The VINEYARD framework for heterogeneous cloud applications: The BrainFrame case

Harry Sidiropoulos
School of ECE
Institute of Communication and Computer Systems, ICCS
Athens, Greece
harry@microlab.ntua.gr

George Chatzikonstantis
School of ECE
Institute of Communication and Computer Systems, ICCS
Athens, Greece
georgec@microlab.ntua.gr

Dimitrios Soudris
School of ECE
Institute of Communication and Computer Systems, ICCS
Athens, Greece
dsoudris@microlab.ntua.gr

Christos Strydis
Erasmus MC
University Medical Center Rotterdam,
Department of Neuroscience,
Rotterdam, The Netherlands
c.strydis@erasmusmc.nl

***Abstract*—Emerging cloud applications like machine learning, AI, big data analytics and scientific computing require high-performance computing systems that can sustain the increased amount of data processing without consuming excessive power. To this end, many cloud operators have started deploying hardware accelerators, like GPUs and FPGAs, to increase the performance of computationally intensive tasks. However, increased performance, comes at a higher cost of increased programming complexity for utilizing these accelerators. VINEYARD has developed a versatile framework that allows the seamless deployment and utilization of heterogeneous accelerators in the cloud without increasing the programming complexity while offering the flexibility of software packages. This paper presents the main components that have been developed in the VINEYARD framework and focuses on BrainFrame, the neurocomputing case that demonstrates the new framework's value. BrainFrame not only accelerates neuronal simulations but also has an architecture that allows easy access to neuroscientists, hiding the system complexity, and enabling a modular integration of new accelerated simulators.**

***Index Terms*—reconfigurable computing, hardware accelerators, FPGAs, neurocomputing, cloud computing**

I. Introduction

As the traffic in data-centers, continues to increase rapidly, data-center operators are looking for novel systems that can provide higher performance than typical processors without consuming excessive amounts of power. In the domain of embedded systems, vendors and designers have embraced the heterogeneity paradigm in order to provide high performance systems that are also energy-efficient. General Purpose processors (high performance and low power) are used to provide flexibility and support software stacks, while specialized co-processors are used to offload these processors for the most widely used tasks such as encryption, compression, and signal processing. Currently, it seems that cloud computing and data-center operators are beginning to embrace the advantages of heterogeneity in order to provide high-performance and energy-efficient systems [1] [2].

Data-center operators, in the last few years, introduced general-purpose GPUs (GP-GPUs) in order to provide users with high-performance systems. Lately they are beginning to deploy also FPGAs and they are looking at ways to allow the cloud users to utilize the performance of these specialized systems.

Amazon and Nimbix were among the first cloud providers that allowed the instantiating and utilization of FPGAs and GPUs in the cloud by cloud users. Currently other cloud operators like Huawei, Baidu and Alibaba offer to the users the possibility to rent and deploy FPGAs in order to speedup their applications.

In the domain of research, VINEYARD has introduced a digital marketplace for open-source modules for the research community, called Accel-store [3]. Companies like InAccel, rENIAC and Falcon Computing develop FPGA modules that can deployed in FPGAs that are hosted by cloud operators like Amazon AWS, Alibaba Cloud and Huawei. Accelize and Amazon already provide marketplaces for these accelerated modules. Cloud developers can browse the marketplaces and rent the accelerated modules in the form of IP cores. Then the cloud users can select a data center to deploy the accelerators based on cost, availability etc. All of these companies provide a wide range of IP cores. For example, InAccel provides ready-to-use FPGA modules for machine-learning applications like logistic regression, k-means clustering and recommendation engines. The use of a library-based approach can be used to foster the widespread adoption of hardware accelerators in the cloud. The preferred method for utilizing these accelerators is the deployment of easy-to-use hardware modules that can offload tasks from the typical processor without changes in the original code.

In this paper we present the VINEYARD approach towards a unified, integrated framework that allows the seamless utilization of heterogeneous accelerators in the Cloud. Section 2 outlines the VINEYARD approach. Section 3 provides details of the neurocomputing use-case: BrainFrame. Finally, Section 4 gives an overview of the main results stemming from our performance evaluation regarding VINEYARD project.

Project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 687628.

978-1-5386-8237-1/18/$31.00 ©2018 IEEE

II. The VINEYARD approach

A. VINEYARD Integrated Framework

VINEYARD, an EC-funded project, aims to allow cloud users to easily utilize accelerators (Xilinx FPGAs, Maxeler dataflow engines and Intel Xeon Phis) in heterogeneous data centers in a manner similar to software packages and with the same flexibility as any other cloud services. VINEYARD provides an integrated framework that hides from the user well-known hardware-accelerator drawbacks, such as resourcing, scheduling, programming and utilization, thus significantly simplifying FPGA deployment. Figure 1 depicts a high-level overview of the VINEYARD framework.

Applications that are targeting heterogeneous data centers with contemporary servers are programmed using typical distributed programming frameworks, such as Spark [11], or more application-specific frameworks such as the PyNN framework that is used for neural networks [12]. In these applications, VINEYARD provides the required APIs that enable the utilization of heterogeneous infrastructures without any other modifications in the source code. Some of the tasks such as sorting of data, encryption, compression, pattern matching, are extremely computationally intensive. These tasks have been implemented in hardware as customized intellectual-property (IP) accelerators that can achieve much higher performance with lower power consumption. These hardware accelerators are stored in an IP repository (Accel-Store) that interface with the VINEYARD resource manager and scheduler. For each application there is a range of versions based on the available accelerator platform (FPGA, DFE, and Xeon Phi).

To interface with hardware accelerators, vendor-specific libraries are used for low-level communication with hardware resources. (e.g. Xilinx's SDAccel and Intel's OPAE library [5]). On top of these interfaces, VINEYARD has developed accelerator-specific drivers that are required for communication with vendor-specific libraries. On top of the accelerator drivers, VINEYARD has developed the VINEYARD controller (VineController) that allows the abstraction of accelerator drivers from vendor-specific libraries and the utilization of these accelerators from high-level programming frameworks 1.

For the cloud-computing applications, the software stack of each node contains the VMs that are running on the processor, the Local scheduler that dispatches the job to the local accelerators, VineTalk layer that allows the virtualization of the underlying hardware resources, and VineController that serializes the jobs to the hardware resources. Applications can either use directly VineTalk [6] and VineController for the utilization of the resources or through the use of a central scheduler. In the first case, multiple applications can share the resources of a single accelerator through the virtualization of resources (i.e. multi-tenancy). The scheduler and the resource manager are used when these applications want to access several heterogeneous infrastructures. In that case, VineTalk can be used optionally if several applications want to share the hardware resources. On top of the VINEYARD applications, a web-based GUI can be used to allow control and utilization of the available resources. For example, in case of neurocomputing applications, BrainFrame [8] has been developed that allows the utilization of the PyNN framework via VINEYARD's heterogeneous resources. In cases of cloud-computing applications that use programming frameworks like Spark, the required APIs have been developed for allowing the utilization of heterogeneous resources both at the scheduler and the resource-manager level. The resource manager allocates resources from the heterogeneous platform and dispatches jobs to these nodes based on the application requirements. The resource-manager tracks the information about the status of these accelerators. The resource-manager also communicates with an accelerator library controller that is used to fetch and dispatch the right hardware accelerator from the Accel-Store IP library based on the available resources (FPGA, DFE, or Xeon Phi, based servers).

B. Accelerator Repository

To allow the widespread deployment of hardware accelerators in the cloud, and support an ecosystem-based approach, VINEYARD has adopted a library-based approach that will decouple developers of the accelerator engines form the cloud users. Towards this end, VINEYARD has released an open-source repository for the IP cores that can be used in FPGA, dataflow engines (Maxeler), GPUs and Compute engines deployed in cloud-computing systems [3]. The repository currently contains several IP cores for applications like neurocomputing, machine-learning and financial applications. The repository supports platforms like dataflow engines from Maxeler, SDAccel from Xilinx and Amazon AWS. The repository can either contain just the configuration file for the specific platform (i.e. bitstream) or both the configuration file and the source files for the IP cores. The repository is open to the research community that is able to upload their accelerator specific IP cores on the cloud. The current repository contains the following accelerators for four main application classes:

1) Neurocomputing
   - Inferior-Olive spiking neuron simulator
   - Ordinary Differential Equations (ODE) solvers
2) Financial
   - Black and Scholes algorithm
   - Black77 algorithm
   - Binomial
3) Machine Learning
   - Logistic Regression
   - K-means clustering
4) Data Management
   - Compression

III. BrainFrame use-case

A. BrainFrame as an evaluation workload

The integrated VINEYARD framework has been evaluated under three real-life workloads and industrial benchmarks for financial applications, data management, and neurocomputing applications.

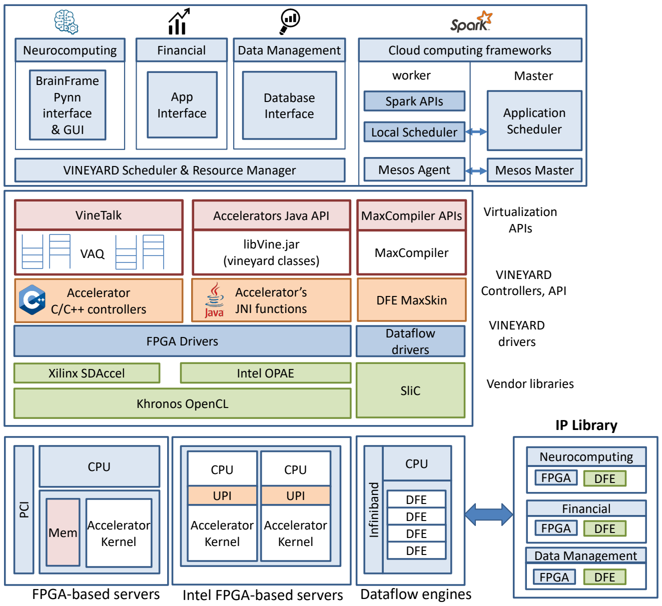


Fig. 1. High-level overview of the VINEYARD framework and the accelerator platforms.

The neurocomputing workload evaluated lies in the domain of scientific computing, and more specifically in the domain of computational neuroscience which aims at better understanding the working of the human brain through the use of mathematical models of biological neural networks [4]. This particular application is a high-performance, high-accuracy simulation of the Inferior-Olivary nucleus of the brain and is simply called the Inferior-Olive application. The Olivocerebellar system is critical in facilitating motor function in humans. Better modeling and understanding of its function can lead to major breakthroughs in the treatment of cerebellum-related degenerative diseases (such as autism, fragile-X syndrome etc.).

As is shown in Figure 2, the neurocomputing application use case of VINEYARD has the highest throughput requirements but not so strict latency constrains.

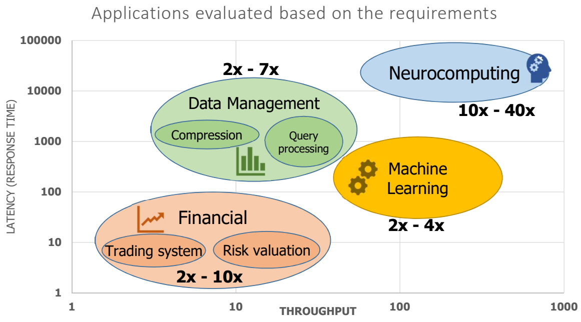


Fig. 2. Accelerated VINEYARD applications grouped by domain and requirements (throughput-latency).

B. *A case for hardware heterogeneity in Computational Neuroscience*

Through the efforts of biologists and computational neuroscientists in recent decades, advance models of biological neurons were developed using Spiking Neural Networks (SNNs) [7]. These models do not abstractly capture aspects of biological processes, like Artificial Neural Networks (ANNs), but directly emulate them.

A major challenge is the sheer computational complexity that many SNN models entail, compared to simpler modeling classes. Even the simple types of SNNs have significant demands as the studied neuronal network increases in size and density both in terms of computation, but also data transfer and storage.

The neuroscience community uses a large number of models in order to simulate biophysically accurate biological counterparts. Some of these models can be CPU-bound, whilst other models might require high communication throughput. Therefore, a single HPC fabric -whether hardware or software-cannot cover all possible cases without sacrificing performance for generality.

Even in the same neuron model, changing the characteristics of the simulation, like neuron network size and density of the network, has different effects on the performance of each HPC accelerator fabric. This has been presented in [8] where a state-of-the-art, extended Hodgkin-Huxley neuron model of the inferior-olivary nucleus was used as a benchmark to evaluate the BrainFrame framework. This worked as a proof-of-concept of the need for heterogeneous fabrics ;there was no single accelerator fabric, that offered best performance across all model instances.

Furthermore there are different types of neuroscientific experimentation that can help expand the research on the human brain. There is the study of the behavior of very large networks, like in [9], where using the neuromorphic hardware SpiNNaker the researchers simulated 80,000 neurons and 0.3 billion synapses. There is also the need of exploring the parameter space of neuron and synaptic models which requires a large number simulations of smaller neuronal networks. Lastly, real-time experimentation with complex, biologically accurate neuron models, where the researcher could probe and stimulate the network as if it was a real part of the brain would significantly push the boundaries of neuroscientific research. All of the aforementioned experimentation methods present diverse computational and I/O workloads that cannot be simulated effectively (or even at all) in a simple homogeneous computational system.

To summarize in the field of computational neuroscience, experimentation comprises of really diverse workloads due to:

- A large array of neuron and synapse models, neuroscientists use, that have different computational complexities.
- Different experimentation methods i.e. Large networks dynamics, parameter space exploration, real-time response, etc.
- Simulation characteristics and parameters like neuronal network size, density, neuroplasticity, etc.

The BrainFrame approach is to provide scientists with an acceleration platform that has the ability to adjust to the aforementioned variety of workload characteristics. A heterogeneous system that integrates multiple HPC technologies, instead of just one, would be able to provide this. In addition, a framework for a heterogeneous system using a popular user interface for all integrated technologies can also provide the ability to select a different accelerator, depending on availability, cost and performance desired. Such a hardware back-end must overcome additional challenges to be used in the field. It requires a front-end which should provide two crucial features:

- An easy and commonly used interface through which neuroscientists can employ the platform, without the constant mediation of an engineer.
- A programming interface that can reuse the vast amount of models already available to the community.

C. Brainframe architecture

BrainFrame, in order to provide an easy to use interface to neuroscientists, is being developed as a web service. The main actions the user can take in this web interface are three:

1) Upload simulation scripts.
2) Select the simulator and the accelerator type that wants to use (there is also the option of letting the resource manager automatically select the best resource available).
3) Run experiments and choose how the results are reported.

The implementation of BrainFrame architecture is shown in Figure 3. When a user requests a new simulation, the system will retrieve information about utilization of the accelerated and non-accelerated platforms. It will then based on the set of simulation parameters and utilization information pick the best out of the available machines, create a corresponding simulation script and starts the simulation. This framework is easily extensible to support new neuronal models and accelerated simulators. Furthermore the system can be scaled elastically on a cloud infrastructure such that if none of the systems are available to run the simulation it will be then executed on one of the cloud providers (AWS, Azure, Google Cloud, etc.) provided the user owns a BrainFrame account and sufficient credit. Once the simulation finishes on one of the systems the information about the status will be sent back to the user via the interface host and all associated information (like simulation results).

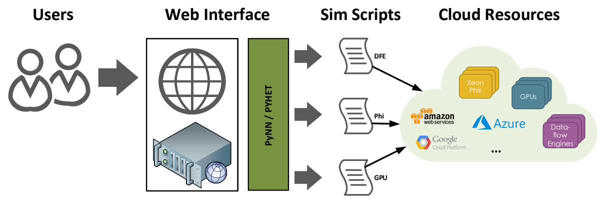
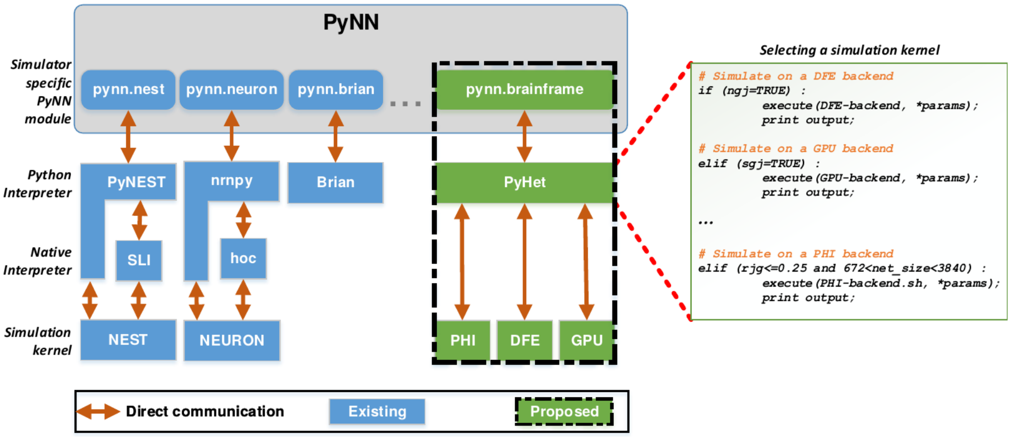


Fig. 3. This is a high level overview of the BrainFrame architecture.

As a language for the simulation scripts we selected PyNN, a widely known and used framework by computational neuroscientists. PyNN is a simulator-independent language for building neuronal network-models. The PyNN API aims to support modeling at a high-level of abstraction (populations of neurons, layers, columns and the connections between them) while still allowing access to the details of individual neurons and synapses when required. PyNN provides a library of standard neuron, synapse and synaptic-plasticity models, which have been verified to work in an identical fashion on different simulators.

In order to use PyNN with our accelerated simulators, we developed a new PyNN backend, pynn.brainframe. Figure 4 depicts how pynn.brainframe fits in the existing PyNN structure. With the existing approach, each simulation platform uses a custom-built interpreter to run models. Brainframe is using an intermediate layer called `PyHet` that is able to ’translate’ the simulation options and elements to our custom accelerated implementation, thus providing compatibility with PyNN and the other simulators. As a part of this work, we have targeted development of three heterogeneous back-ends: Intel Xeon Phi, Maxeler DFE and NVidia GPGPU.

The common method for extending PyNN with new back-end simulators, like BrainFrame, was to let PyNN create the neuronal-network and then call the selected simulator to execute the simulation. We disabled this function and decided to translate PyNN simulation parameters to our custom accelerated simulator parameters. Although this was more complex than sending a PyNN exported neuronal network, we opted for the script due to size and network delays. Some large experiments may need GB of data to describe the neuronal



Selecting a simulation kernel

```
# Simulate on a DFE backend
if (ngj=TRUE) :
         execute(DFE-backend, *params);
         print output;

# Simulate on a GPU backend
elif (sgj=TRUE) :
         execute(GPU-backend, *params);
         print output;

...

# Simulate on a PHI backend
elif (rjg<=0.25 and 672<net_size<3840) :
         execute(PHI-backend.sh, *params);
         print output;
```

Fig. 4. PyNN front-end used to implement BrainFrame tool-flow

network so it is faster to create the network locally on the accelerator fabric than send it over network.

D. Experimental results

Our vision for the BrainFrame platform is to help the neuroscientific community further the knowledge for the human brain. Towards this end, as mentioned previously, the road is not paved with a single neuron model or experimentation method but with a plethora of neuron models and diverse experimentation. We aspire to create a modular system that is supported by an active community of computer scientists that constantly develop new accelerated simulators for new or already existing models and a corresponding active community of computational neuroscientists that experiments with those simulators.

For this reason we wanted to experiment with diverse HW resources, different simulators and different simulation characteristics and showcase that way the benefit of the BrainFrame framework.

As a first evaluation we showed in [8] that simulating Inferior olivary nucleus neurons with different simulation parameters resulted in diverse performance results. Figure 5 shows the selection for our use-case instances after performance analysis. On the left, (y axis), the RGJ, NGJ and SGJ represent the different neuronal-connectivity simulation options while on the x axis is the number of neurons simulated. The RGJ case selection, which presents the most complex case in terms of accelerator choice, shifts between all three options depending on the connectivity density (25, 50, 70 and 100%). For example, for RGJ 100% and less than 4800 neurons the optimal HW platform for simulation is the Maxeler DFE, but for simulating more than 4800 neurons a GPU is a better choice. For the SGJ case, (a simpler case than RGJ) the GPU is always the accelerator of choice, while for the NGJ case (the simplest of all cases, with no neuron interconnectivity at all) the DFE yields optimal results under all experiment parameters. Lastly, if the experiment is flagged as a real-time experiment, the algorithm exclusively chooses the DFE to accelerate the application, as it is the only clearly viable accelerator for real-time experiments.

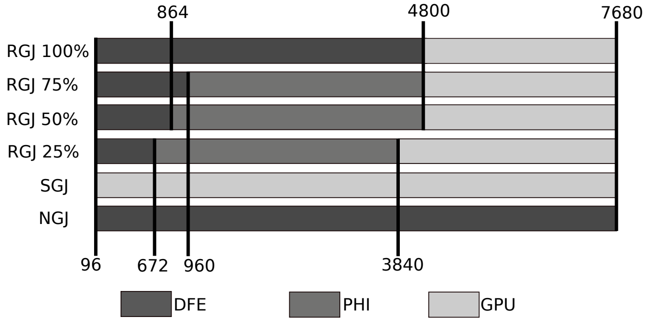


Fig. 5. BrainFrame accelerator-selection map on the Inferior olive simulator. Selection is heavily dependent on the experiment, involving all three accelerator fabrics.

In order to also support large-scale experiments we are working on a version of the aforementioned simulator that operates on multiple accelerator nodes. Early results show that BrainFrame using the VINEYARD heterogeneous cloud can simulate lighting fast even large-scale networks provided the availability of the accelerator fabric. Figure 6 shows some early results in experimenting with scaling over multiple accelerator nodes, in this case Intel Xeon Phi, KNL edition, to support large-scale simulations. Although this is an early result and we are currently studying more optimization and experimentation scenarios, BrainFrame, using the VINEYARD heterogeneous cloud achieved a 4.2× speedup simulating 100,000 neurons with 100,000,000 synapses for 100 ms of brain simulated time in 5 seconds (5000 ms).

IV. Conclusions

This paper has presented the main overview of the VINEYARD integrated framework that aims to facilitate the utilization of hardware accelerators in the cloud and specifically

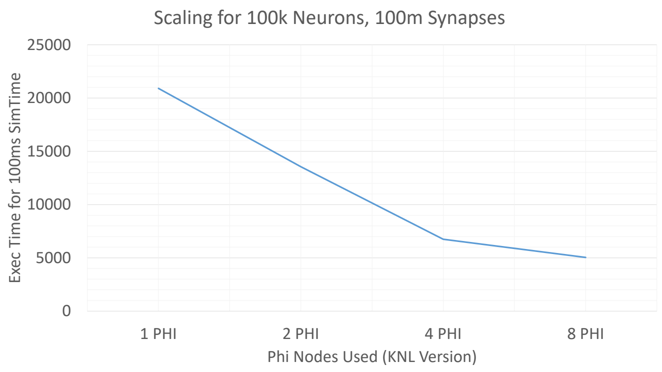


Fig. 6. BrainFrame accelerating a large scale simulation of 100k neurons with 100 million synapses in 1,2,4 and 8 Intel Xeon Phi KNL nodes [10]. Execution time is measured in ms per simulated 100ms of brain activity.

zooms in the BrainFrame neuroscientific use-case. We provided proof of the need for a heterogeneous cloud consisting of different accelerator fabrics by showcasing the diverse needs of the computational-neuroscience domain. With the use of these accelerators we can speedup significantly the performance of the applications and at the same time to reduce the total cost of ownership and the power consumption in the data centers. Also, the use of the VINEYARD repository aims to decouple cloud users from the accelerators' developers and build an ecosystem thus facilitating the use of accelerators in the cloud-computing community. The performance evaluation of the BrainFrame real use-case shows the main advantage of the adoption of the hardware accelerators in the cloud.

References

[1] S. Byma, J.G. Steffan, H. Bannazadeh, A. Leon-Garcia, and P. Chow. 2014. "FPGAs in the Cloud: Booting Virtualized Hardware Accelerators with OpenStack." In Field-Programmable Custom Computing Machines (FCCM), 2014 IEEE 22nd Annual International Symposium on. 109–116. https://doi.org/10.1109/FCCM.2014.42

[2] C. Kachris and D. Soudris. 2016. "A survey on reconfigurable accelerators for cloud computing." In 2016 26th International Conference on Field Programmable Logic and Applications (FPL). 1–10. https://doi.org/10.1109/FPL.2016.7577381

[3] VINEYARD Aceelerator repository http://www.accel-store.com/

[4] George Chatzikonstantis, Diego Jiménez, Esteban Meneses, Christos Strydis, Harry Sidiropoulos, and Dimitrios Soudris. 2017. "From Knights Corner to Landing: A Case Study Based on a Hodgkin-Huxley Neuron Simulator." In International Conference on High Performance Computing. Springer, 363–375.

[5] Intel. [n. d.]. Intel Open Programmable Acceleration Engine (OPAE),. https: //01.org/OPAE

[6] Stelios Mavridis, Manolis Pavlidakis, Ioannis Stamoulias, Christos Kozanitis, Nikolaos Chrysos, Christoforos Kachris, Dimitrios Soudris, and Angelos Bilas. 2017. "VineTalk: Simplifying software access and sharing of FPGAs in datacenters." In Field Programmable Logic and Applications (FPL), 2017 27th International Conference on. IEEE, 1–4.

[7] G. Wulfram and W. Werner, "Spiking Neuron Models". Cambridge University Press, 2002.

[8] Georgios Smaragdos, Georgios Chatzikonstantis, Rahul Kukreja, Harry Sidiropoulos, Dimitrios Rodopoulos, Ioannis Sourdis, Zaid Al-Ars, Christoforos Kachris, Dimitrios Soudris, Chris I De Zeeuw, et al. 2017. BrainFrame: a node-level heterogeneous accelerator platform for neuron simulations. Journal of neural engineering 14, 6 (2017), 066008.

[9] van Albada Sacha J., Rowley Andrew G., Senk Johanna, Hopkins Michael, Schmidt Maximilian, Stokes Alan B., Lester David R., Diesmann Markus, Furber Steve B., "Performance Comparison of the Digital Neuromorphic Hardware SpiNNaker and the Neural Network Simulation Software NEST for a Full-Scale Cortical Microcircuit Model", Frontiers in Neuroscience, vol 12, 2018.

[10] Jeffers, J., Reinders, J., Sodani, A.: "Intel Xeon Phi Processor High Performance Programming: Knights Landing Edition". Morgan Kaufmann, Boston (2016).

[11] Apache Spark, a unified analytics engine for large-scale data processing. https://spark.apache.org

[12] PyNN, a simulator-independent language for building neuronal network models. http://neuralensemble.org/PyNN/