 20.06.2019

JÜLICH
Forschungszentrum

Overview

- Introduction and context
- Selected application benchmark results
- Summary and conclusions
- Outlook: Open Edge and HPC Initiative

Background

Fenix and the ICEI project

- Goals
  - Establish HPC and data infrastructure services for multiple research communities
    - Prime community: brain research community organised in the Human Brain Project
  - Develop and deploy services enabling federation
  - Follow a use-case driven co-design approach
- Fenix sites: BSC, CEA, CINECA, CSCS, JSC
- Project focusses on coordinated procurements for realising infrastructure

ICEI benchmark suite

- Set of 8 benchmarks covering the identified through HBP science and use cases

Topic of this talk

- Given a specific ICEI cluster procurement, where do we stand in terms of Arm-based solutions?

Hardware platforms for benchmark evaluation

	JUAWEI	JURON	JUWELS
Type of CPU	Hi1616	POWER8	Xeon Platinum 8168
#sockets * #cores	2 * 32	2 * 20	2 * 48
SIMD width [bit]	128	128	512
Throughput [Flop/cycle]	256	160	1,536
Memory capacity [GiByte]	256	256	96
Memory bandwidth [GByte/s]	136	230	255

Compilers

JUAWEI

- gcc 8.2.0
- clang version 7.0.1
- Arm C/C++/Fortran Compiler version 19.2 (build number 155) (based on LLVM 7.1.0)

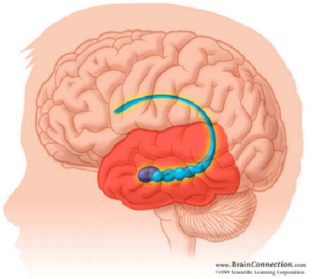
JURON

- gcc 5.4.0 and gcc 6.3.0

JUWELS

- gcc 5.5.0 and gcc 8.2.0

NEST: Application background

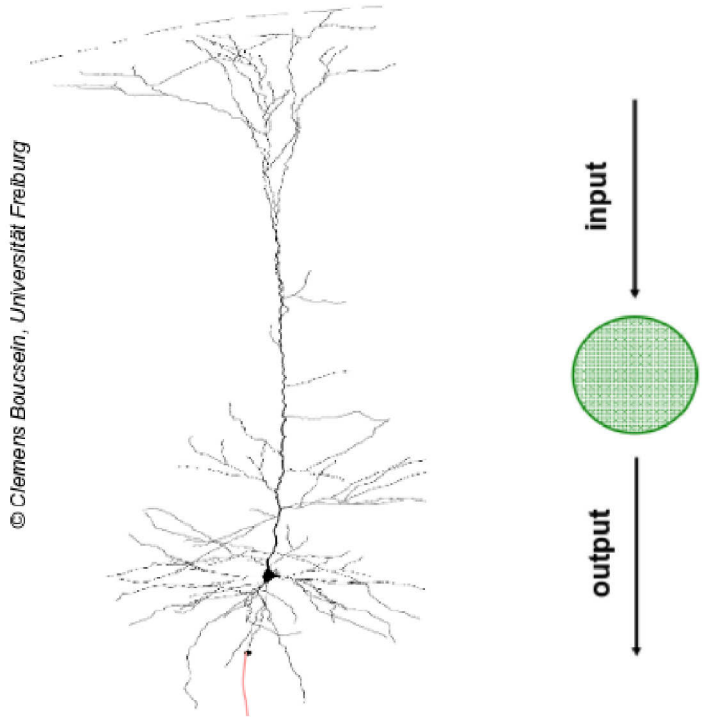


Features

- Focus on simulation of point-neurons, simple integrate-and-fire model
- Application phases: Network construction and connection + simulation

Performance features

- High level of concurrency
- Simple models
  - Most time spent on spike delivery
  - Little time on neuron modelling
- Complex control flow

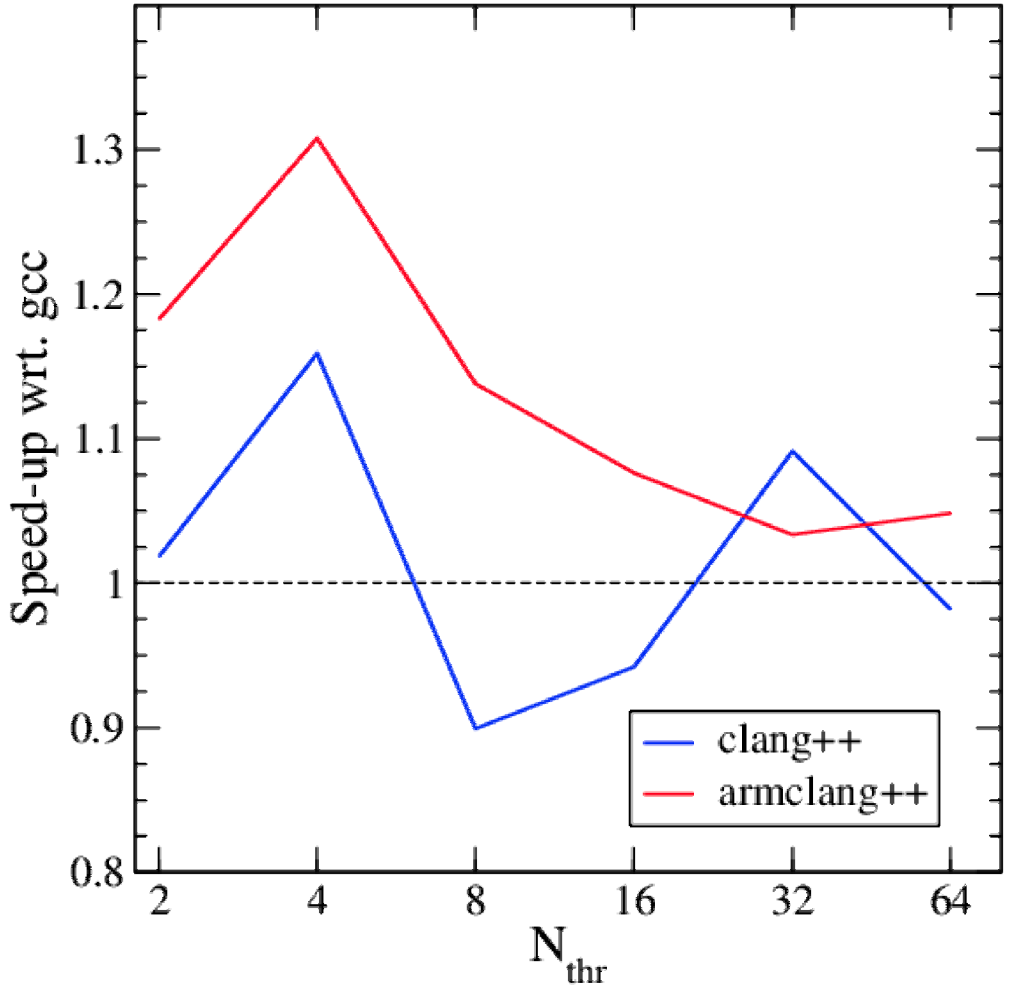


Implementation

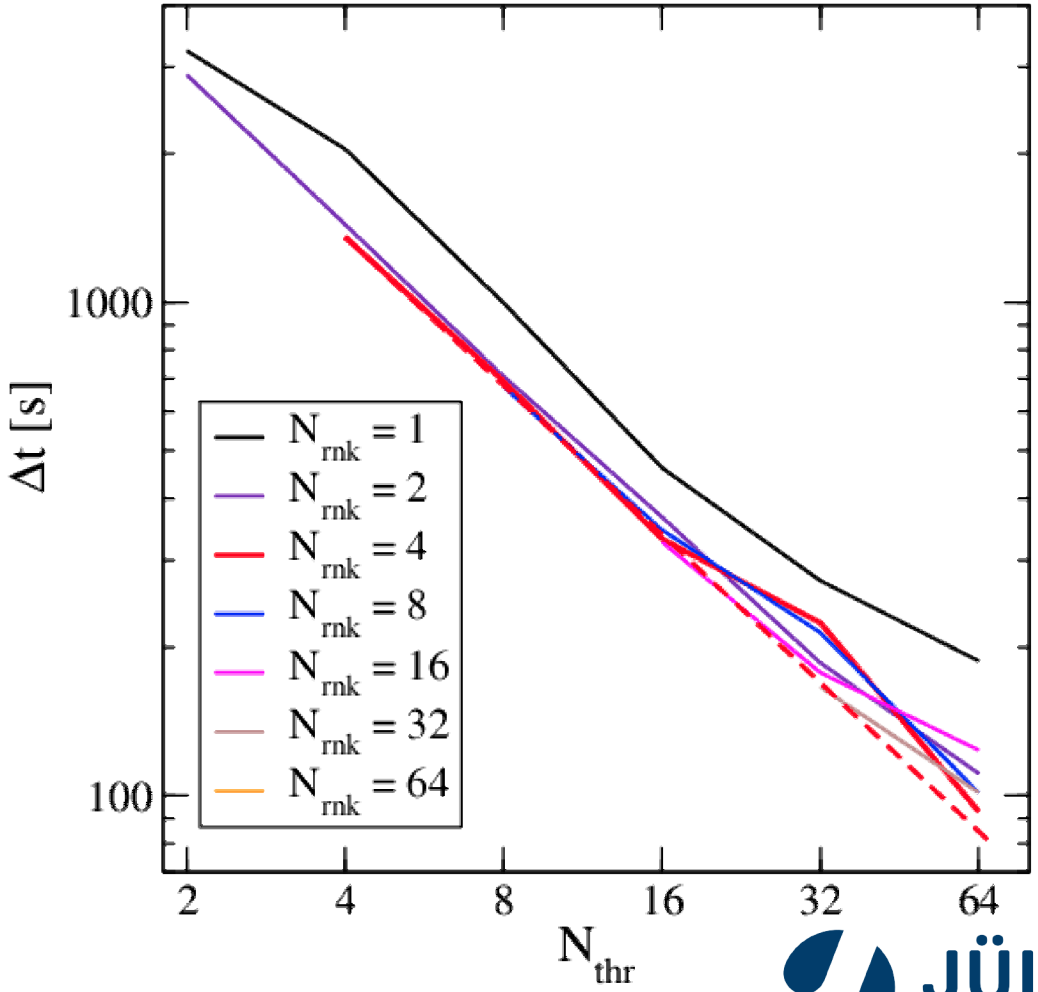
- Advanced C++
- Parallelisation: MPI, OpenMP, no vectorisation
- No third-party dependencies (except for Python front-end)

NEST: Single node performance Hi1616

Comparison compilers (taking gcc as reference)

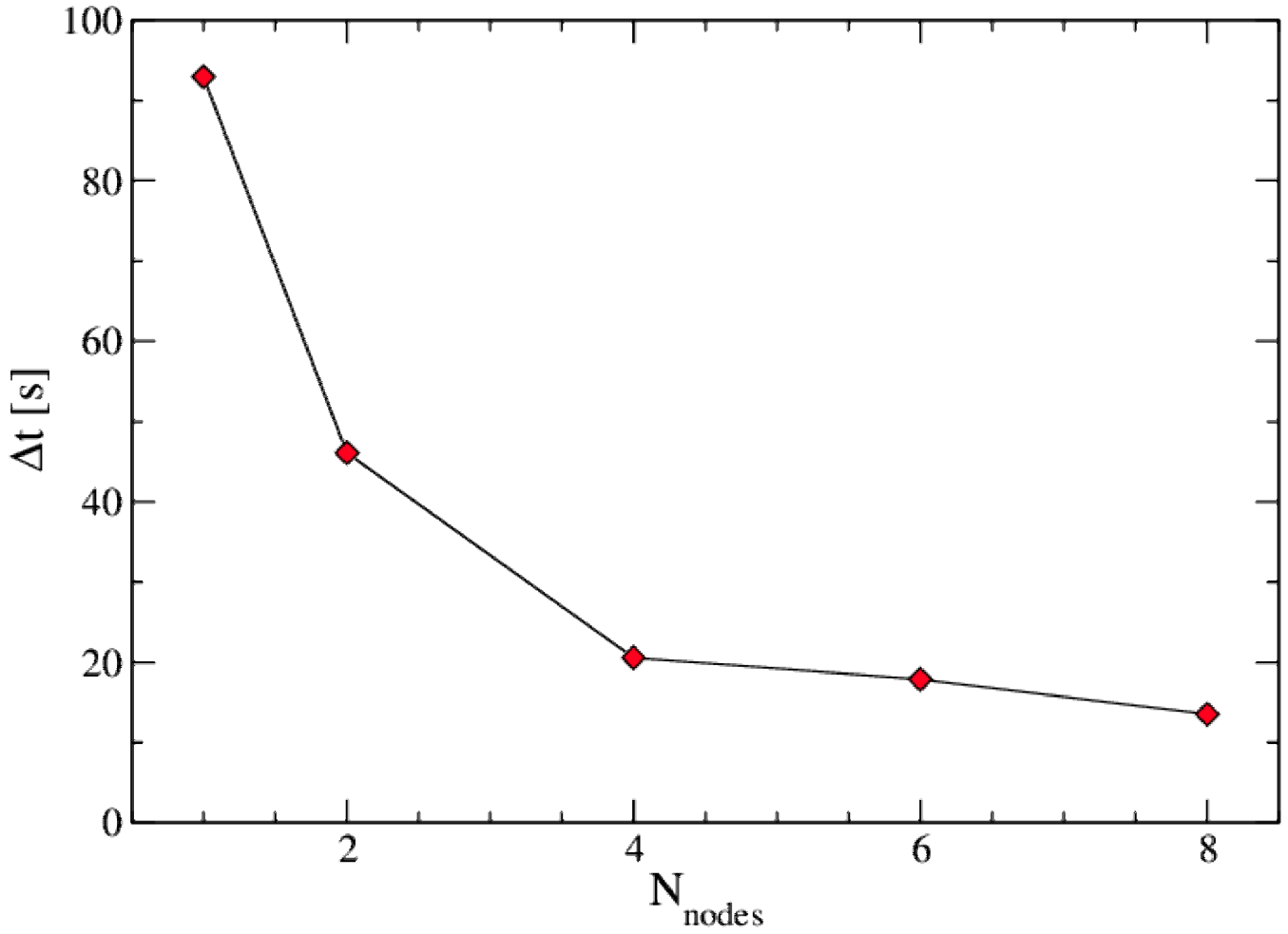


Weak scaling for gcc



JÜLICH Forschungszentrum

NEST: Multi-node strong scaling

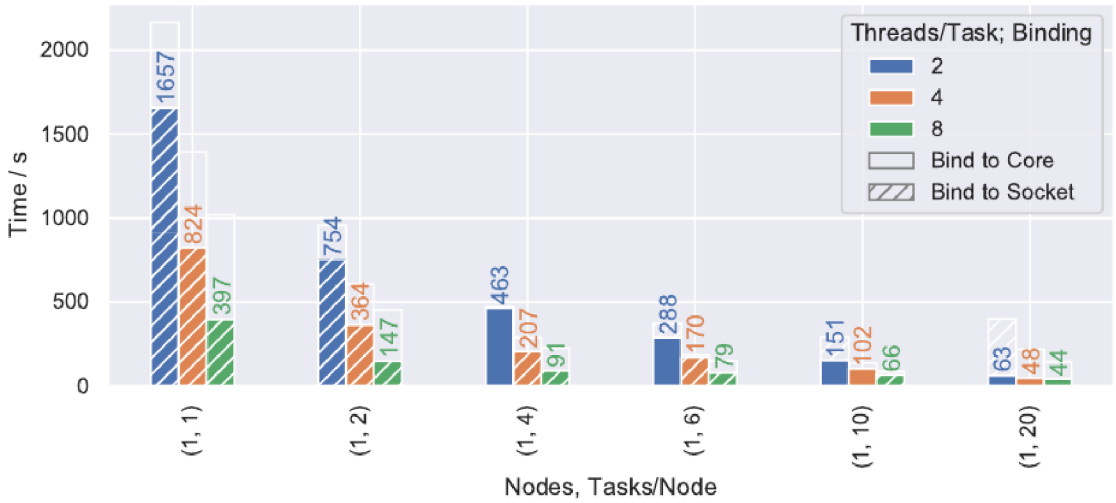


- MPI communication performance on this system not yet close to hardware capabilities
- Application developers are more interested in weak scaling

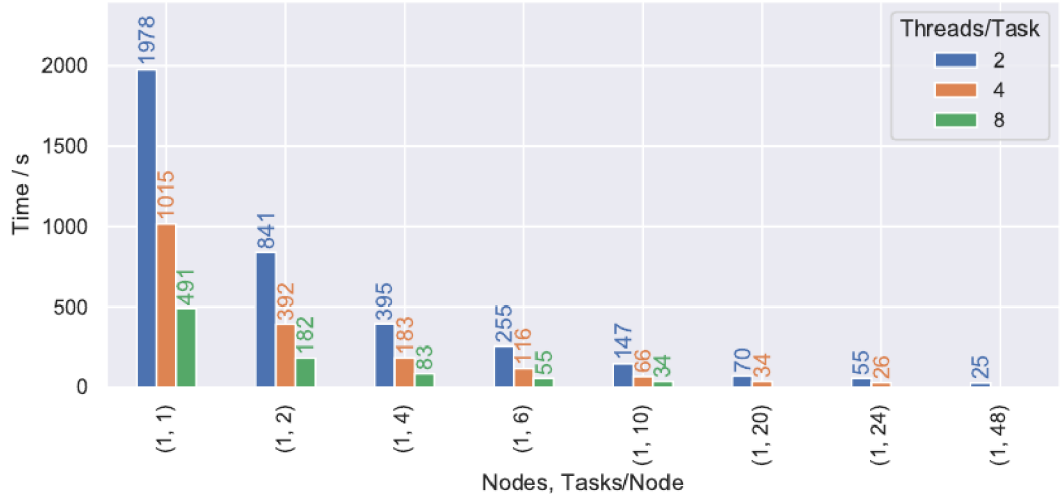
NEST: Comparison with others

[A. Herten et al., IWOPH'19]

JURON



JUWELS



Observations

- Single-node scalability excellent on all architectures due to high amount of concurrency
- JURON and JUWELS about 2-3x faster consistent with expected memory bandwidth limitations
- Choice of compilers is a 5% effect

Arbor: Application background

Features

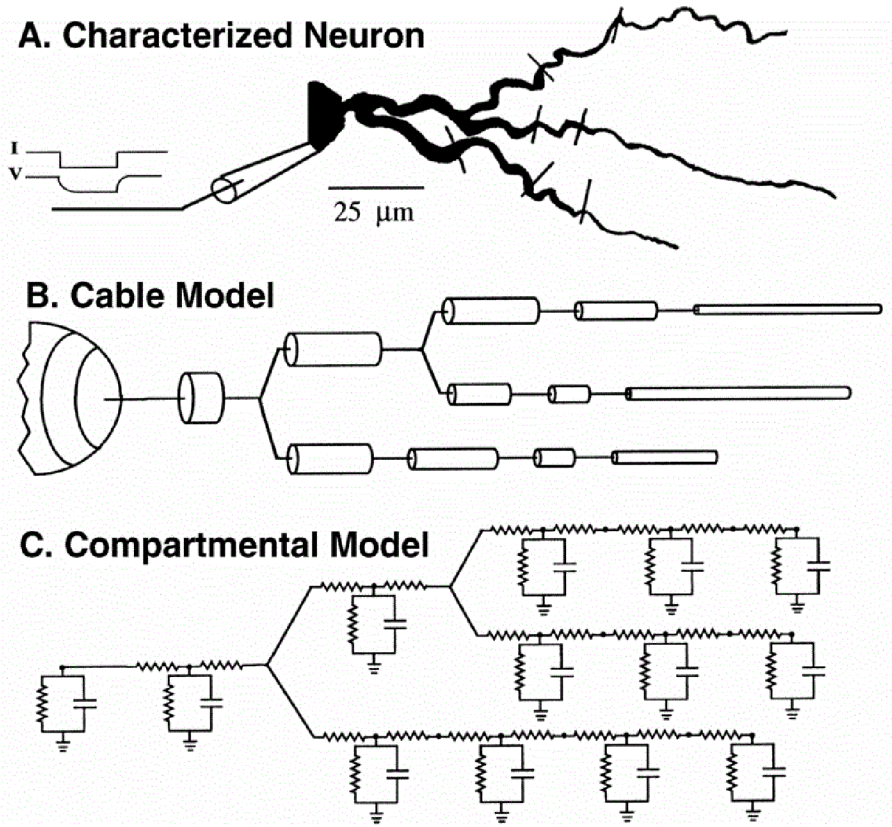
- Similar to NEST, but using detailed models instead of point neurons

Performance features

- Even higher level of concurrency
- Compute-intensive kernel: Hines matrix solver
- More structured memory accesses compared to NEST

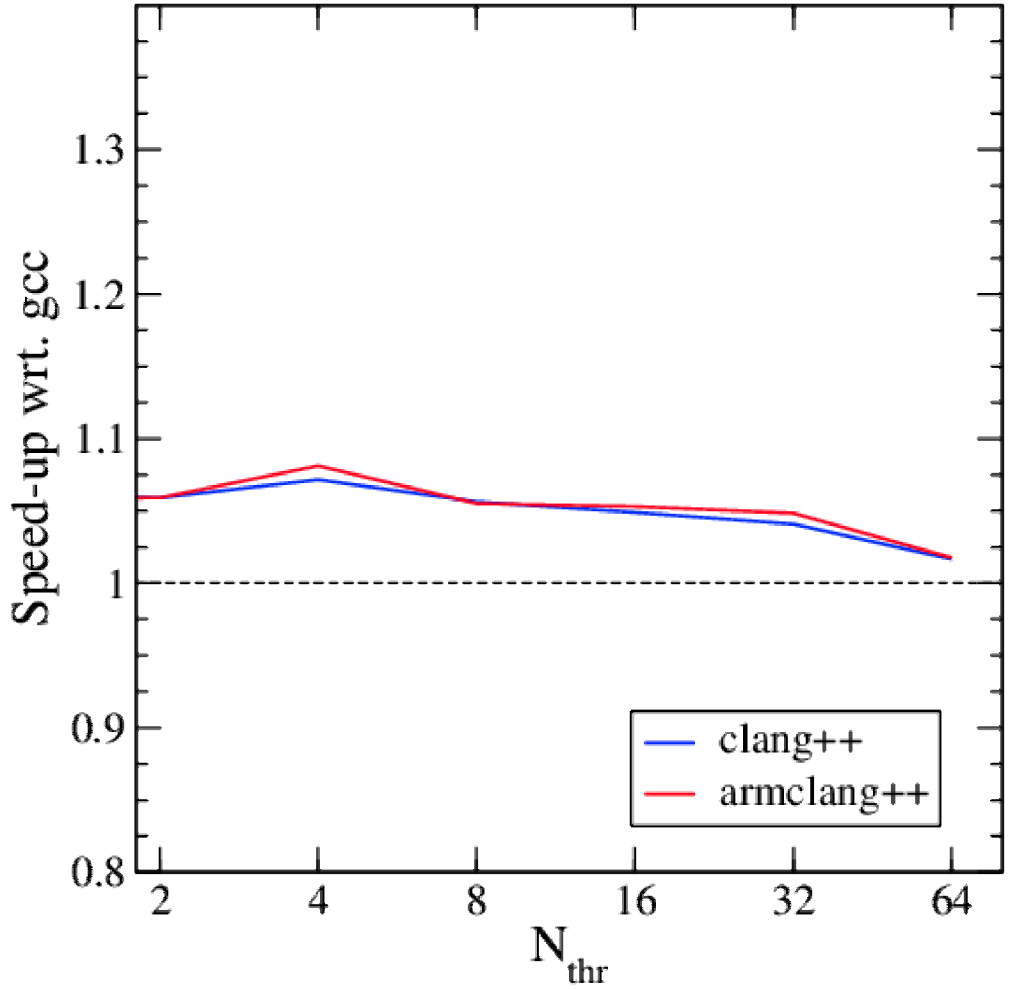
Implementation

- Advanced C++
- Parallelisation: MPI, OpenMP, vectorisation
- No third-party dependencies (except for Python front-end)

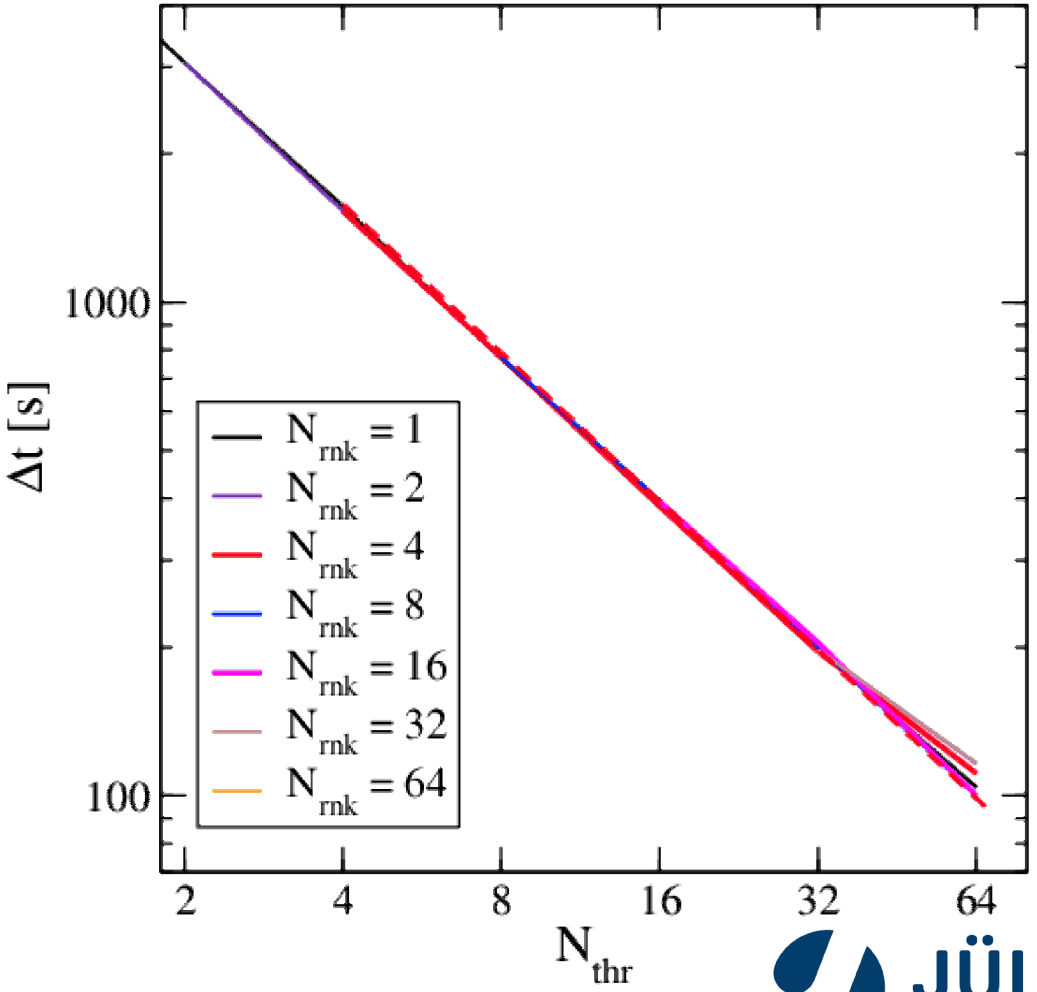


Arbor: Single node performance Hi1616

Comparison compilers (taking gcc as reference)

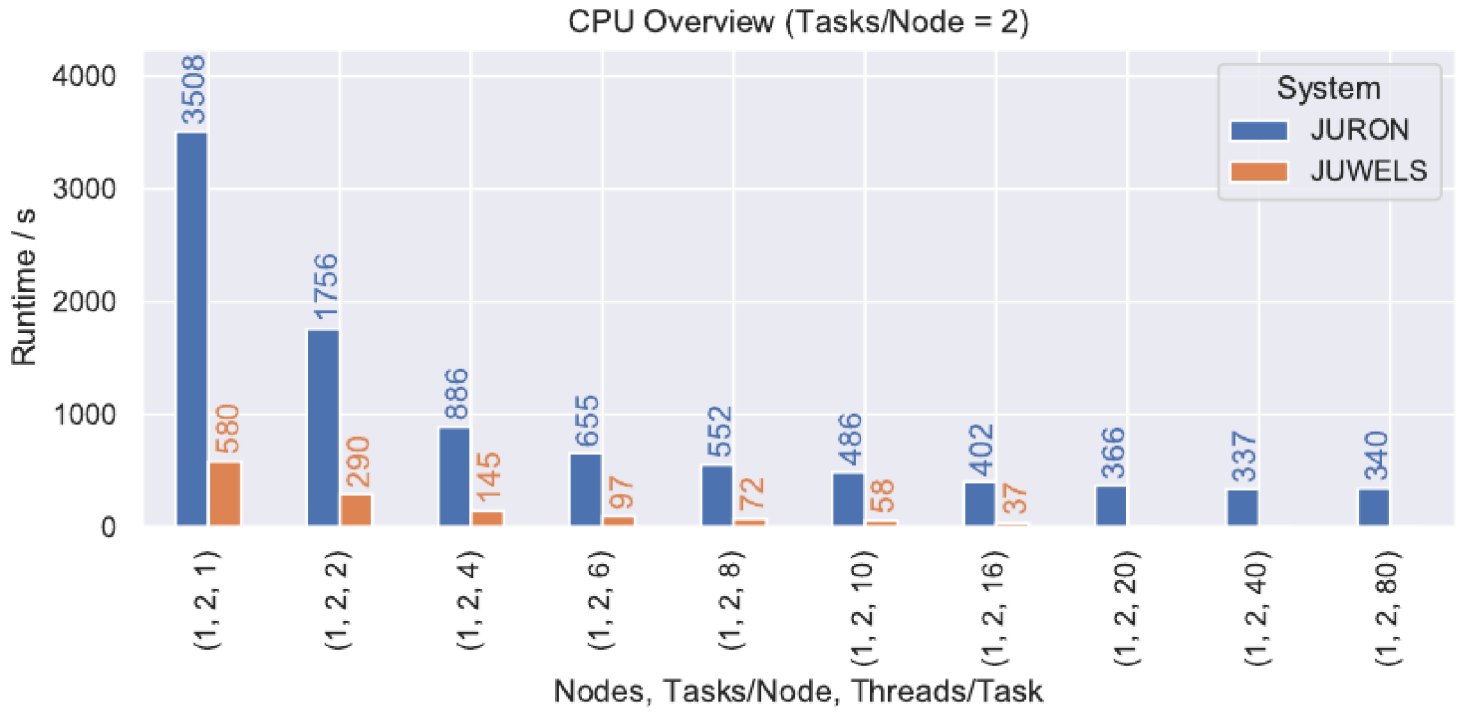


Scaling for gcc



JÜLICH Forschungszentrum

Arbor: Comparison with others



Observations

- Perfect scaling within a node due to the high level of concurrency
- Performance on JUAWEI 3.4x faster and 2.7x slower compared to JURON and JUWELS, respectively
  - Naively expect similar performance and 7x slow-down, respectively
- LLVM about 5% faster compared to gcc

TVB-HPC: Application background

Features

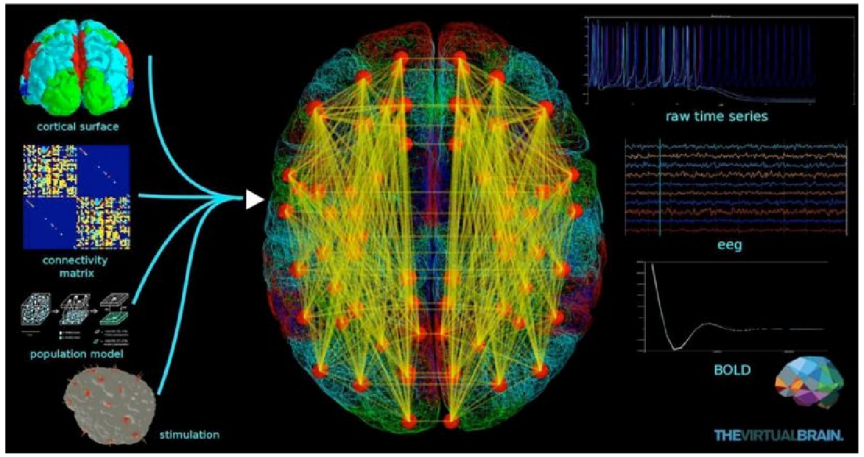
- Full brain network simulator mesoscopic models of neural dynamics
- Generates data that can be used for comparison with experimental data, e.g. EEG data

Performance features

- Kernel: Kuramoto model update
- Regular control flow, but irregular memory access
- Dominantly compute of trigonometric functions

Implementation

- Python + Numba



[P. S. Leon et al., doi:10.1186/1471-2202-14-S1-P193]

Digression: Introduction to Numba

Solution for acceleration of Python codes

- Numba translates Python functions to optimised machine code at runtime using the LLVM compiler library

Supported platforms

- x86 (32-/64-bit) , ppcle64
- NVIDIA GPUs, AMD ROC dGPUs
- ARMv7 (32-bit)

Technology

- Python byte code translated to Numba IR
- Numba IR translated to machine code

```
from numba import jit

@jit
def f(x, y):
    return x + y
```

TVB-HPC: Comparison of results

JUAWEI	40 s
JURON	31 s
JUWELS	35.48 s

Observations

- Similar performance on all considered architectures
- Full support of Numba lacking
  - Requires Numba to be compiled from scratch (officially not recommended)
  - Various unit tests fail (further analysis pending)
- Failed to enable all features of Numba on JUAWEI

Other requirements

GPU support

- Deep learning for brain image analysis
- Visualisation
- Increase of compute density and power efficiency

Parallel filesystem support

- In our case: GPFS → not officially supported

Openstack support

- Technically seems to work, but full testing lacking
- Commercial support likely an issue

Potential for SVE-Based Solutions

NEST and Elephant ASSET

- Codes not suitable or not ready for vectorisation

Arbor

- Support for SIMD built-in using templating approach
  - Including vectorised implementation of exponentials
- Challenge: Fixed vector length assumed
- Solution: Fix vector length at compile time
  - Tested using armie

TVB-HPC / Numba

- Numba directly does not support vectorisation and relies on compiler auto-vectorisation

Conclusions

Fenix: Federated IT infrastructure provided by leading European supercomputing centres

- Focus on supporting brain research

Given a specific ICEI cluster procurement, where do we stand in terms of Arm-based solutions?

- Clear difference of performance for considered generation of technology
  - Expect performance gap to close with upcoming solutions
- Programming models like Numba still to be fully supported on Arm
- Various other requirements in this specific context still difficult to meet with Arm-based solutions
  - Lacking GPU support will be closed in the future

Open Edge and HPC Initiative

http://www.openedgehpcinitiative.org/

Our Mission

Foster the development of an open and feature-rich **ecosystem for Arm®-based technologies** to support the evolving needs of the various industries undergoing digitalization and of all their respective stakeholders

HPC Ongoing Activities

- Identifying a list of computing kernels that can benefit strongly from Arm®-based systems
- Collaborating and supporting ISVs in porting their packages
- Access to state-of-the-art Arm-based hardware
  - ThunderX2 servers from E4
  - Kunpeng 916 and 920 servers from Huawei

Visit website for Codes@OEHI