Project Proposal for HPC Access to JURECA and JUWEL

Period *01.05.2018 - 30.04.2019*

Project title
Arbor Big Scaling

Type of project
New project

Principal investigator
Dr. Alexander Peyser, Forschungszentrum Jülich

Project contributor(s)

- *Wouter Klijn, Forschungszentrum Jülich*
- *Ben Cumming, Swiss National Supercomputing Center, Switzerland*
- *Stuart Yates, Swiss National Supercomputing Center Switzerland*

Contents

1 Introduction

Give a short outline of the scientific background of your research, including references.
Current and future HPC systems are increasingly include heterogenious many-core architectures like GPUs and Knights landing. Arbor is a new morphologically detailed neural simulator designed to exploit these architectures . Designing from the ground-up for heterogeneous and many-core systems it aims to exploid the potential of these large systems for the neuroscientific community.
The performance of Arbor has been tested on small HPC resources and good scaling has been shown on medium to large systems. These tests where done with very constraint and simplistic network. In this compute time proposal we ask time to perform scaling experiments on the diverse compute resources available in Jülich with two neural networks having both a realistic spike rate and biologically realistic connectivity pattern. This will allow us to assertain the performance under realistic loads.
The models used for benchmarking are a simplified cortical model based on 1,2 and a hippocampus model based on 3,4. In pilot experiments it was established that with under specific parameter settings these networks have a property essential for scaling experiments: The networks can be scaled up to arbitrary size without
(about 1 page)

2 Preliminary Work

Provide a brief summary of your preliminary work in connection with the proposed project, including references.

The Arbor simulator has not yet had an official release. Development is done in an open fassion, the software has been online available online for more then a year (5). Preliminiairy results have been presented at numerous accacions in the form of posters and presentation.
Dry run mode
Big scaling week
pilot experiments
(about 1 to 2 pages)

3 Description of the Project

3.1 Project Details

im of the project is to validate the large scale performance of the Arbor simulator on large systems, while simulating two models with biological realistic spike rate and bulk connectivity. Two biological systems at extremes of these two

- Scientific questions you want to address Validation of the performance of the spike communication under realistic load. For a diverse set of realist models with divert levels of details in the cel simulations.
- Scientific objectives This compute time proposal only has computational aspects. No specific scientific questions will be explored.
- Computational objectives A number of different internode and intra node exchange methods have been implemented for Arbor. What method is optimal for the different hardware architectures and loads is currently not known. A dryrun mode is available for initial estimations of performance, This method can not reproduce the exact resource load. Cache cehorence and memory bandwith bottlenecks are highly depending on the workload and different hardware configurations available.
- Approach and expected outcome The different communication methods currently available will be compared for a predetermined set neuroscientific models. Each models with increasing complexity and shifting workloads. If major bottle necks are encountered these will be investigated and hopefully alleviated.
- Expected impact on the research area The Arbor simulator will allow large scale simulations of morphologically detailed neural networks on many-core architectures. Like the knight landing, GPU's and at a later stage FPGA and the Chinese hardware (ref)
- Scientific and technical innovation potential The Arbor software is developed with a modular approach: The different components are abstracted with clear internal APIs this allows other simulators to reuse components to improve performance or expand capability.
- Progress beyond the state-of-the-art

3.1.1 Sub-project 1

3.1.2 Sub-project 2

(1 to 2 pages per sub-project)

3.2 Review Processes

Has the underlying research project already successfully undergone a scientific review process? Is the project funded by public money? If yes, please also provide information about the funding source (e.g. State, BMWi, BMBF, DFG, EU, . . .)

4 Numerical Methods and Algorithms

Describe the numerical methods and algorithms that you are planning to use, improve, or de-

velop.

(1 to 2 pages)

5 Computer Resources

5.1 Code performance and workflow

Describe the codes, packages or libraries that you need to undertake the project, and how these will enable the research to be achieved. Include for ***each code to be used*** *information about*

- *Which code will be used The code used can be found at (6).*
- *How is the code parallelized (pure MPI, mixed MPI/OpenMP, Pthreads, CUDA, etc.) The Arbor simulator is build with multiple backends in mind. The paralllization is thus diverse. The node paralellizetion is performed with MPI. Node threading is performed with the Intel Tread building blocks and a hardwritten C threads librairy. Further paralellizetion is done depending on the many-core backend: vectorization with AVX on the appropriate backend including intrinsics. For GPUs Cuda is used.*
- *The amount of memory necessary (per core, per node and in total) The amount of memory used is minimal. Running a network with 400000 neurons, ten times more then typically expected, uses a maximum of 1712.982 MB on a KNL system. Arbor is compute limitted in the extreme and memory constraints are not under considiration.*
- *Scaling plots* ***and*** *tables with speedup results for runs with typical, parameter sets, problem size, and I/O* ***of the planned project*** *(no general benchmark results are accepted)*
- *Describe architecture, machine/system name, and problem size used for the scaling plots*
- *Current job profile (independent jobs, chained jobs, workflow, etc.)*

Important: *please take into account the corresponding technical guidelines and requirements (e.g. required minimal code scalability, memory restictions, etc.) of the machine you have chosen!*
?.

Table 1: Scaling behavior

#cores	absolute timing (s)	speedup
3	600	1.0
6	600	2.0
24	600	8.0
48	600	16.0
96	600	32.0
192	600	64.0
384	600	128.0
768	600	256.0
1536	600	512.0
3000	600	1000.0

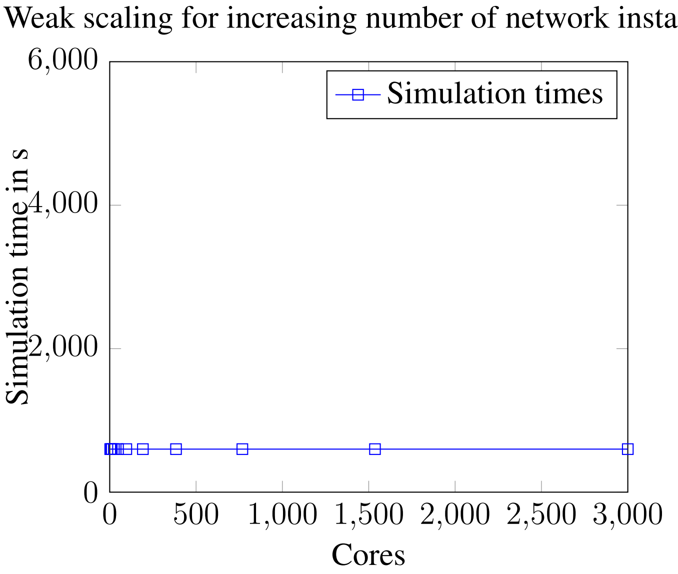


Figure 1: Weak scaling for increasing number of network instances with different parameters on increasing number of cores. Each instance of the network is independent from each other, thus having a perfect linear scaling.

5.2 Justification of resources requested

Sub-project	Size (neurons)	nodes /run JuropaTest	core-h /run	# runs	core-h
Sub-project 1					
100 neuron network	$1 \cdot 10^2$	125 (3000 cores)	3000*5=15000	300	4,500,000

Total requested time on JURECA: 4,500,000 core-h
125 cores/run will be used.

6 Resource Management and Work Schedule

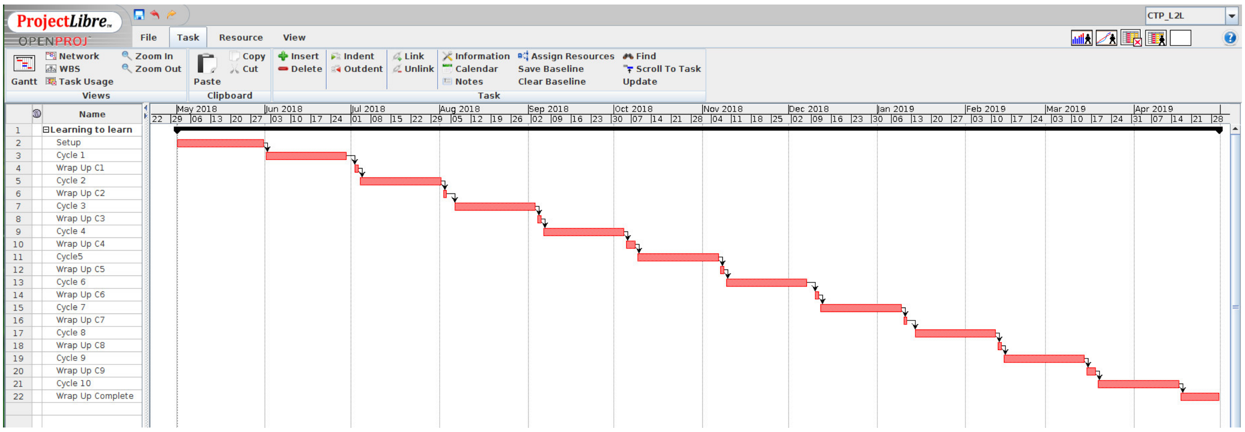
6.1 Resource management

Describe how you intend to manage the resources you have requested. This should include a description of the methods you will deploy to monitor progress of the project and how project results are documented.

(0.5 to 1 page)

6.2 Work schedule

Gantt chart for the project Learning to Learn



	Name
1	⊟Learning to learn
2	Setup
3	Cycle 1
4	Wrap Up C1
5	Cycle 2
6	Wrap Up C2
7	Cycle 3
8	Wrap Up C3
9	Cycle 4
10	Wrap Up C4
11	Cycle5
12	Wrap Up C5
13	Cycle 6
14	Wrap Up C6
15	Cycle 7
16	Wrap Up C7
17	Cycle 8
18	Wrap Up C8
19	Cycle 9
20	Wrap Up C9
21	Cycle 10
22	Wrap Up Complete

Figure 2: Work schedule for the project.

7 Key Personnel and Experiences

Give a short introduction of the key persons involved in the project and their experience (max 3 persons).

(half a page)

8 Bibliographic References

V1.4-2017JUL05