NEST Progress Report

Jochen Martin Eppler
<eppler@biologie.uni-freiburg.de>

FACETS CodeJam III
Timeline of new features

- Testsuite reorganization
- New init/reset framework
- Compilation on BlueGene
- Adaptation to new API
- Support for distributed tests
- Improved connection handling
- Replaced corrupted files ;-)
- New topology module

Changes (lines of code)

SVN revision
General improvements and numbers

- Since the last CodeJam (05/2008), we had 1217 repository checkins
- We added 84357 lines and removed 22291 lines of code
- 19+5 people contributed code to NEST between r7226 and r8441
- There are 105 subscribers on the nest_user mailing list
- We have a new set of logos!

The homepage will be updated soon...
- Parser for infix math notation
  
  \[(1+2*3) \text{ CompileMath } \Rightarrow \{1 \text{ 2 3 mul add}\} \]
  \[(1+2*3) \text{ ExecMath } \Rightarrow 7\]

- Support for efficient functions

  /f \[/x /y\] (1+x*y) Function def
  2 3 f \Rightarrow 7

- Support for function inlining

  \{1 2 number mul add\} << /number 3 >> Inline
  \Rightarrow \{1 2 3 mul add\}

- Long info/warning/error messages are more readable

- New commands \texttt{apropos} and \texttt{which} to make finding help easier

- New command \texttt{LambertW}, which is the inverse function of
  \[x = W*\exp(W)\]

- Commandline switch \texttt{-c} for NEST to execute SLI code directly
PyNEST improvements

- We cleaned up the API and improved the installation procedure
- The new API is published as
  Eppler JM, Helias M, Muller E, Diesmann M, and Gewaltig M-O
  PyNEST: A convenient interface to the NEST simulator (2008)

- Python now has its own unit tests, which are run during automatically
  by `make installcheck` or using `nest.test()`
- Error messages now contain the complete information from SLI
- The data conversion now understands array selections (`a[:,1]`) and array scalars and treats all numpy integer types equally
- `sli_func()` eases the use of SLI functions in PyNEST:
  
  ```python
  x = sli_func("add", 1, 2)
  ```
New models and model related improvements

- The new **multimeter** allows to record arbitrary state variables from neurons. However, only few models support this so far.
- **iaf_cond_alpha_mc** is a simple three-compartment model, which is a good example on how to use different receptor types
- **pulsepacket_generator** now generates a different pulse for each target and supports overlapping pulse packets
- The update function of **sli_neuron** can be written in SLI instead of C++ and allows easy prototyping of new neuron models

- The new init/reset framework for nodes allows to carry out multiple experiments on the same model without completely rebuilding

- A video tutorial explains how to write own models for NEST
  
  http://arken.umb.no/~plesser/iaf_cond_alpha.mov
Improved connection handling

- Until recently, retrieval and setting of connection parameters was problematic in multi-threaded setups
- We changed the function **FindConnections** to return a list of connection ids for the outgoing connections

```python
n1, n2 = Create('iaf_neuron', 2)
Connect([n1], [n2])
c = FindConnections([n1])
```

- Connection ids can be used with **SetStatus** and **GetStatus** like global ids of nodes

```python
SetStatus(c, 'weight', 2.5)
print GetStatus(c)
```

```python
[{'synapse_type': 'static_synapse',
  'target': 2, 'weight': 2.5, 'delay': 1.0,
  'source': 1, 'receptor': 0}]
```
Support for complex topologies was implemented as SLI library in the NEST 1.0.x release series.

We have completely reimplemented the topology module in C++.

In PyNEST, the topology module is available as `nest.topology`
The multi-simulator coordinator (MUSIC) is a standard and a library to allow applications the exchange of data during run time. This is important to bridge the gap between simulations on different scales.

Support for MUSIC has been added to NEST and MOOSE. The interface has been tested by coupling a detailed striatum model in MOOSE with a large cortical model in NEST.

We currently write an article about the MUSIC interface in MOOSE and NEST, which will be submitted to Neuroinformatics soon. The interface will be made available publically after the article has been published.

Still more simulators need to support MUSIC (talk to Mikael and me).
- We now have built-in support for the BlueGene architecture by using the configure switch `--enable-bluegene`
- This is necessary because the head node is set up very differently from the work nodes on these machines.

Random network of 100,000 excitatory and 25,000 inhibitory neurons and STDP simulated for 1 second (Brunel 2000)
Performance improvements

- We removed the CPEX algorithm in favor of `MPI_Allgather()`.
- This significantly improved performance and scaling with Infiniband interconnect when using OpenMPI.
The testsuite currently contains 124 test scripts


We now have the possibility to test the distributed version of NEST with the correct version of \texttt{mpirun}

Tests are structured (self tests, unit tests, regression tests and tests for MPI) and run in a hierarchy from simple to complex

Tests can now be written in SLI or in Python

The testsuite is run upon \texttt{make installcheck} or by using the SLI command \texttt{test}
For the summer school in Okinawa and the BCCN tutorials at the CNS, we created a LiveCD containing NEST, Python, NumPy, SciPy and Matplotlib.

At the CNS, I proposed to extend the CD to be a showcase of free software in Neuroscience and get users started more easily.

Candidates are NEURON, PCSIM, Brian, PyNN, NeuroTools, ...

I'm looking forward to discuss this with you during the CodeJam :-(
We are currently working on a possibility to store the architecture and state of networks to file. This allows to test new ideas more rapidly, especially for long-running simulations with plasticity.

We explore strategies to optimize network storage and communication on very large clusters (10k+).

We plan to implement SLI/PyNEST wrappers for inter process communication via MPI to allow a more flexible simulation control and easier verification of the network.

A public repository with read-only access to the source code of NEST will be evaluated during the CodeJam.
Acknowledgements

Kittel Austvoll
Andrew Davison
Moritz Deger
Markus Diesmann
Mikael Djurfeldt
Håkon Enger
Marc-Oliver Gewaltig
Alexander Hanuschkin
Moritz Helias
Susanne Kunkel
Rüdiger Kupper

Jens Kremkow
Eilif Muller
Abigail Morrison
Eilen Nordlie
Hans Ekkehard Plesser
Tobias Potjans
Wiebke Potjans
Sven Rebhan
Sven Schrader
Bernd Wiebelt
Pierre Yger

... thank you for your attention!