Massively Parallel Robot Simulations with the HBP Neurorobotics Platform

Florian Walter, Mahmoud Akl, Fabrice O. Morin, Alois Knoll
Chair of Robotics, Artificial Intelligence and Real-Time Systems, Technical University of Munich
{walter, akl, morinf, knoll}@in.tum.de

Abstract

The success of deep learning in robotics hinges on the availability of physically accurate virtual training environments and simulation tools that accelerate learning by scaling to many parallel instances. However, most current AI frameworks do not integrate easily with common software stacks from robotics, while fully-fledged robot simulators lack capabilities for parallelization. In this paper, we introduce an extension for the Neurorobotics Platform of the Human Brain Project (HBP) that offers the full feature set of a robot simulator and at the same time is arbitrarily scalable for massively parallel robotics experiments.

1. Introduction

With the widespread adoption of deep learning in robotics, training environments have gained increasing relevance because the experience of an embodied agent can only be obtained through active interaction with the world around it. Gathering the required data on physical robots is costly, slow, and potentially unsafe, which makes simulators essential. In particular, research on deep reinforcement learning has recently fostered the development of virtual training environments for challenging tasks ranging from computer games to object manipulation. They are typically centered around ready-to-use setups with lightweight programming interfaces that are tailored to the requirements of machine learning applications. Some of them can even spawn multiple instances to parallelize training. Often missing, however, are key features of robot simulators such as modeling tools for creating new setups, extensions for different sensor models, and interfaces for commonly used tools in robotics. Even though some training environments support a subset of these features such as URDF and ROS interfaces, none of them meets all needs of robotics research. Conversely, current robot simulators offer only limited support for machine learning applications, especially in terms of parallelization.

In the following, we present a new cloud-based simulation framework that combines the ease of use and scalability of virtual training environments from machine learning with the full feature set of a robot simulator. It is based on the *Neurorobotics Platform (NRP)* [2] developed in the HBP [1] and supports massively parallel simulations for applications in AI and neuroscience on both cloud and high performance computing platforms. Section 2 outlines the NRP's core features and system architecture. In section 3, we introduce an extension for massively parallel distributed simulations that has been successfully applied to a robot grasping setup described in section 4. Section 5 concludes the text and provides an outlook to future work.

2. The HBP Neurorobotics Platform

The NRP is a cloud-based simulation framework for neuroscience, neuromorphic engineering, robotics and embodied AI. At its core is the so-called *Closed-Loop Engine (CLE)* that connects a physics-based environment simulation to any type of algorithm or cognitive model, ranging from data-driven brain simulations for virtual neuroscience to artificial neural networks for deep reinforcement learning. The current version (3.1) of the NRP is based on the open-source robot simulator Gazebo, but future releases will provide generic interfaces for easily integrating other simulation engines. There is further built-in support for spiking neural network simulators such as NEST and Nengo. Deep learning tool kits like TensorFlow or PyTorch can be directly accessed through their Python-based APIs. This makes the NRP one of the only tools that combines a fully-fledged robot simulator with broad support for various machine learning frameworks.

A unique feature of the NRP is that its system architecture was designed from the ground up to support deployment in the cloud and high performance computing environments. It is split into a front end that manages multiple back ends where the actual simulations are running. Both components are packaged as Docker images for seamless deployment. Users can access the front end through a web interface or from Python to create, manage, visualize and evaluate *experiments*, i.e. full specifications of simulation tasks including robots, environments, models and algorithms. Active experiments run independently from each

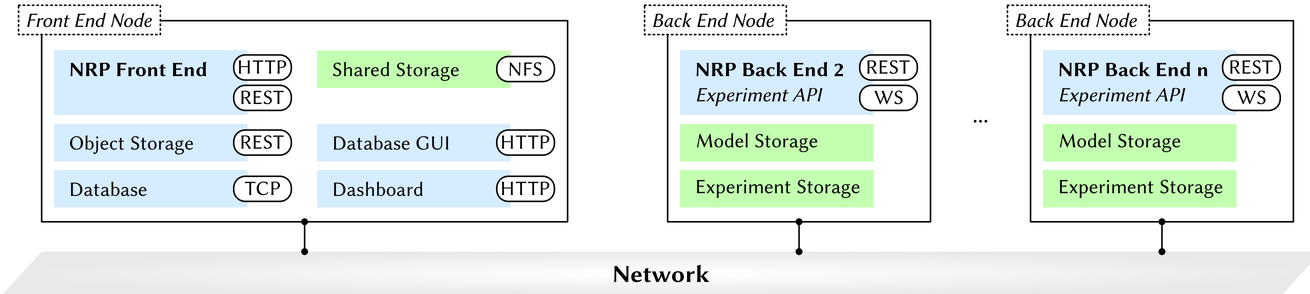

Figure 1. Extended NRP system architecture for massively parallel data collection and training.

other on the back ends, which is an important prerequisite for parallelization.

Its cloud-optimized architecture makes the NRP distinct from *Gym-Ignition* [3] and *RLBench* [4], which also provide virtual training environments based on robot simulators but no built-in cloud support. The web interfaces and services implemented by the front end and back end components also go beyond the feature set of *AWS RoboMaker*, which only provides Gazebo's standard user interface and is limited to AWS clusters.

3. Massively Parallel NRP Experiments

The NRP's default architecture outlined above supports the execution of an arbitrary number of experiments in parallel but does not provide a common context for data sharing and interactions between them. To enable the accelerated collection of data sets on many concurrent simulations or distributed training with possibly hundreds of workers, we have extended the NRP as depicted in figure 1. In addition to the standard NRP Docker images for front end and back end, there are additional services for central back end-accessible data storage (object storage, database) and analysis (database GUI, dashboard). The data base is used to both distribute experiment parameters to all back ends and to store meta information of e.g. training samples that reside in the object storage. If required, simulation models and experiment definitions can be stored locally on every back end to avoid bottlenecks.

4. A Task Environment for Robot Grasping

We have created an experiment setup for robot grasping tasks based on [5] to evaluate the NRP extension. It is comprised of a KUKA LBR iiwa robot model placed in front of a tray with space for objects and an RGB camera that captures the workspace. Robot control is implemented using common tools from the ROS software stack (ros_control, MoveIt). Like in other virtual training environments, users can connect their machine learning models through a new high-level *Experiment API*. Besides gathering data and actuating the robot, it also supports loading random user-defined objects, any number of which can be placed anywhere in the scene. The data collected from this setup can, for example, be used to train fully data-driven controllers for grasping such as the one proposed in [5]. Importantly, however, the proposed extension is not limited to this use case and can be applied to any experiment running in the NRP as long as it implements the required interfaces.

We have successfully applied the new NRP extension to parallelize the sampling of robot grasps from this experiment. Our tests were running on an OpenStack cluster with more than 100 back end instances and we expect that the system will also scale to considerably larger installations. While sampling performance for a single simulation depends on the selection of objects in the tray, the overall speedup can be considered linear during the collection of random grasp attempts where all simulations run asynchronously. Scaling is in essence only limited by the performance of the central data storage. Tests on a high performance computer are already in preparation.

5. Conclusion and Outlook

We have introduced an extension to the NRP that enables massively parallel experiments with a full-featured robot simulator. To our knowledge, the NRP is the only virtual training environment with a cloud-optimized architecture and support for deployment on high performance computers. In this work, our focus was on the accelerated collection of synthetic data sets. In the future, we plan to add services for distributed reinforcement learning algorithms and to include support for alternative environment simulators that will be provided by upcoming NRP releases. The NRP is an open-source project and available on neurorobotics.ai.

Acknowledgments

This research has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 945539 (Human Brain Project SGA3).

References

[1] Katrin Amunts, Alois C. Knoll, Thomas Lippert, Cyriel M. A. Pennartz, Philippe Ryvlin, Alain Destexhe, Viktor K. Jirsa, Egidio D'Angelo, and Jan G. Bjaalie. The Human Brain Project—Synergy between neuroscience, computing, informatics, and brain-inspired technologies. *PLOS Biology*, 17(7):e3000344, 2019. 1

[2] Egidio Falotico, Lorenzo Vannucci, Alessandro Ambrosano, Ugo Albanese, Stefan Ulbrich, Juan Camilo Vasquez Tieck, Georg Hinkel, Jacques Kaiser, Igor Peric, Oliver Denninger, Nino Cauli, Murat Kirtay, Arne Roennau, Gudrun Klinker, Axel von Arnim, Luc Guyot, Daniel Peppicelli, Pablo Martínez-Cañada, Eduardo Ros, Patrick Maier, Sandro Weber, Manuel Huber, David Plecher, Florian Röhrbein, Stefan Deser, Alina Roitberg, Patrick van der Smagt, Rüdiger Dillman, Paul Levi, Cecilia Laschi, Alois C. Knoll, and Marc-Oliver Gewaltig. Connecting Artificial Brains to Robots in a Comprehensive Simulation Framework: The Neurorobotics Platform. *Frontiers in Neurorobotics*, 11:2, 2017. 1

[3] Diego Ferigo, Silvio Traversaro, Giorgio Metta, and Daniele Pucci. Gym-Ignition: Reproducible Robotic Simulations for Reinforcement Learning. In *2020 IEEE/SICE International Symposium on System Integration*, pages 885–890, 2020. 2

[4] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The Robot Learning Benchmark Learning Environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 2

[5] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2017. 2