Spiking Neuron Networks
Distributed Event-Driven Simulations
with DAMNED

Anthony Mouraud
FACETS CodeJam Workshop

June 2010, 23rd
Simulation framework

- **Distributed**: allow large scale simulations
- **Event-driven**: sparse activities
- **Multithreaded**: computations and communications overlap

**DAMNED**: Distributed And Multithreaded Neural Event-Driven simulator

[PDCN’06], [NEUROCOMP’06]
DAMNED: Distributed And Multithreaded Neural Event-Driven simulator
Basic element: Event-driven Cell (EC)
DAMNED concepts

Basic element: Event-driven Cell (EC)

- Decentralized Global Virtual Time handling
- Mutexes on shared data structures
Simulation framework

DAMNED Simulator

- Neurons & synapses Models
- Networks Models
- Environment & stimulation protocols
- Hardware
Neuron models definition

**C++ classes inheritance**

- Neuron ← ObjetEvenementiel (EC)
- Synapses handled by neuron model
- Few methods definition
Neuron models definition

C++ classes inheritance

- Neuron ← ObjetEvenementiel (EC)
- Synapses handled by neuron model
- Few methods definition

```cpp
class Neurone : public ObjetEvenementiel {
public:
    Neurone();
    ~Neurone();
    Evenement* runRecepEvt(Evenement*);
    void ajouterConnexion(ObjetEvenementiel*);
    unsigned int getDelai(const Point4D&);
};
```
Neuron models definition

**C++ classes inheritance**
- Neuron \(\leftarrow\) ObjetEvenementiel (EC)
- Synapses handled by neuron model
- Few methods definition

**Event-driven constraint**
- Irregular updates
  - \(\rightarrow\) Predictions
  - \(\rightarrow\) Coupled ED

**Portability**
- Very few constraints on models
Synaptic connexions

Weights
- Post-synaptic handling
- Unsupervised local learning processes

Delays
- Applied post-synaptically
- Delay learning / shift
Simulation framework

DAMNED Simulator

Neurons & synapses
Models

Networks
Models

Environment & stimulation protocols

Hardware
## Network models

<table>
<thead>
<tr>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Populations set</td>
</tr>
<tr>
<td>Projections set</td>
</tr>
</tbody>
</table>
## Network models

### Network

- Populations set defining:
  - Neuron model
  - size
  - name, etc
- Projections set

### Inside API:

```java
Network net = createNewNetwork();
net.setNbPop( N );
net.setNomPop( 1, ”Excit 2D MAP” );
net.setNeuronModel( 1, ”NeuronTypeT” );
net.setTaillePop( 1, 100 );
...
```
Network models

Network

- Populations set
- Projections set defining:
  - Projection
  - Weight
  - Specific parameters

Inside API:

```java
int aproj = net.addProjection( 1, 2, "SomeProjType" );
net.setPoidsProj(aproj, 12);
net.setParamSpec(aproj, "paramName", 0.5 );
```
Simulation framework

- Neurons & synapses Models
- Networks Models
- Environment & stimulation protocols
- DAMNED Simulator
- Hardware

Protocols & stimulation Environment

Distributed SNN Models Networks

Hardware
Environnement models

Particular role: online interactions

- Applies stimulations (through input cells)
- Gets output activities (through output cells)
- May act outside simulation (e.g. robot command)

Environment is a full process

- A Thread running during simulation
- A specific (EC): CPC → run() method definition
Environnement models

Particular role: online interactions

- Applies stimulations (through input cells)
- Gets output activities (through output cells)
- May act outside simulation (e.g. robot command)

Environment is a full process

- A Thread running during simulation
- A specific (EC): CPC \rightarrow \texttt{run()} method definition

Inside API:

```java
setEnvModel("ModEnvironment");
net.setNbPopIn(2);
net.setNbPopOut(2);
```
Simulation framework

- Neurons & synapses
- Networks
- Environment & stimulation protocols

DAMNED Simulator

- Models
- Models
- Hardware

Distributed SNN
Models
Environments
Distributed hardware

- Mono or multi-core stations
- LAN, Clusters, Grids
- Parallel machines
Distributed hardware

Inside API:

```python
def setNbNodes(3);
def defHardware("FicHosts");
def addMappingRange(0, 0, 300);
def addMappingRange(1, 301, 2000);
def addMappingRange(2, 2001, 4999);
```
Large scale networks: execution time

- Stations: bi-cores 2Ghz, 1Go Ram
- Neuron model: LIF
- Nb neur.: $8 \times 10^4$ et $10^5$
- Nb syn.: $80 \times 10^6$ et $100 \times 10^6$
- Duration: 1 s
- Stim. freq.: 600 Hz
- Average activity: 1 Hz
Large scale networks: speedups

- Speedup: \( \frac{T_{\text{Sequential}}}{T_{\text{Parallel}}} \)
- Extrapolated sequential time

[EANN’09]
Saccade Burst Generator

- GATE
- cMRF
- DUREE
- OPN
- HAUT
- BAS
- DROITE
- EBN
Modèle élaboré
Simulator dependencies

- make
- C++ compiler
- libdl (dynamic library load)
- Posix threads
- MPI (MPICH2)
- ssh

Plots: imagemagick (convert tool)
The end

Thanks for your attention

http://sourceforge.net/projects/damned