Parallel Computation

"Faster" is the only reason

But...

greater programming complexity

new kinds of bugs

...and not much help for fixing them.

Parallel Computation

"Faster" is the only reason

But...

greater programming complexity

new kinds of bugs

...and not much help for fixing them.

Can the day or week of user effort be recovered?

Parallel Computation

"Faster" is the only reason

But...

greater programming complexity

new kinds of bugs

...and not much help for fixing them.

Can the day or week of user effort be recovered?

16384 core EPFL IBM BlueGene/P

1 hour at 850MHz

6 months at 3GHz

Parallel Computation

A simulation run takes about a second

want to do 1000’s of them,

varying a dozen or so parameters.

A simulation run takes hours.

want to spread the problem over several machines.

Parallel Computation

A simulation run takes about a second

want to do 1000’s of them,

varying a dozen or so parameters.

- Screensaver   Calin−Jageman and Katz, 2006
- Bulletin−board (Linda)

A simulation run takes hours.

want to spread the problem over several machines.

Parallel Computation

A simulation run takes hours.

want to spread the problem over several machines.

Network

Subnets on different machines

Cells communicate by:

logical spike events with significant axonal, synaptic delay.

postsynaptic conductance depends continuously on presynaptic voltage.

gap junctions

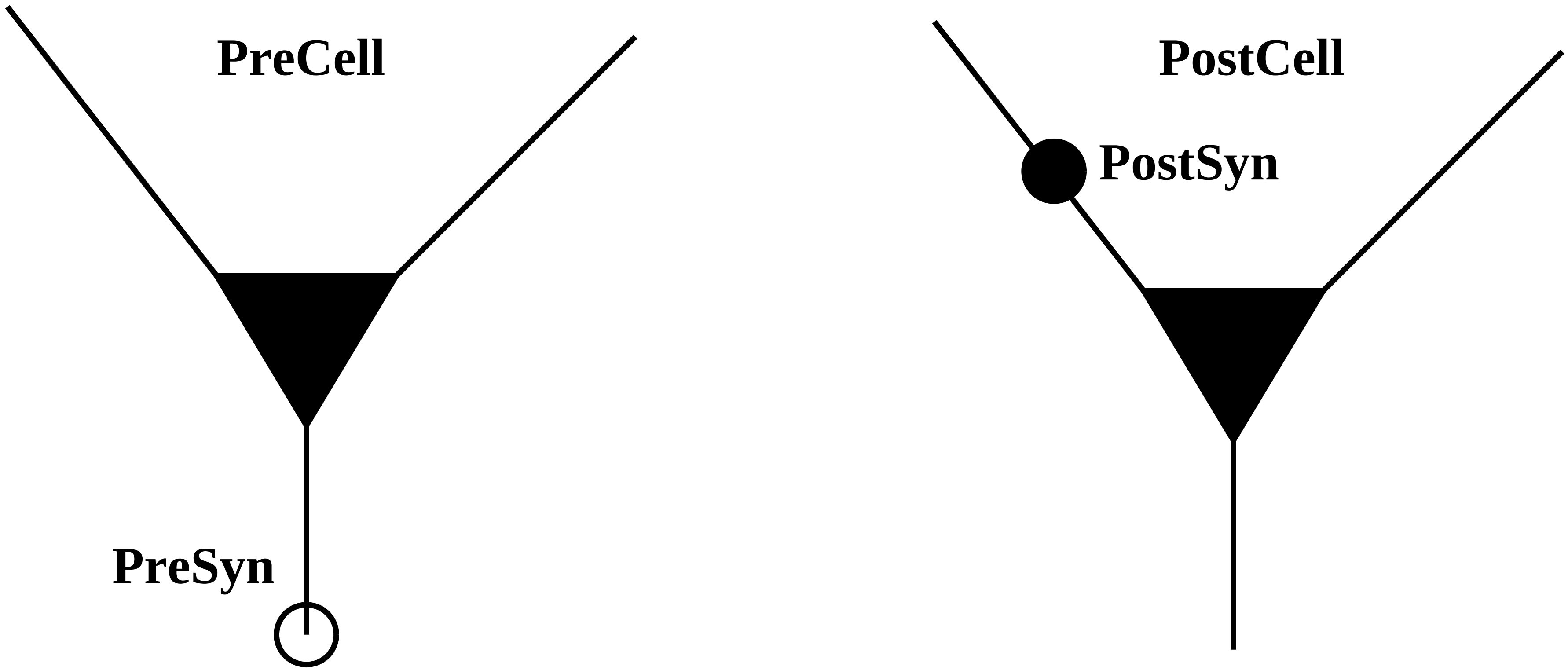
Parallel Computation

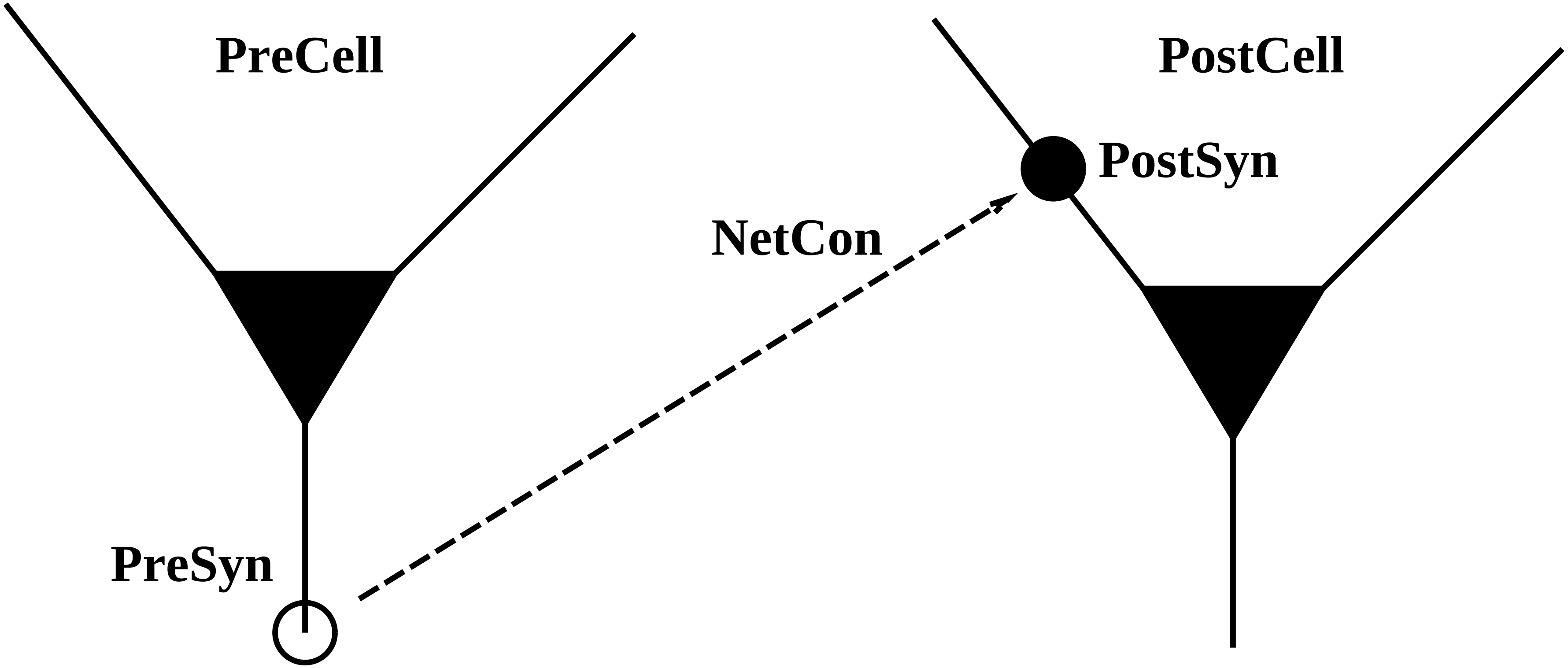
A simulation run takes hours.

- want to spread the problem over several machines.

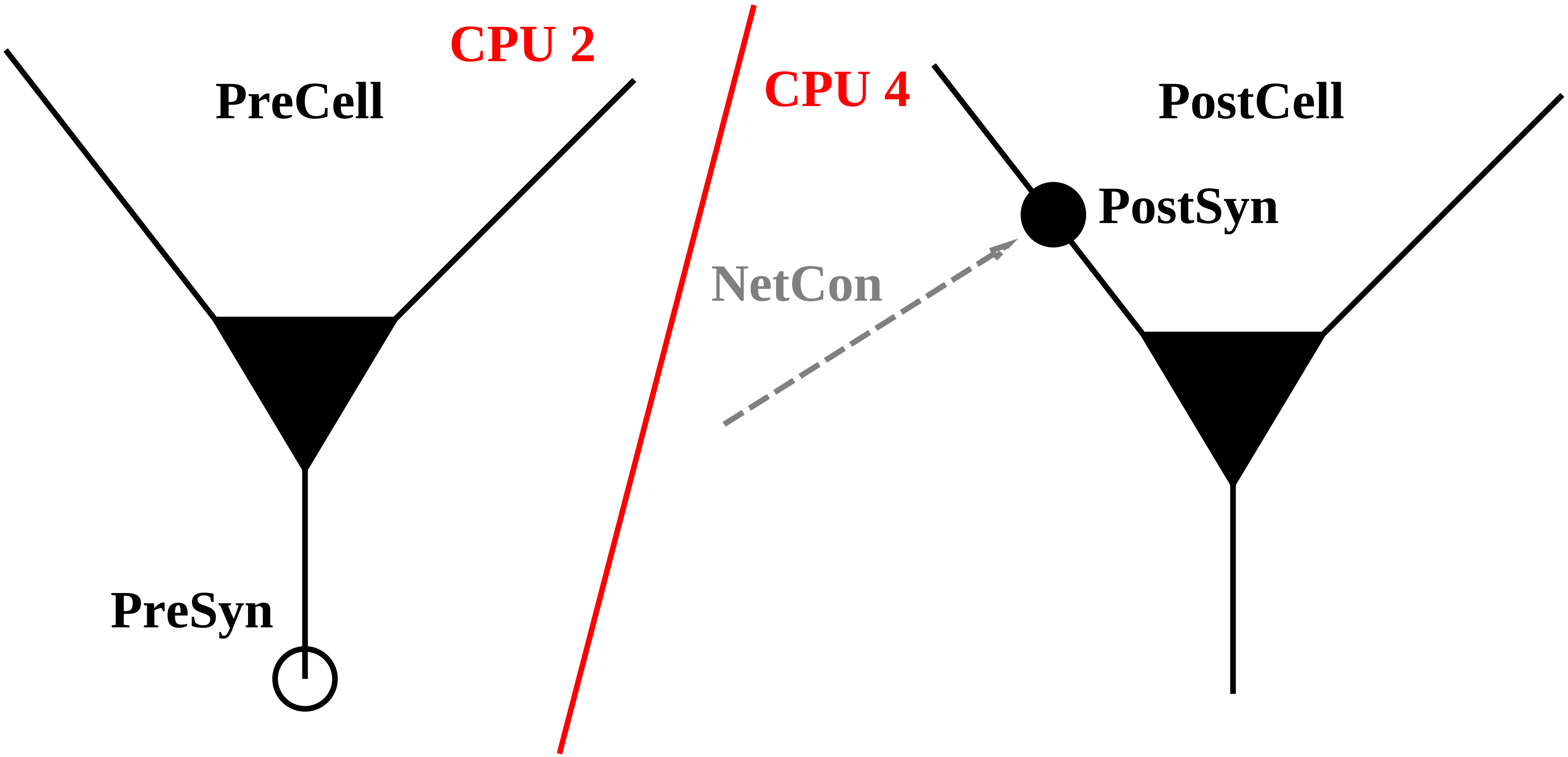
Single cells

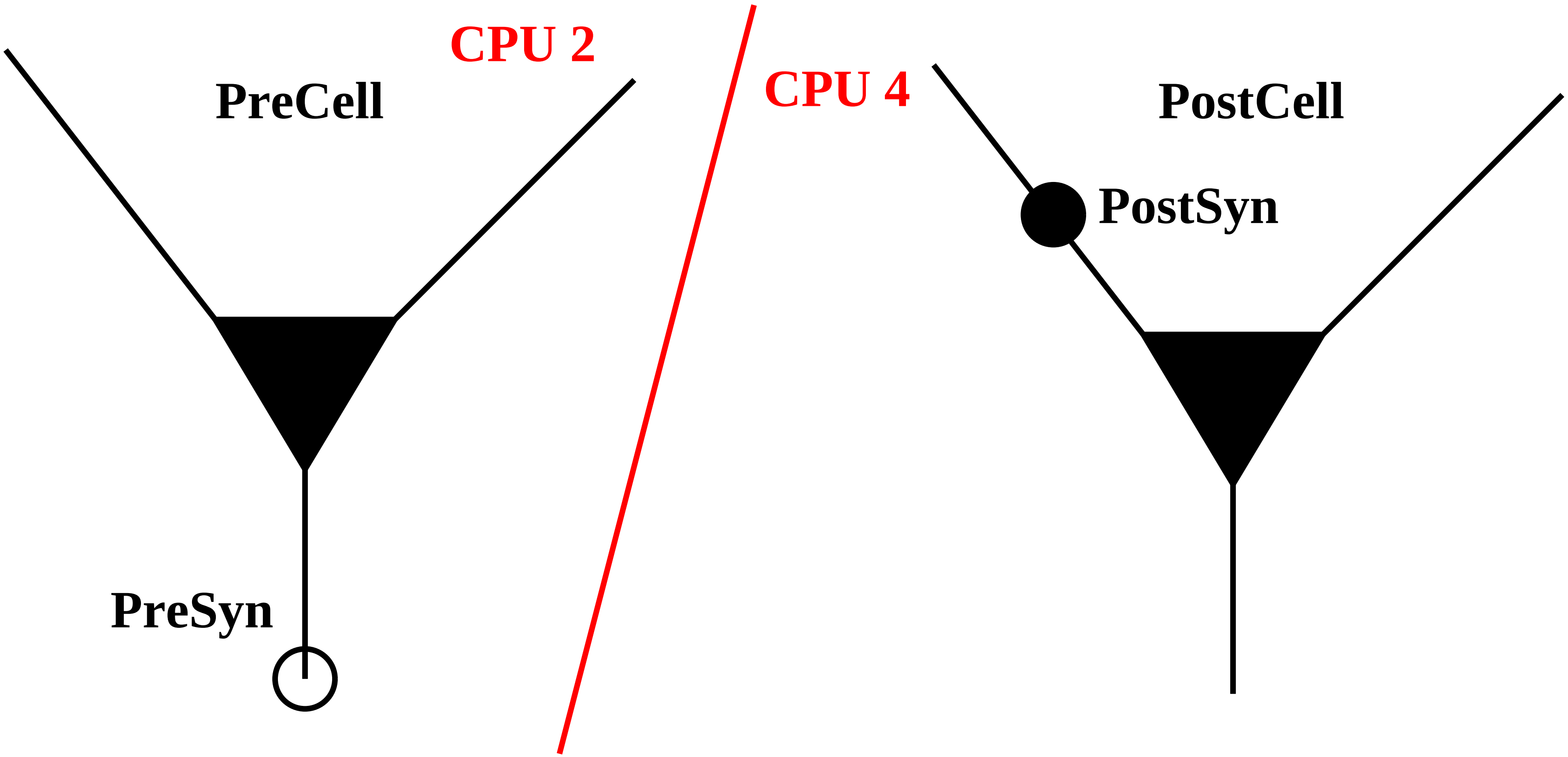
- portions of the tree cable equation on different machines.


PreCell
PostCell
PostSyn
PreSyn

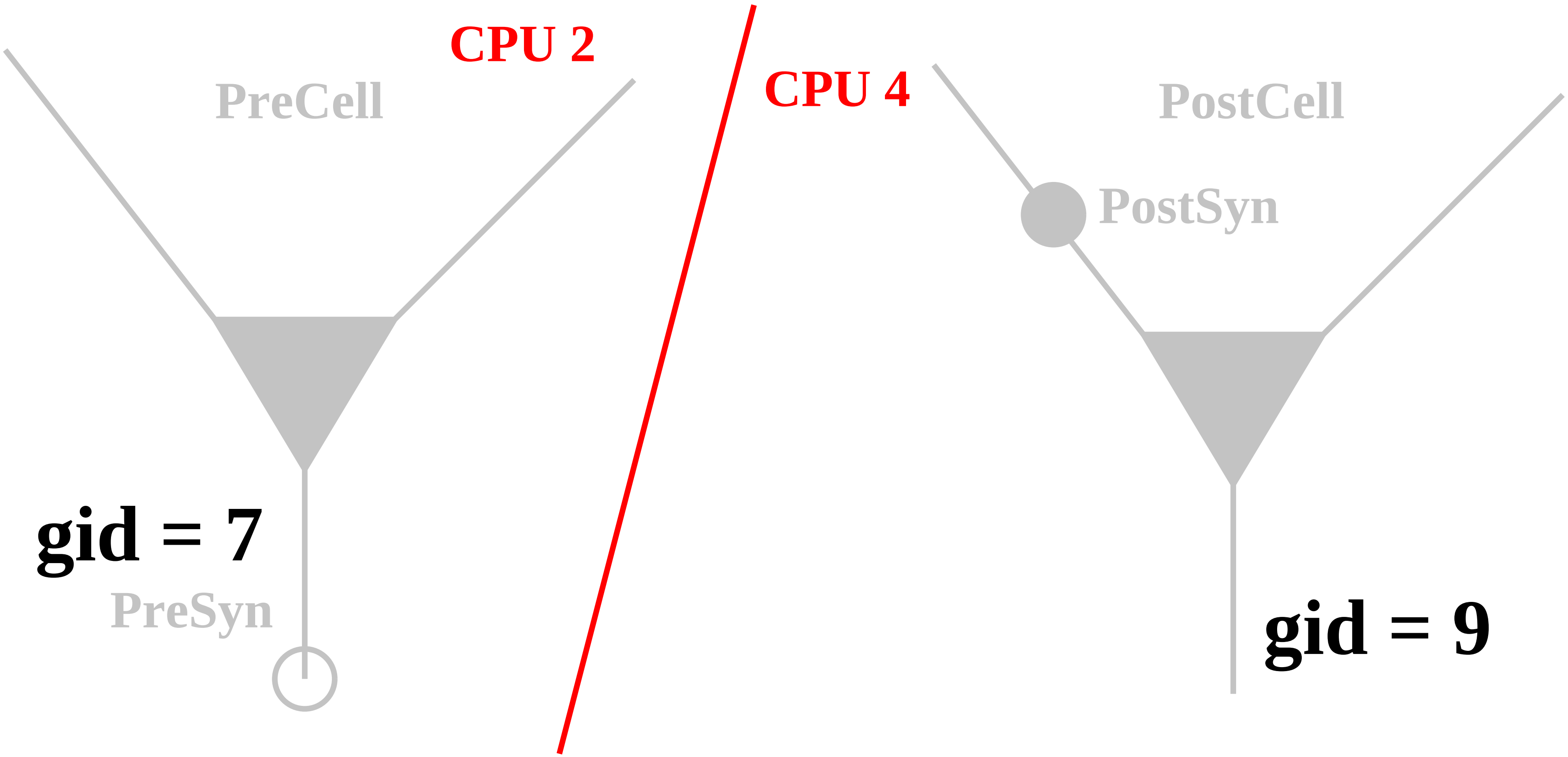



```
nc = new NetCon(PreSyn, PostSyn)
```


CPU 2
PreCell
CPU 4
PostCell
PostSyn
NetCon
PreSyn




```
pc = new ParallelContext()
```



Every spike source (cell) must have a global id number.

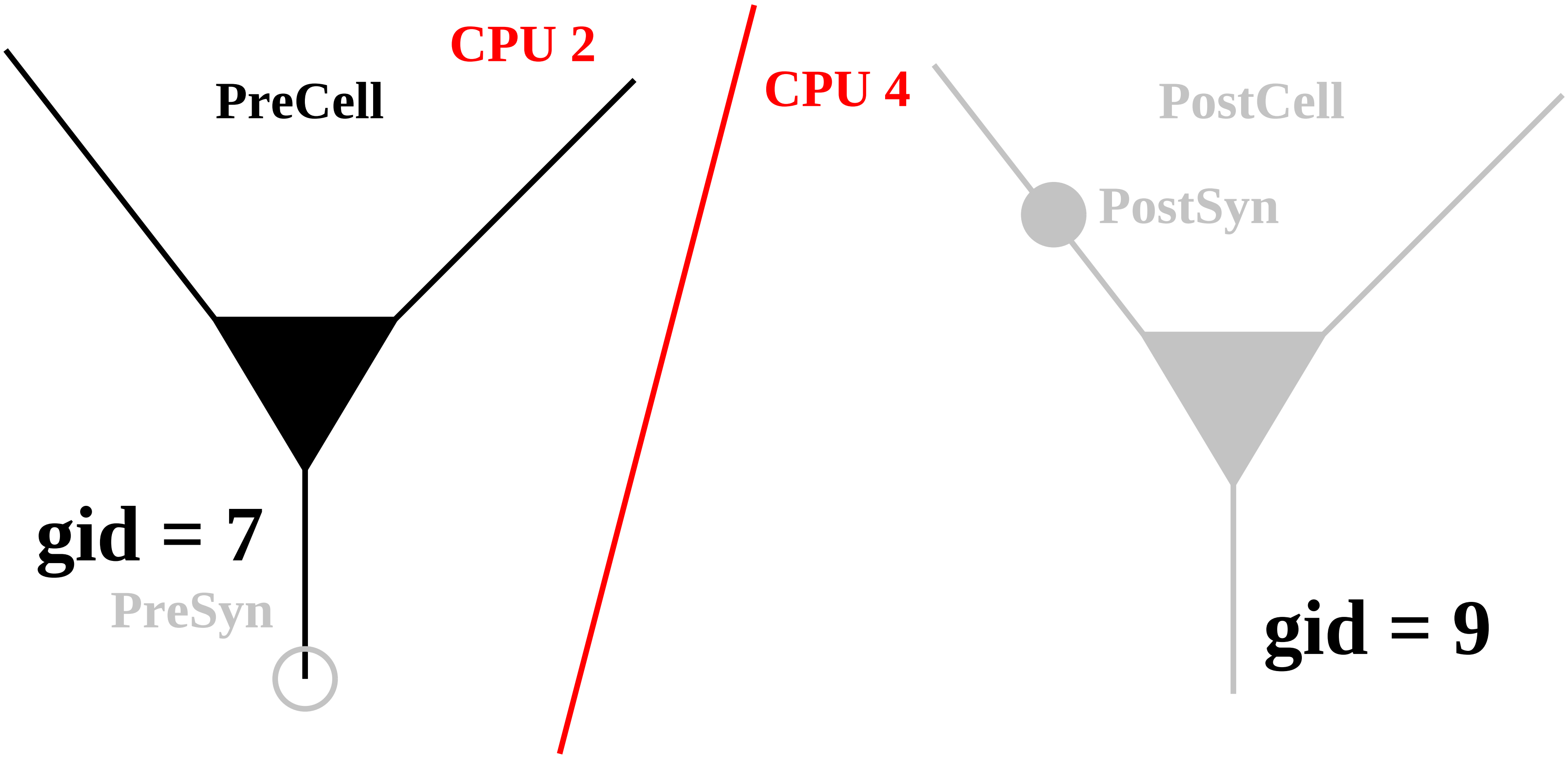
CPU 0		...	CPU 3		CPU 4	
pc.id	0		pc.id	3	pc.id	4
pc.nhost	5		pc.nhost	5	pc.nhost	5
ncell	14		ncell	14	ncell	14

gid (CPU 0)	gid (CPU 3)	gid (CPU 4)
0	3	4
5	8	9
10	13	

An efficient way to distribute:

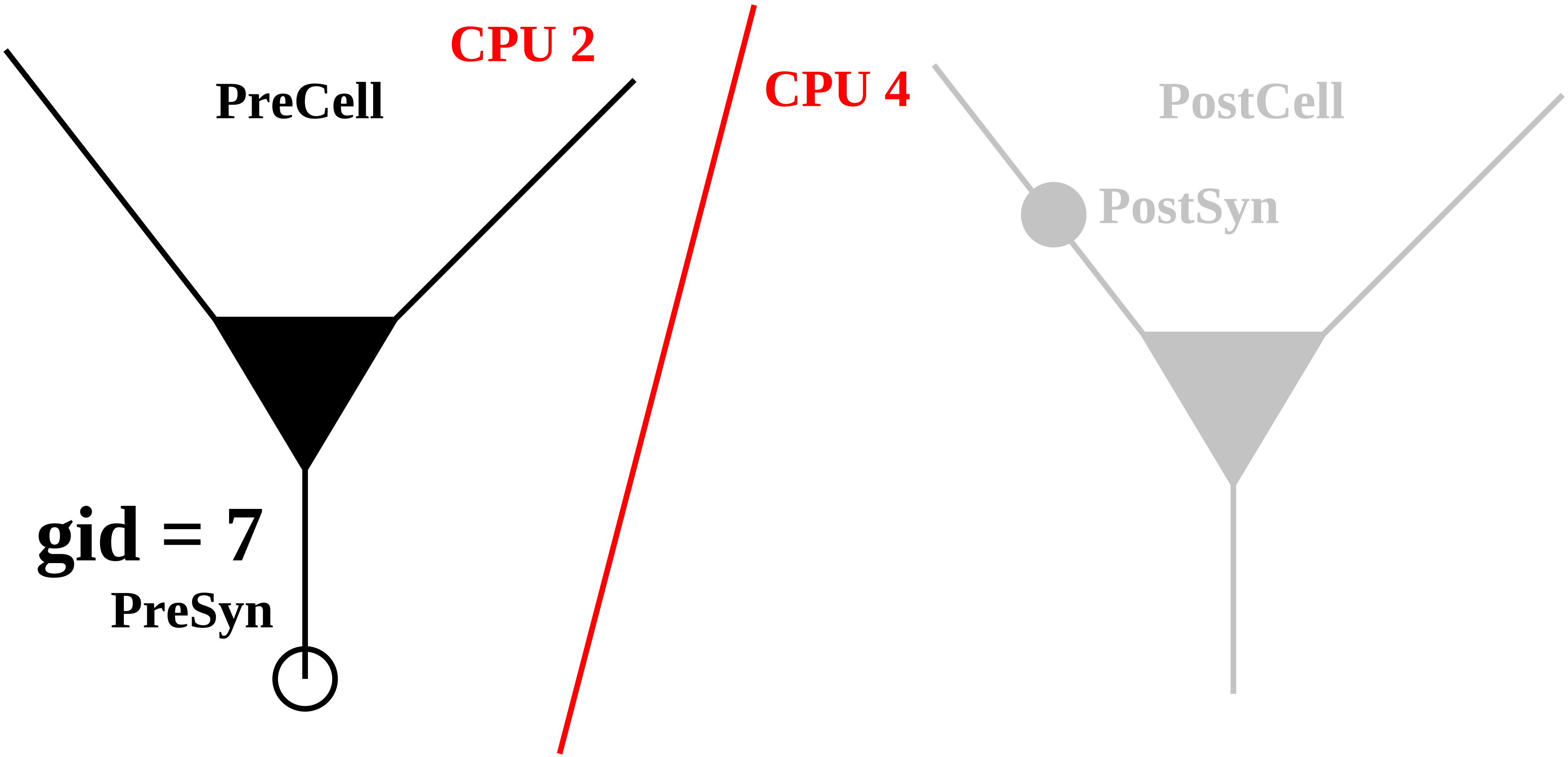
```
for (gid = pc.id; gid < ncell; gid += pc.nhost) {
    pc.set_gid2node(gid, pc.id)
    ...
}
```

body executed only ncell/nhost times, not ncell.



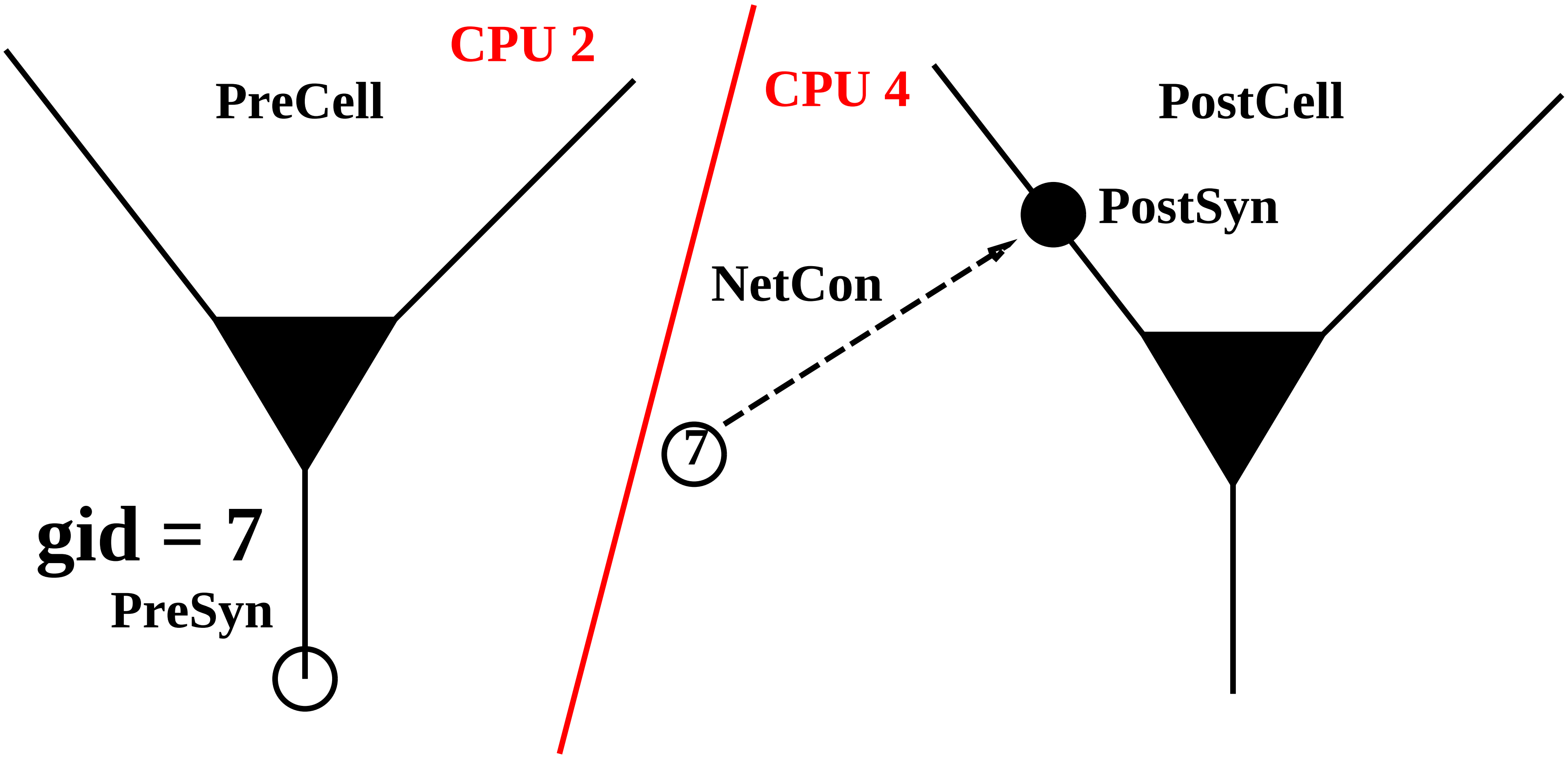
Create cell only where the gid exists.

```
if (pc.gid_exists(7)) {
    PreCell = new Cell()
}
```



Associate gid with spike source.

```
nc = new NetCon(PreSyn, nil)
pc.cell(7, nc)
```



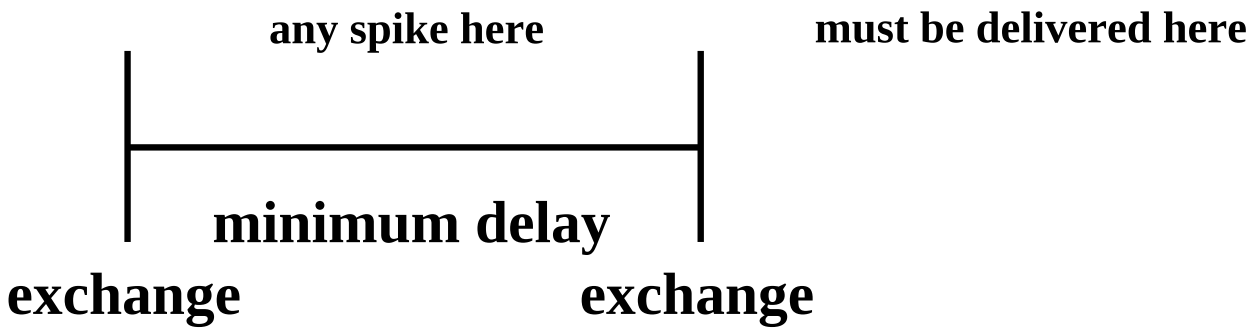
Create NetCon on CPU where target exists.

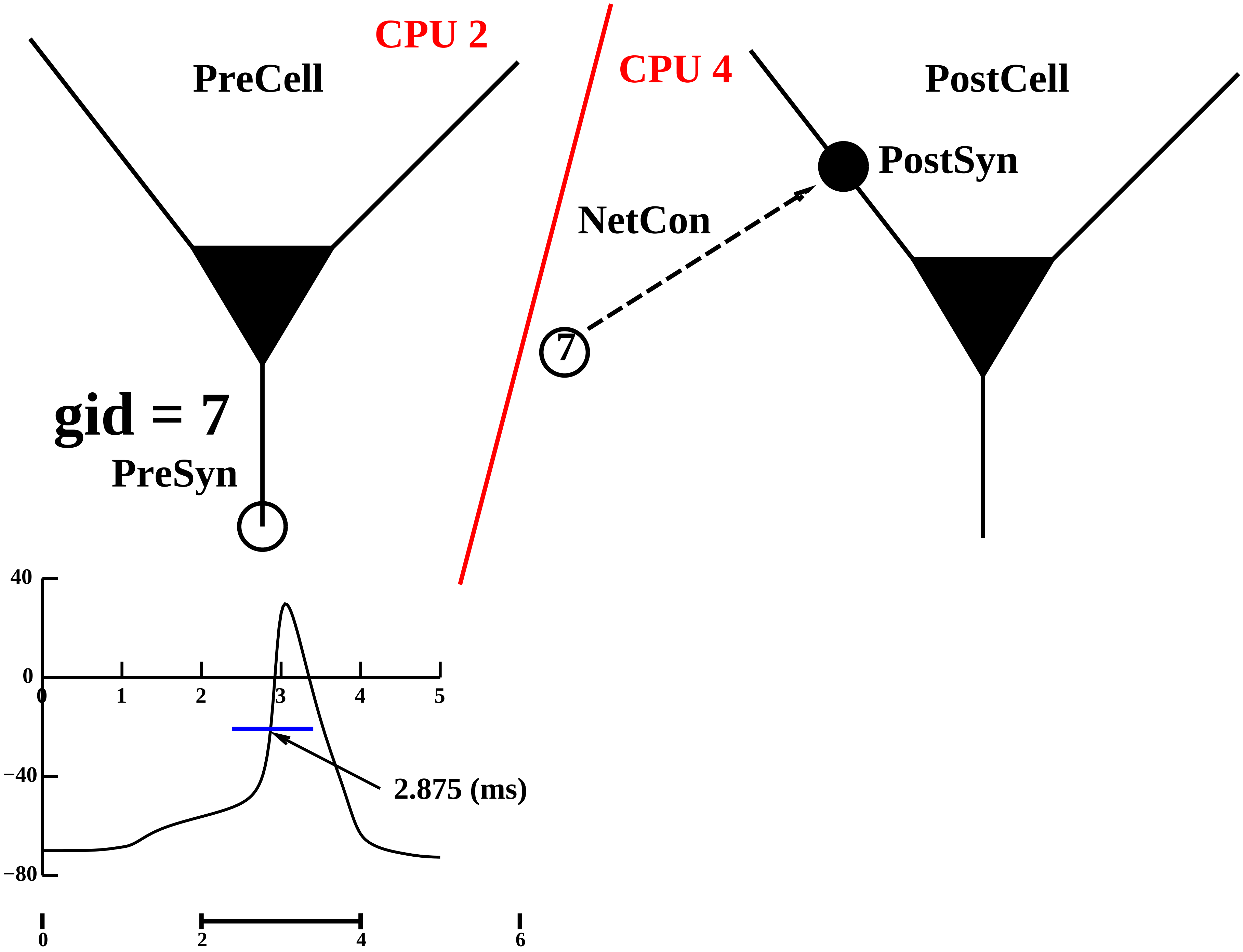
```
nc = pc.gid_connect(7, PostSyn)
```

Run using the idiom

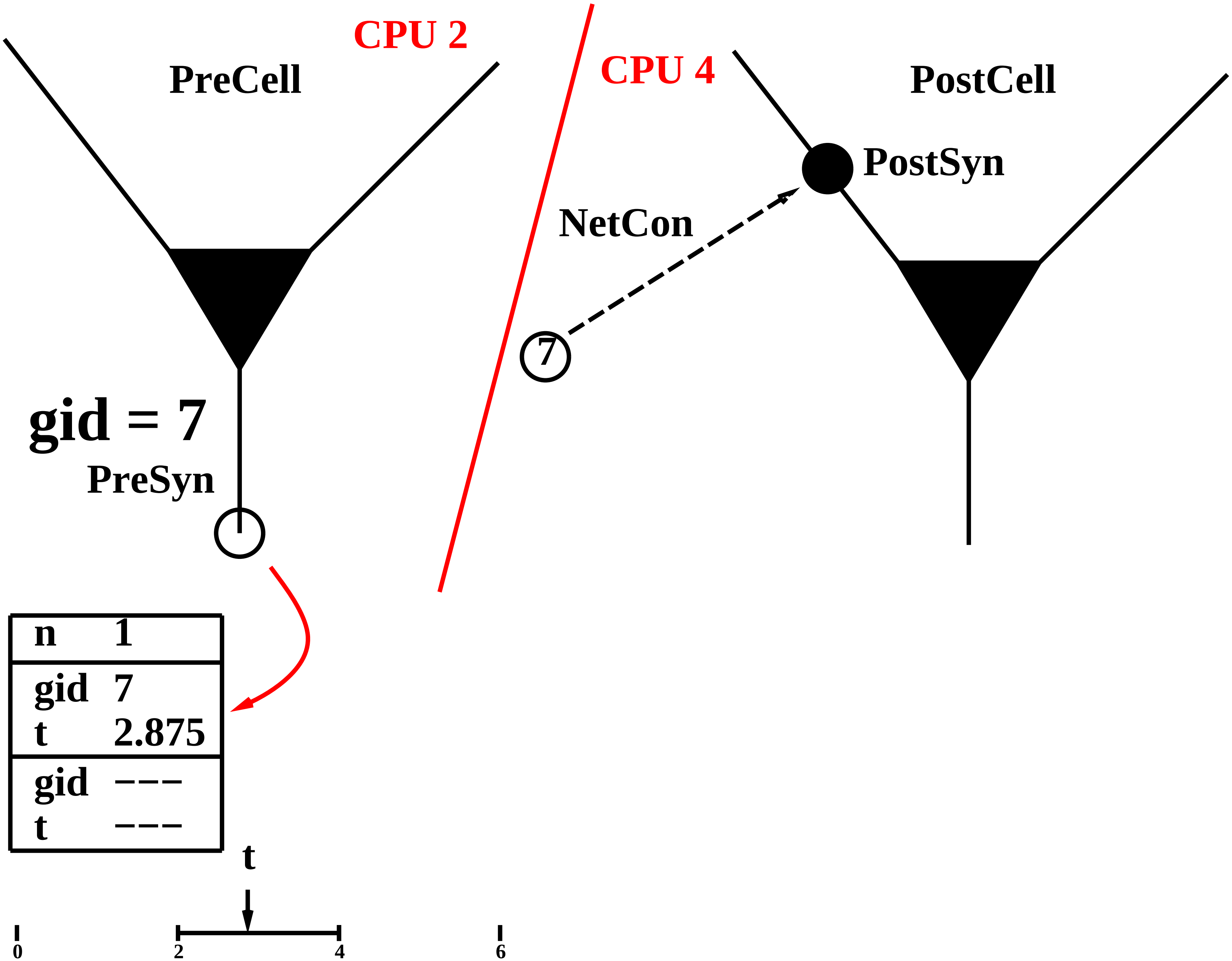
```
pc.set_maxstep(10)
stdinit()
pc.psolve(tstop)
```

pc.set_maxstep() uses
MPI_Allreduce
to determine minimum delay.

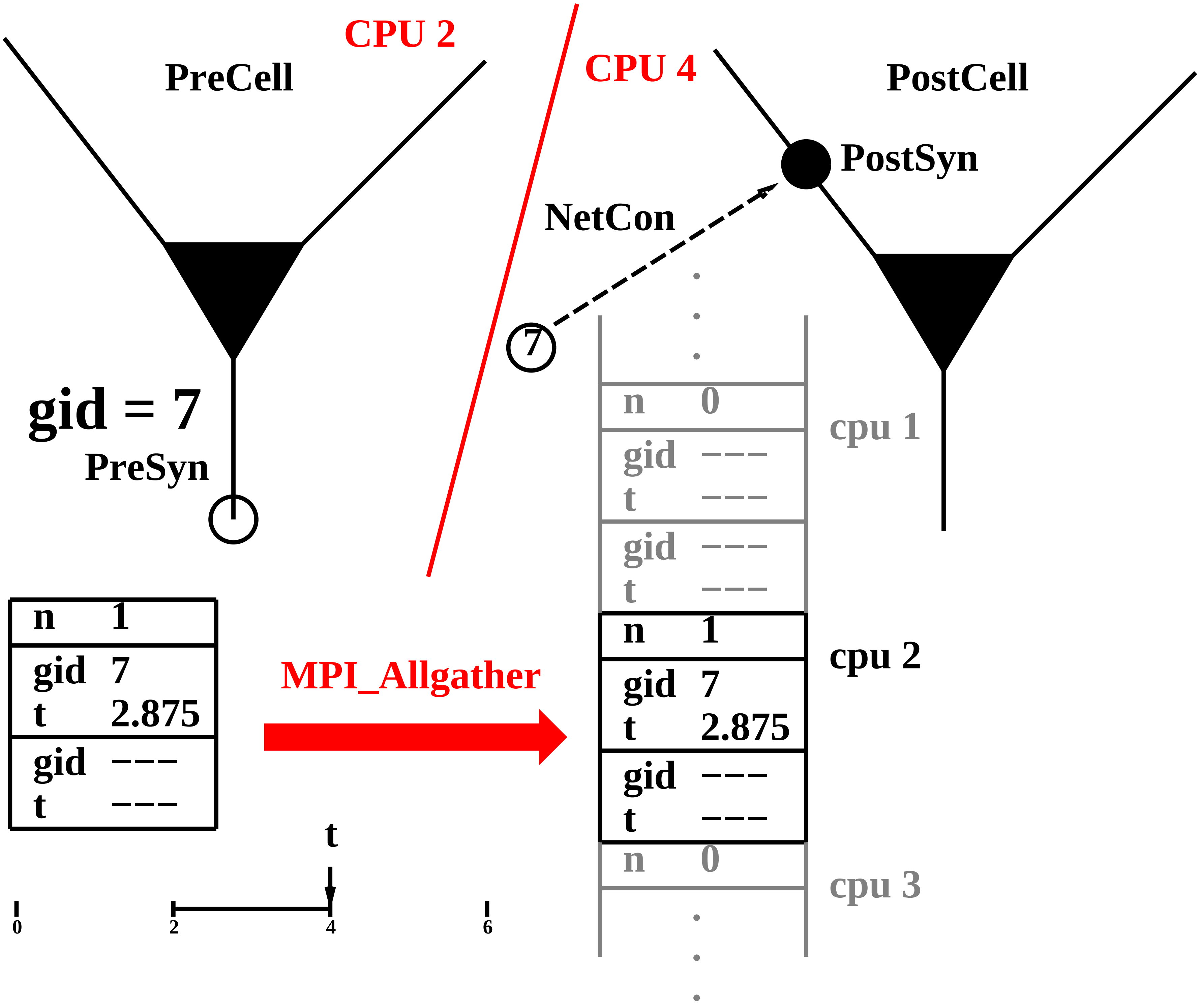




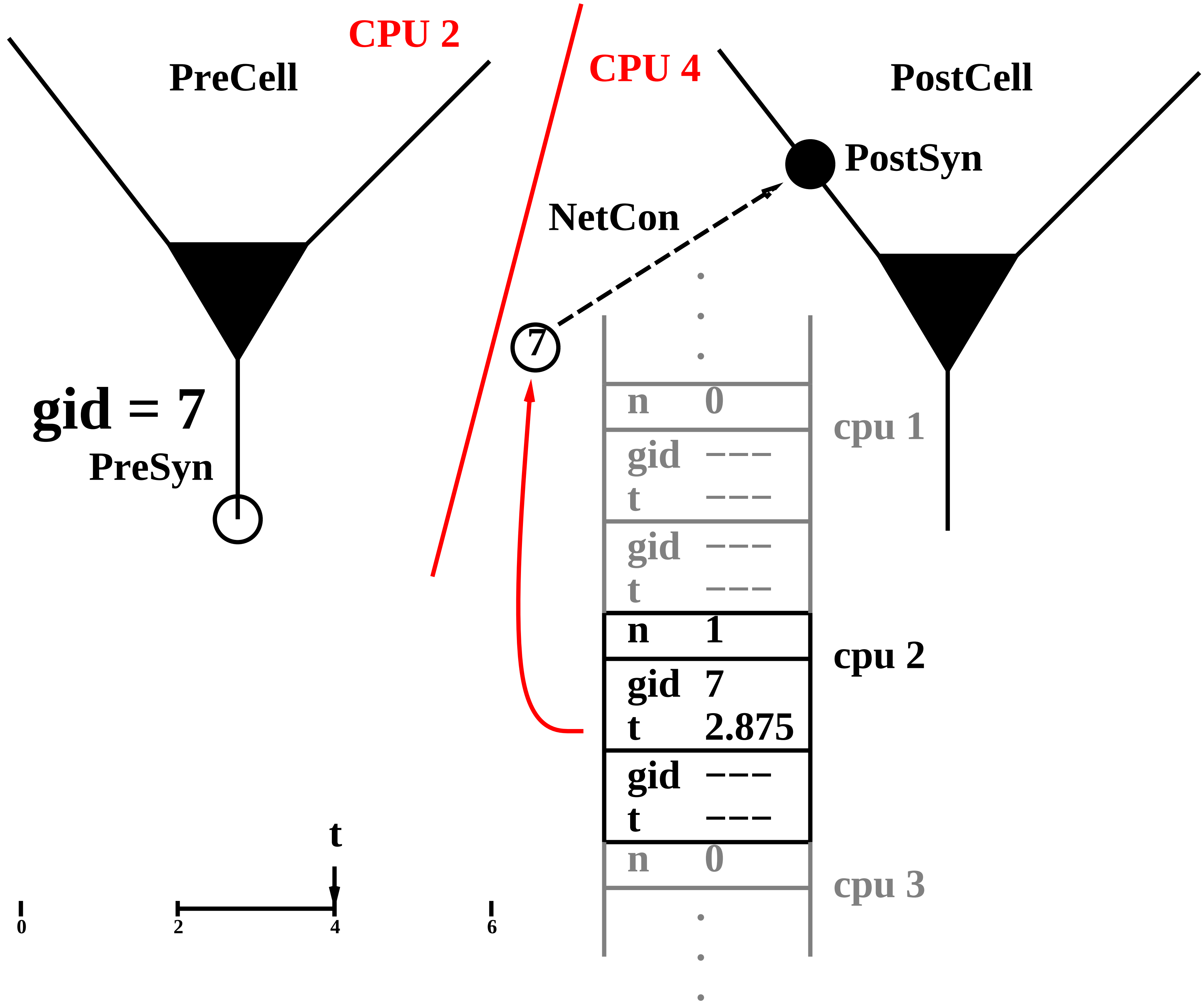
CPU 2
PreCell
CPU 4
PostCell
PostSyn
NetCon
7
gid = 7
PreSyn
40
0
−40
−80
0
1
2
3
4
5
2.875 (ms)
0
2
4
6



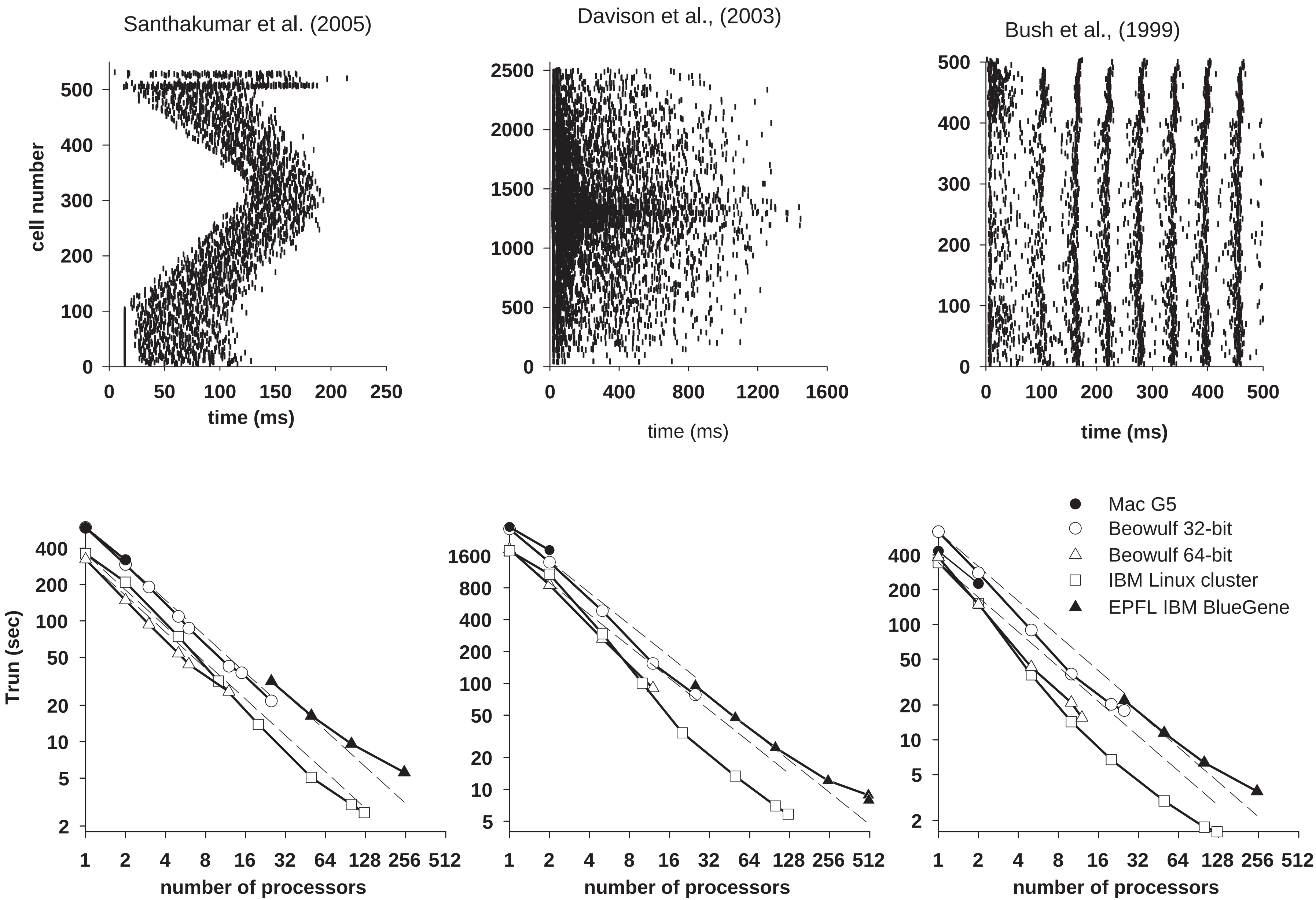
CPU 2
CPU 4
PreCell
PostCell
PostSyn
NetCon
7
gid = 7
PreSyn
n 1
gid 7
t 2.875
gid ---
t ---
t
0
2
4
6



CPU 2
PreCell
CPU 4
PostCell
PostSyn
NetCon
7
gid = 7
PreSyn
n 0
cpu 1
gid ---
t ---
gid ---
t ---
n 1
gid 7
t 2.875
gid ---
t ---
MPI_Allgather
n 1
cpu 2
gid 7
t 2.875
gid ---
t ---
n 0
cpu 3
t
0
2
4
6


CPU 2
CPU 4
PreCell
PostCell
PostSyn
NetCon
gid = 7
PreSyn
7
n 0
cpu 1
gid ---
t ---
gid ---
t ---
n 1
cpu 2
gid 7
t 2.875
gid ---
t ---
n 0
cpu 3
t
0
2
4
6

Migliore et al (2006) J. Comput. Neurosci. 21(2):119

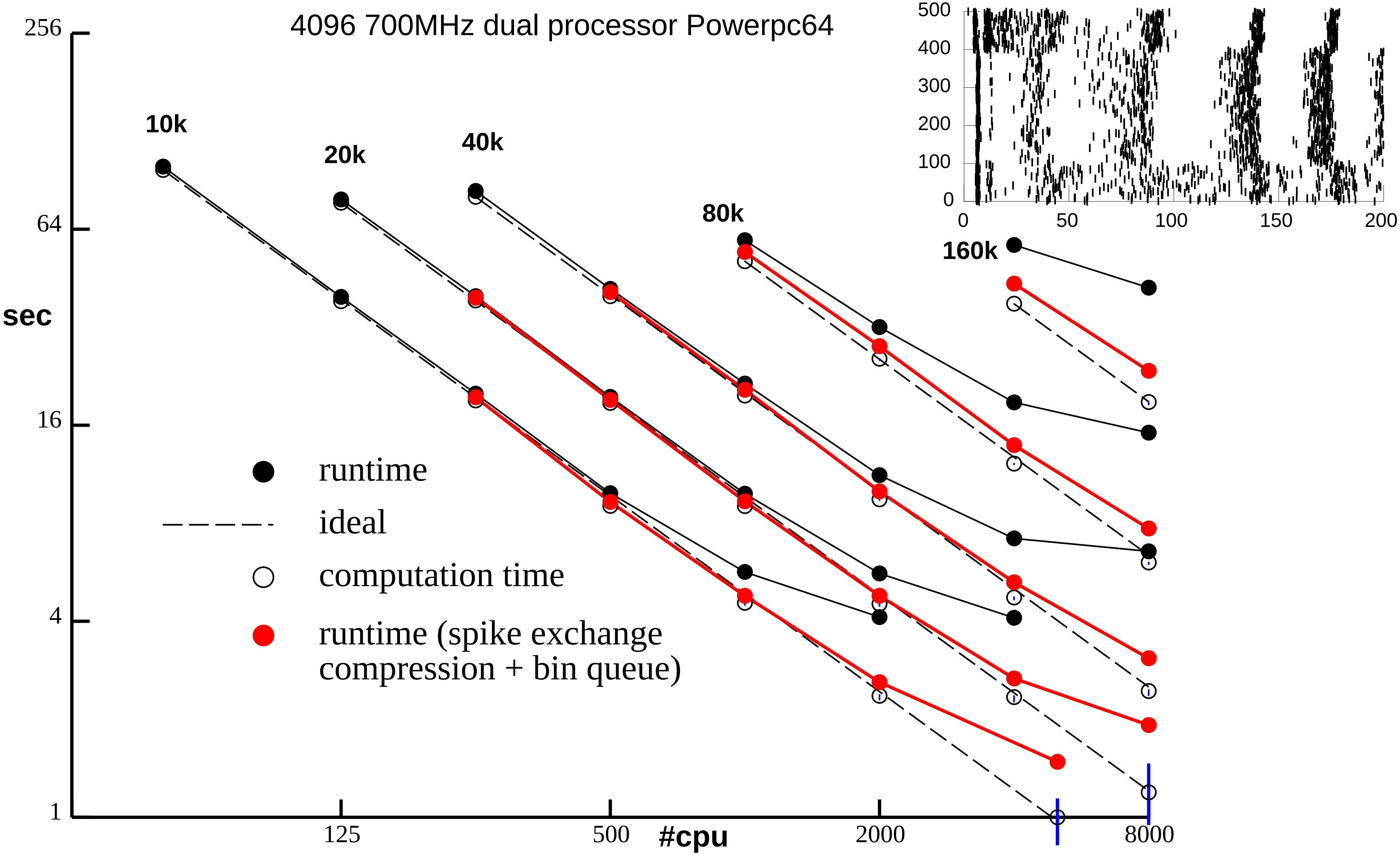


Bush, Prince, & Miller (1999) J. Neurophys. 82:1748

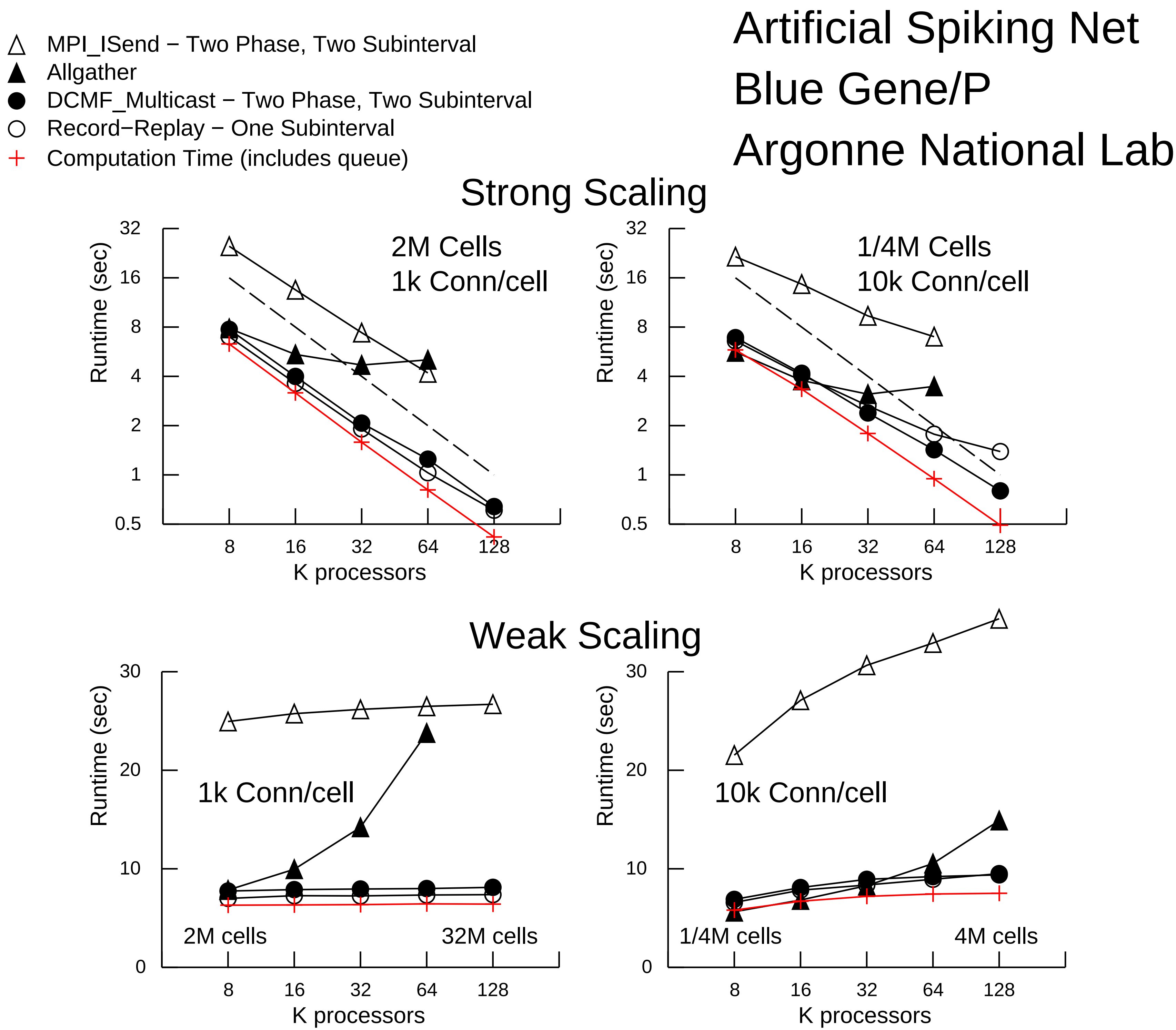
Increased pyramidal excitability and posttraumatic epileptogenesis without disinhibition: a model.

EPFL IBM BlueGene/L

4096 700MHz dual processor Powerpc64

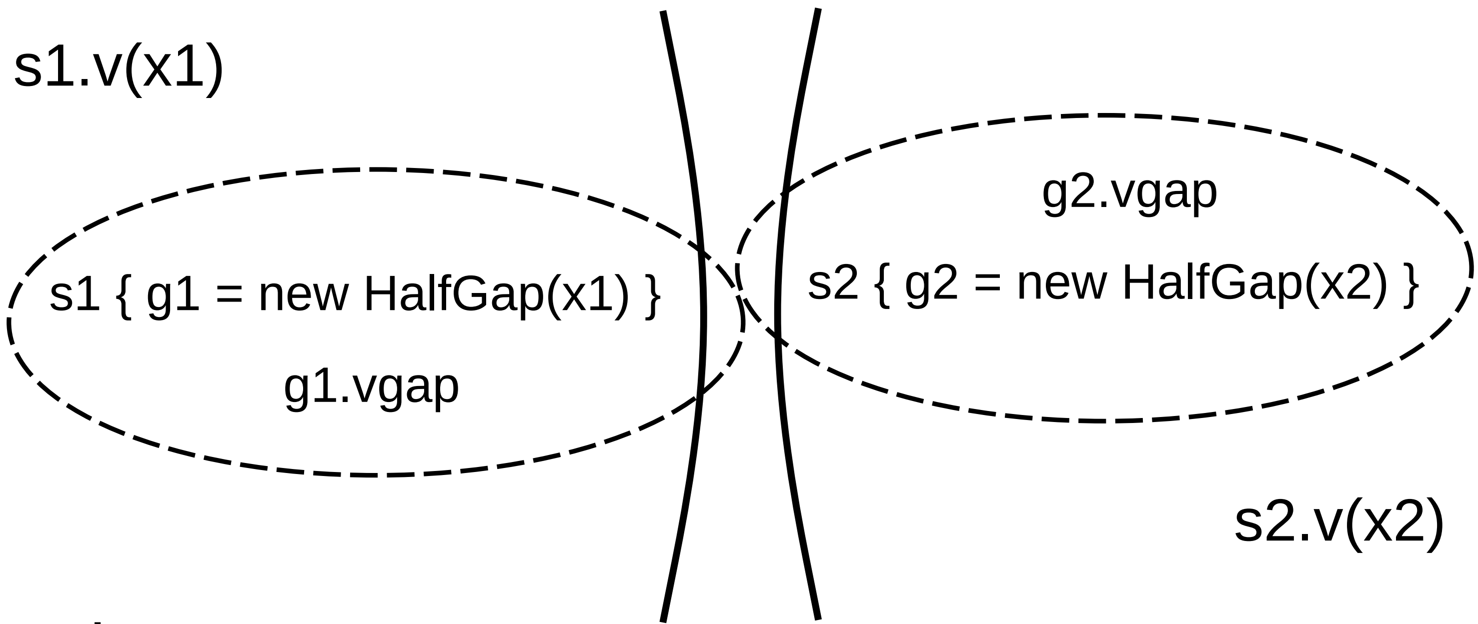


#cells	#states	#netcons	#outputSpikes	#spikedeliver
10k	444,664	17,167,785	31,118	52,040,794
20k	888,664	68,655,566	33,960	110,634,002
40k	1,777,664	274,591,128	69,529	452,987,907
80k	3,553,664	1,098,302,267	294974	2,147,483,647 *
160k	7,107,664	4,393,084,577	844,175	22,847,784,937



MPI_ISend − Two Phase, Two Subinterval
Allgather
DCMF_Multicast − Two Phase, Two Subinterval
Record−Replay − One Subinterval
Computation Time (includes queue)
Artificial Spiking Net
Blue Gene/P
Argonne National Lab
Strong Scaling
2M Cells
1k Conn/cell
1/4M Cells
10k Conn/cell
Runtime (sec)
32
16
8
4
2
1
0.5
8
16
32
64
128
K processors
Weak Scaling
1k Conn/cell
10k Conn/cell
30
20
10
0
2M cells
32M cells
1/4M cells
4M cells

Continuous Voltage Exchange



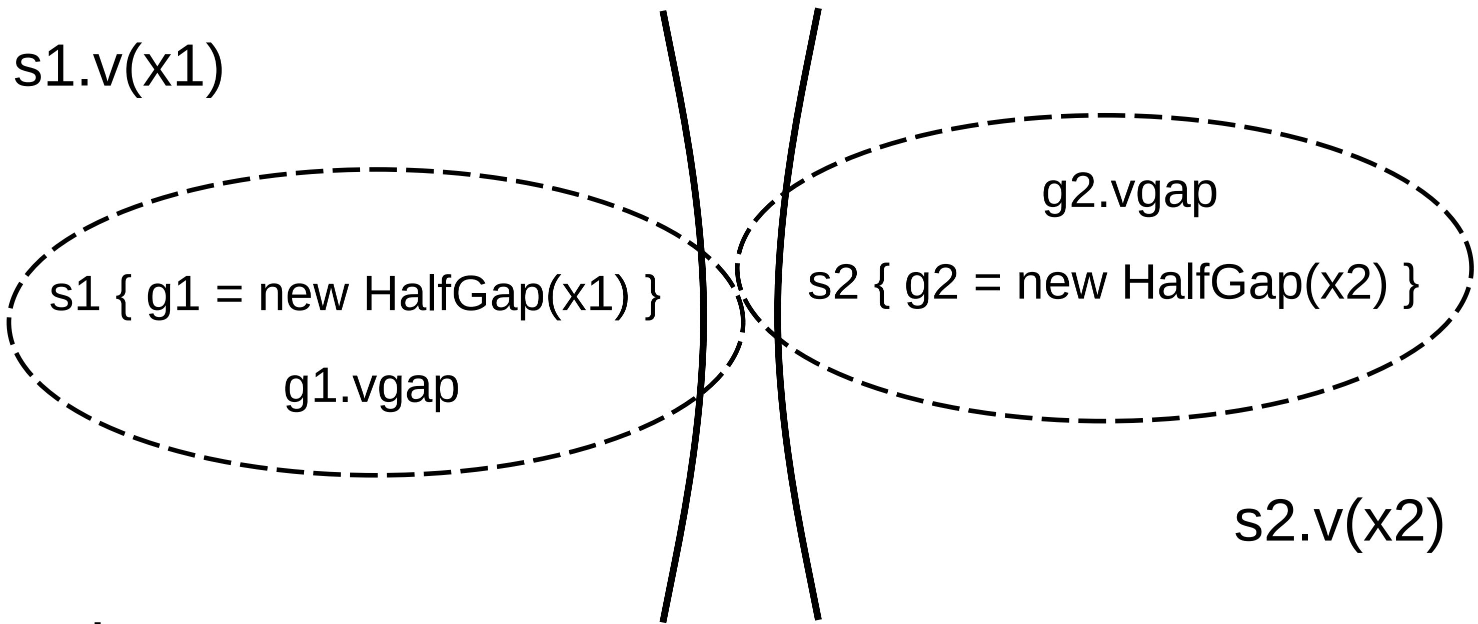
gap.mod

```
NEURON {
  POINT_PROCESS HalfGap
  ELECTRODE_CURRENT i
  RANGE r, i, vgap
}
PARAMETER { r = 1e9 (megohm) }
```

```
ASSIGNED {
  v (millivolt)
  vgap (millivolt)
  i (nanoamp)
}
CURRENT { i = (vgap - v)/r }
```

Continuous Voltage Exchange

pc.source_var(&source_var, sgid)



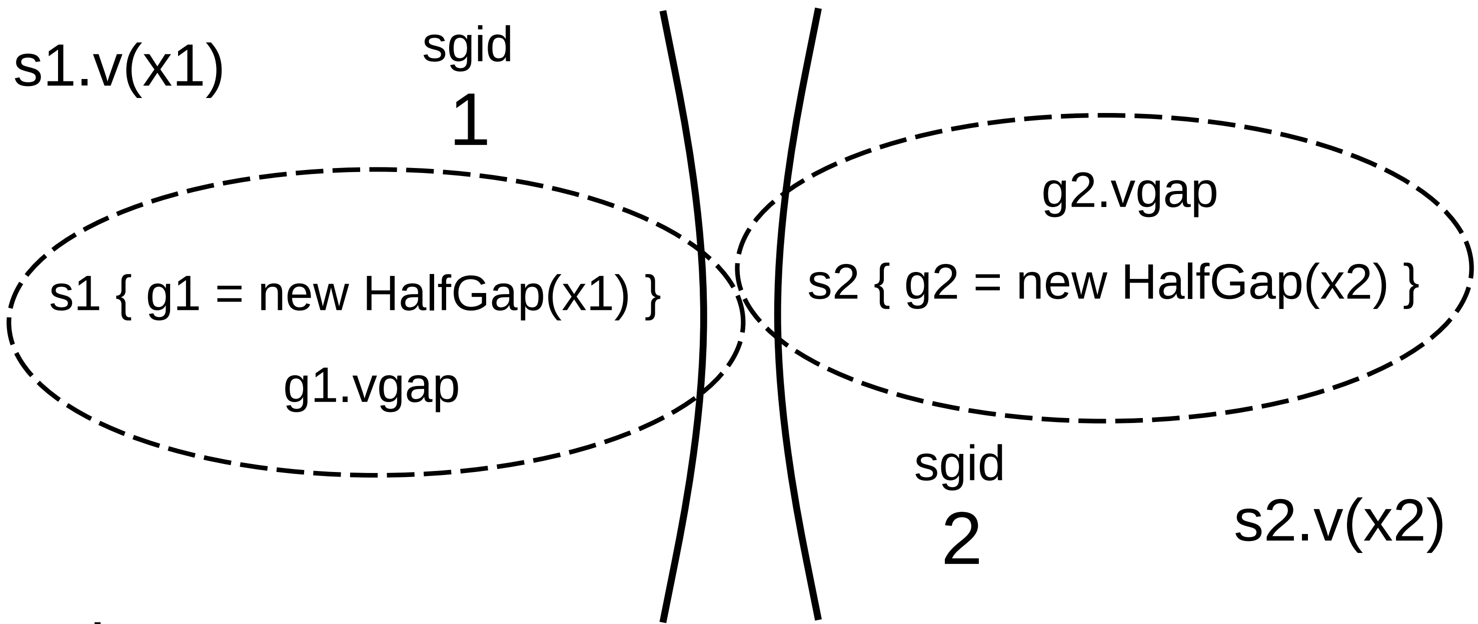
gap.mod

```
NEURON {
  POINT_PROCESS HalfGap
  ELECTRODE_CURRENT i
  RANGE r, i, vgap
}
PARAMETER { r = 1e9 (megohm) }
```

```
ASSIGNED {
  v (millivolt)
  vgap (millivolt)
  i (nanoamp)
}
CURRENT { i = (vgap - v)/r }
```

Continuous Voltage Exchange

pc.source_var(&source_var, sgid)



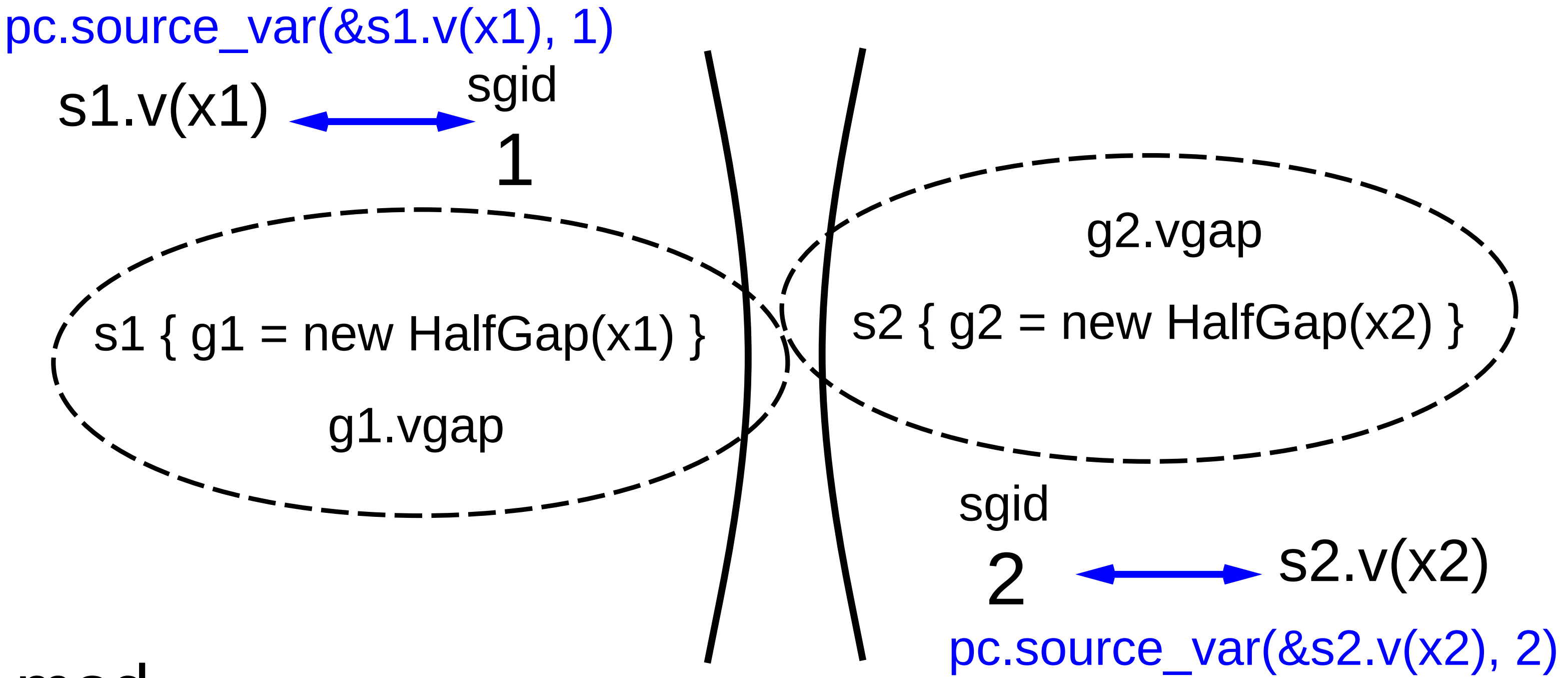
gap.mod

```
NEURON {
  POINT_PROCESS HalfGap
  ELECTRODE_CURRENT i
  RANGE r, i, vgap
}
PARAMETER { r = 1e9 (megohm) }
```

```
ASSIGNED {
  v (millivolt)
  vgap (millivolt)
  i (nanoamp)
}
CURRENT { i = (vgap - v)/r }
```

Continuous Voltage Exchange

pc.source_var(&source_var, sgid)



gap.mod

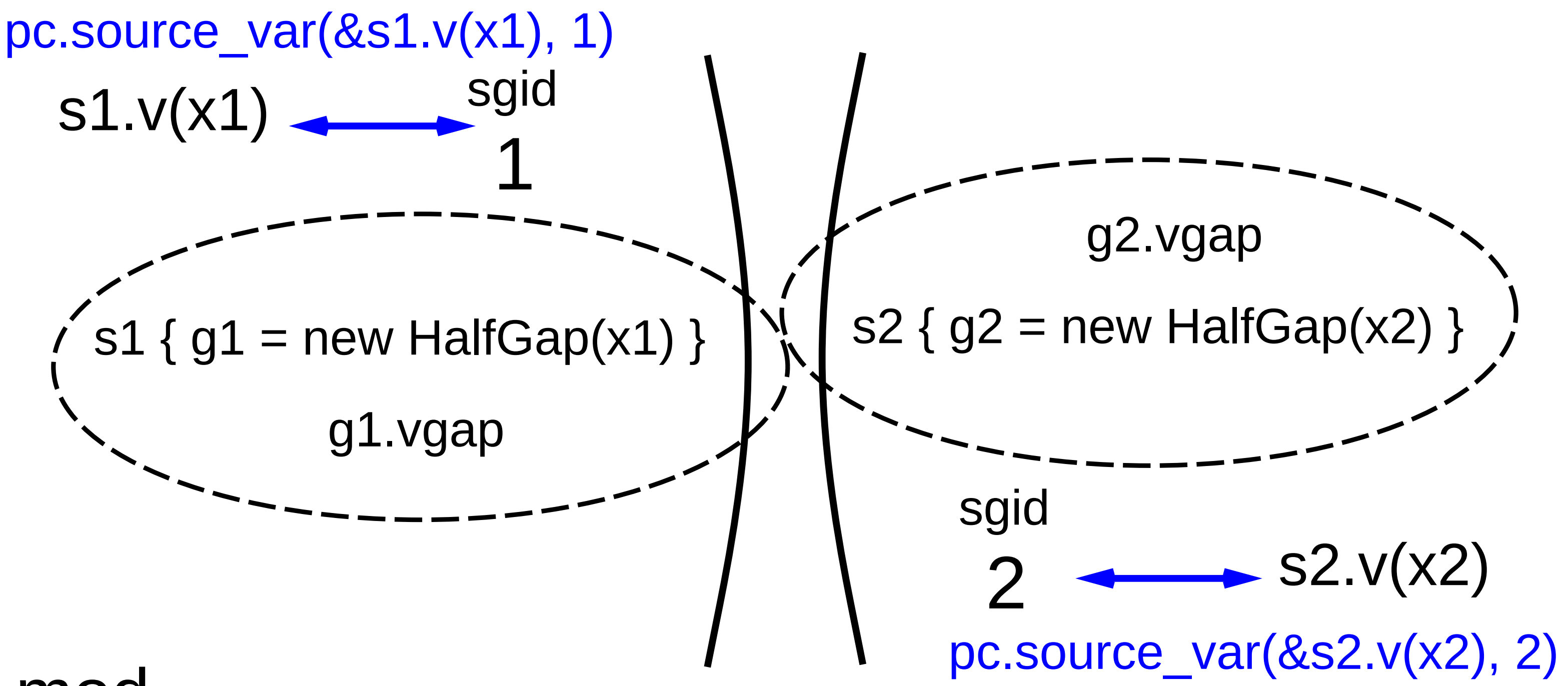
```
NEURON {
  POINT_PROCESS HalfGap
  ELECTRODE_CURRENT i
  RANGE r, i, vgap
}
PARAMETER { r = 1e9 (megohm) }
```

```
ASSIGNED {
  v (millivolt)
  vgap (millivolt)
  i (nanoamp)
}
CURRENT { i = (vgap - v)/r }
```

Continuous Voltage Exchange

pc.source_var(&source_var, sgid)

pc.target_var(&target_var, sgid)



gap.mod

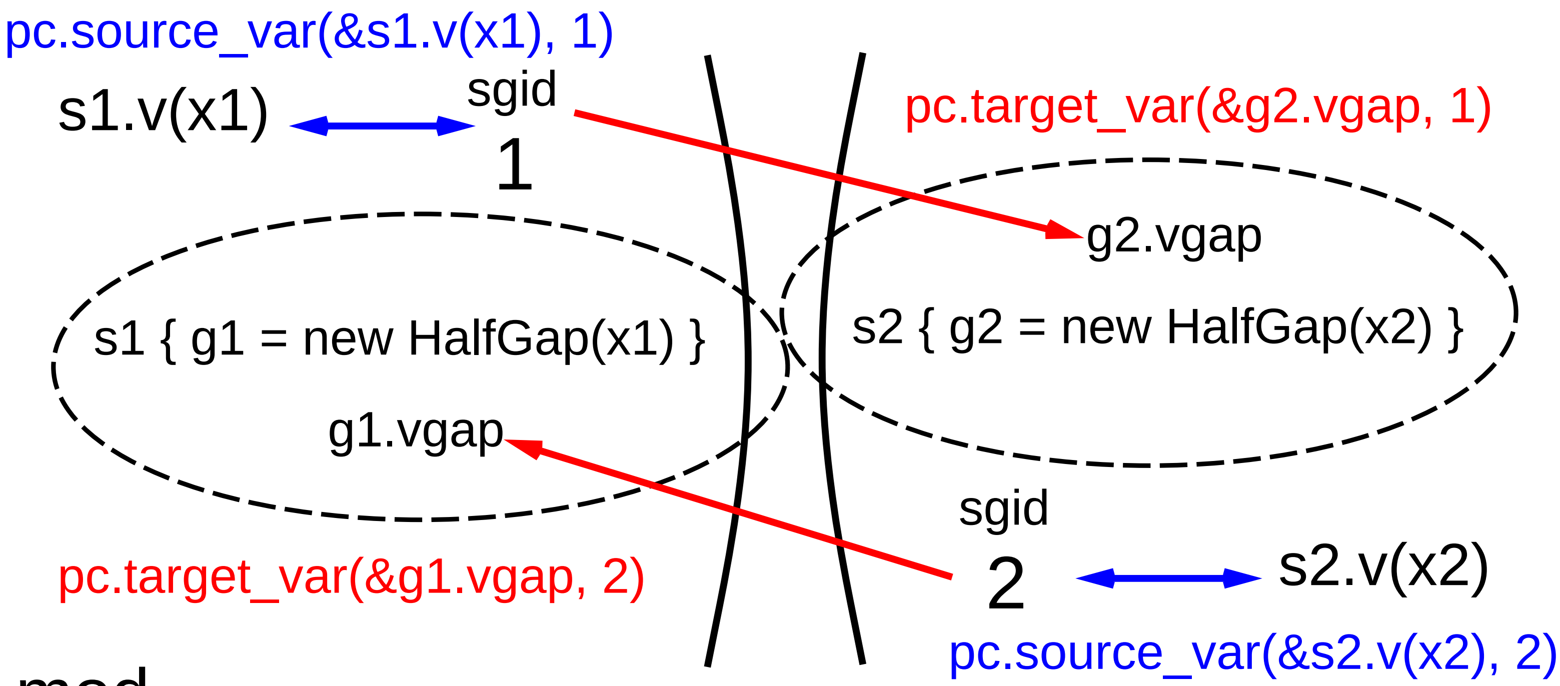
```
NEURON {
  POINT_PROCESS HalfGap
  ELECTRODE_CURRENT i
  RANGE r, i, vgap
}
PARAMETER { r = 1e9 (megohm) }
```

```
ASSIGNED {
  v (millivolt)
  vgap (millivolt)
  i (nanoamp)
}
CURRENT { i = (vgap - v)/r }
```

Continuous Voltage Exchange

pc.source_var(&source_var, sgid)

pc.target_var(&target_var, sgid)



gap.mod

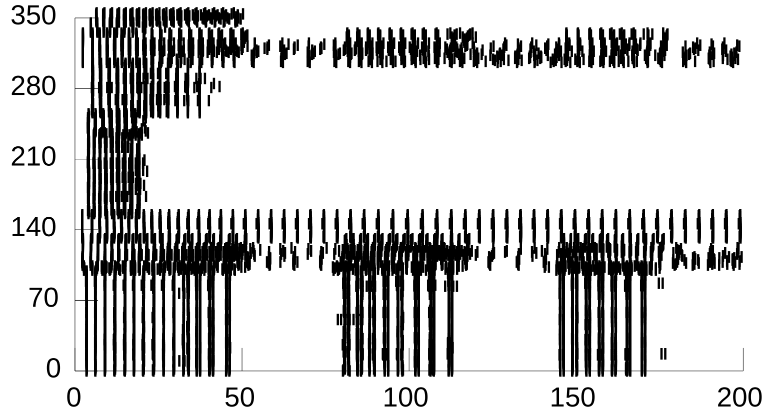
```
NEURON {
  POINT_PROCESS HalfGap
  ELECTRODE_CURRENT i
  RANGE r, i, vgap
}
PARAMETER { r = 1e9 (megohm) }

ASSIGNED {
  v (millivolt)
  vgap (millivolt)
  i (nanoamp)
}
CURRENT { i = (vgap - v)/r }
```

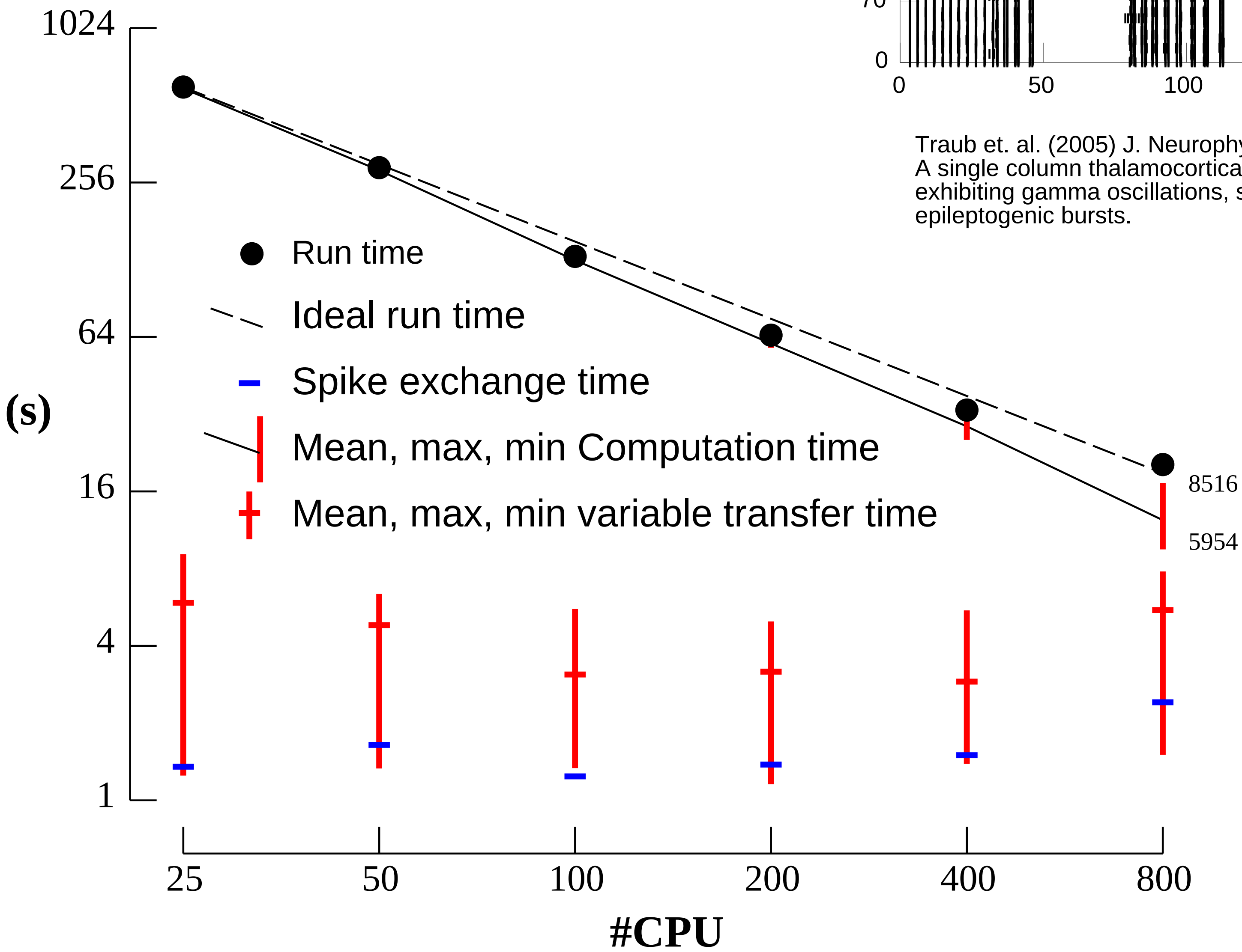
Pittsburgh Supercomputing Center

Bigben Cray XT3

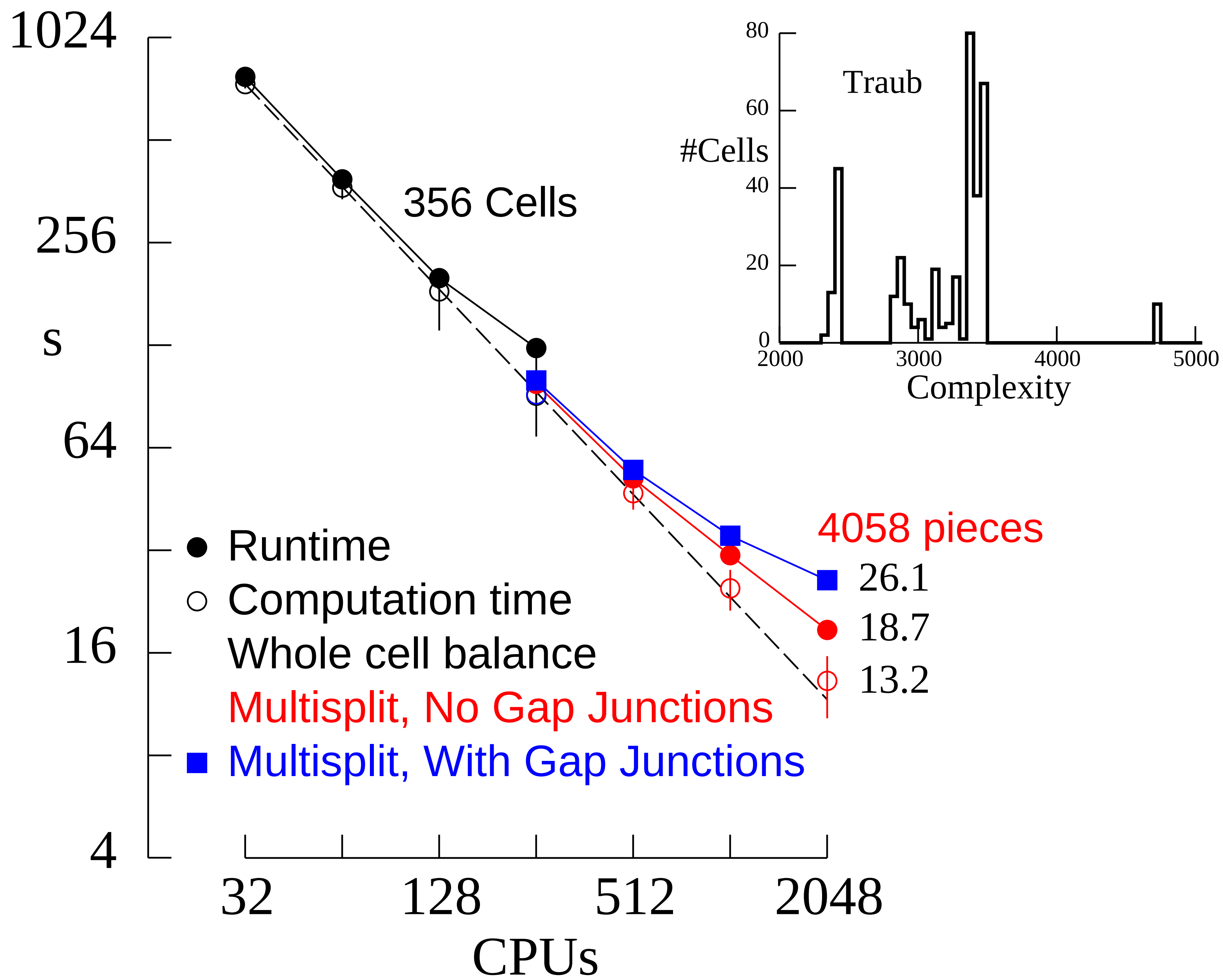
2068 2.4 GHz Opteron Processors



Traub et. al. (2005) J. Neurophysiol 93: 2194
A single column thalamocortical network model exhibiting gamma oscillations, sleep spindles and epileptogenic bursts.



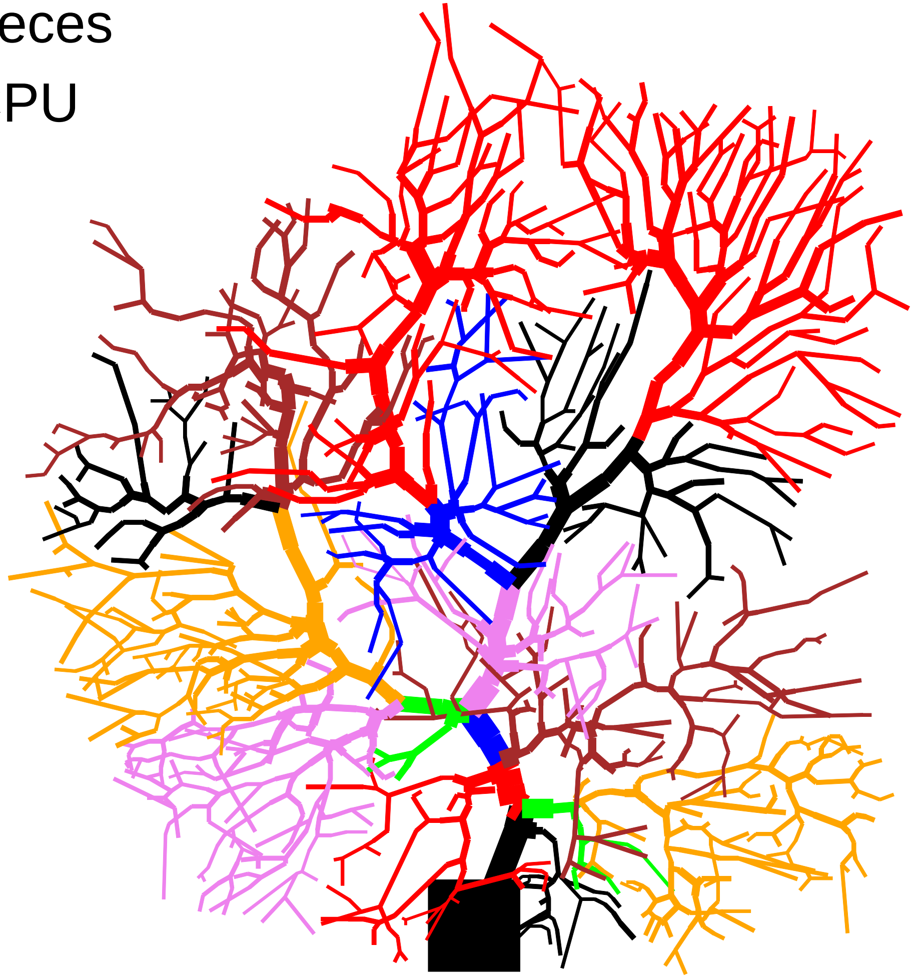
3560 cells 14 types
3500 gap junctions
5,596,810 equations
73,465 spikes
1,122,520 connections
19,844,187 delivered

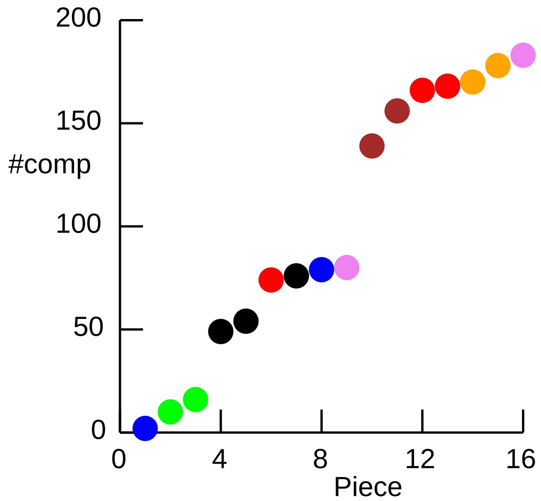


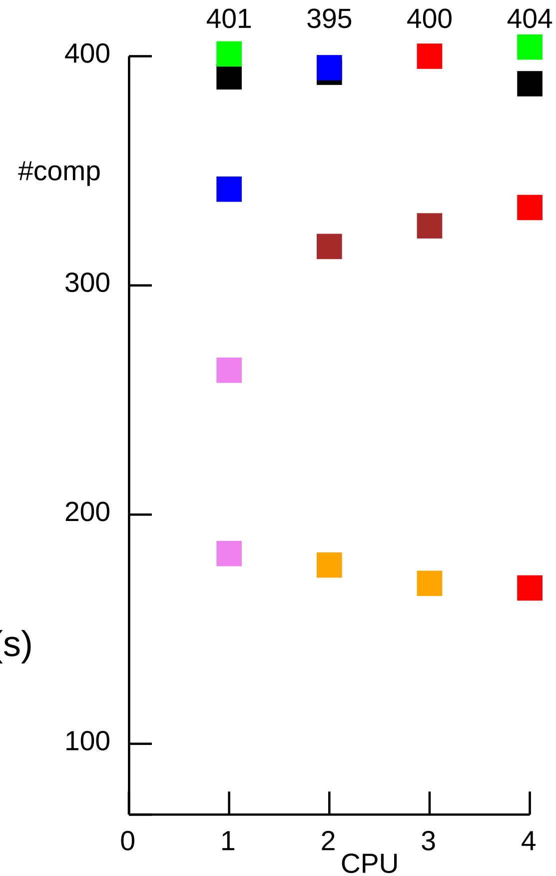
1024
256
64
16
4
s
32
128
512
2048
CPUs
356 Cells
Runtime
Computation time
Whole cell balance
Multisplit, No Gap Junctions
Multisplit, With Gap Junctions
4058 pieces
26.1
18.7
13.2
Traub
#Cells
80
60
40
20
0
2000
3000
4000
5000
Complexity

16 Pieces

4 CPU

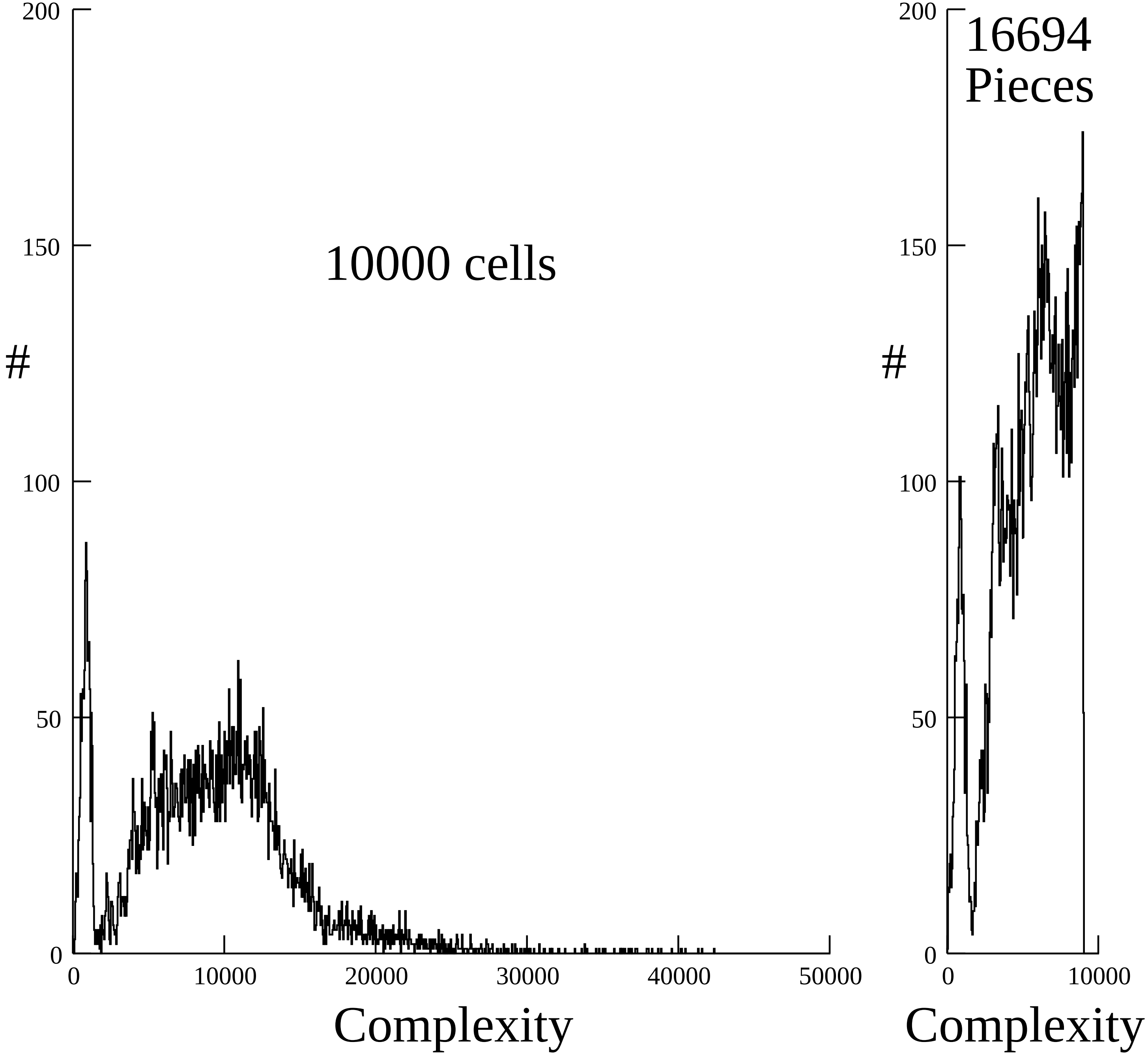


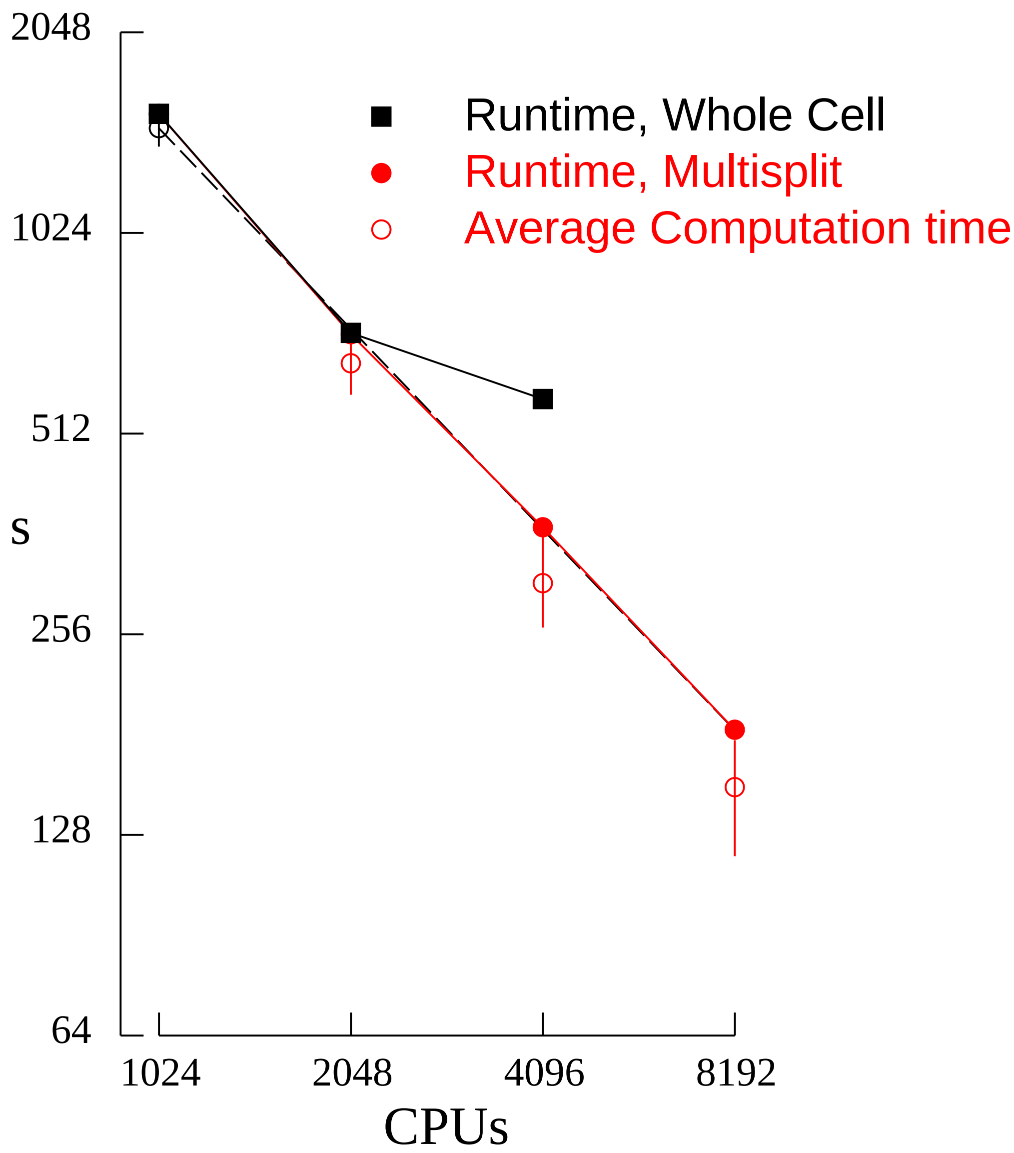




	Time (s)	
CPU	Computation	Exchange
0	13.82	0.56
1	13.35	1.03
2	13.47	0.90
3	13.56	0.82

	Runtime(s)
16 pieces, 1 cpu	55.0
wholecell, 1 cpu	56.2
16 pieces, 4 cpu	14.4


10000 cells
#
Complexity
0
10000
20000
30000
40000
50000
200
150
100
50
16694
Pieces
#
Complexity
0
10000


2048
1024
512
256
128
64
s
Runtime, Whole Cell
Runtime, Multisplit
Average Computation time
1024
2048
4096
8192
CPUs