NoPain – Meeting

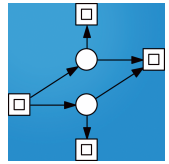
Monika Heiner Christian Rohr

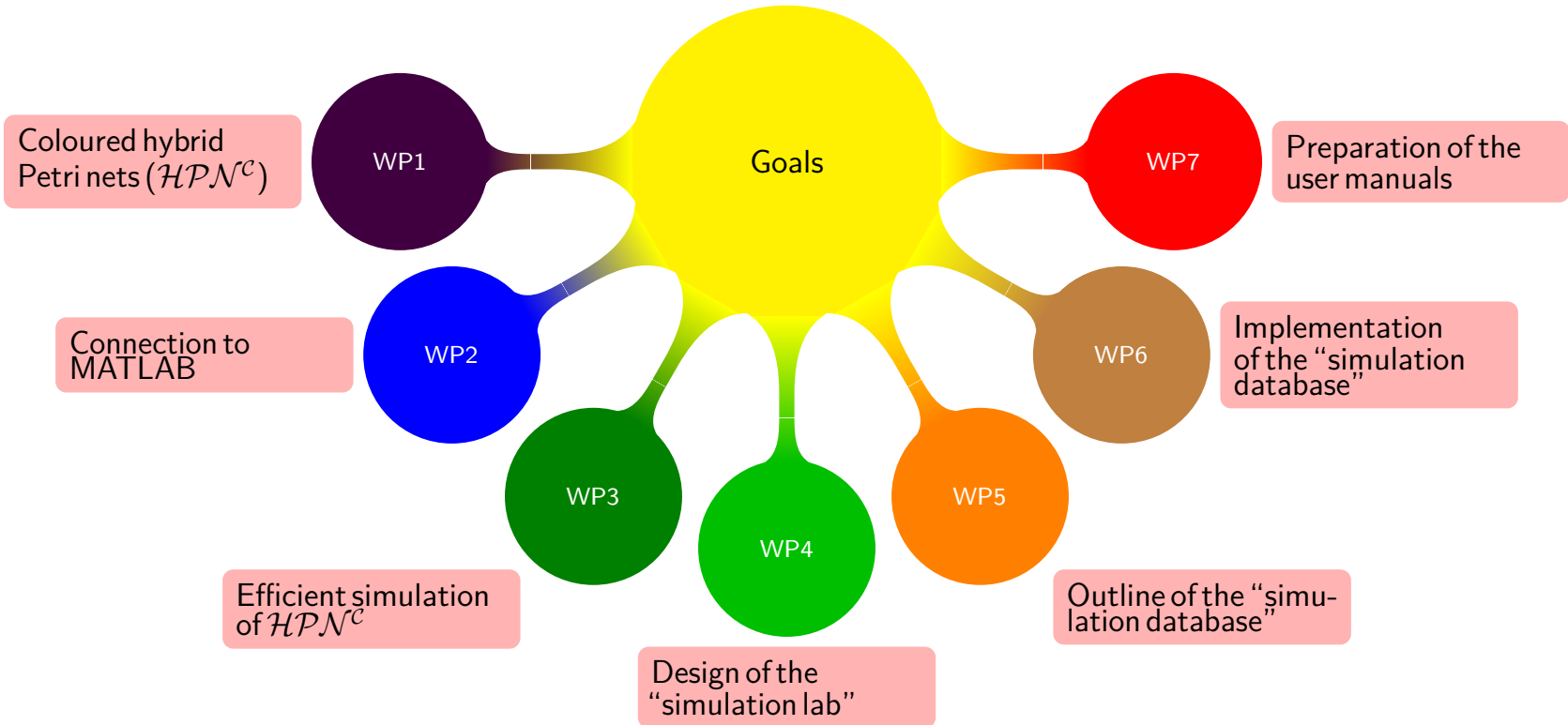
Department of Computer Science
Brandenburg University of Technology Cottbus

http://www-dssz.informatik.tu-cottbus.de

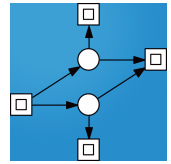
May 14, 2014

Work Packages





WP3



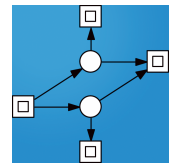
Efficient simulation of $\mathcal{HPN}^{\mathcal{C}}$

a. Predecessor WPs: BTU-WP1, OvGUM-WP1

b. Successor WPs: BTU-WP4

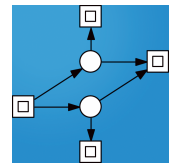
- Examination of the Petri net models with regard to parallelisation potential
- Investigation of optimisation possibilities and performance comparisons with alternative tools, i.e. StochKit2, Cain...

Tool comparison



- Starting point: How many stochastic simulation tools are available?
- SBML Software Summary: > 100 tools related to simulation.
- Selection criteria needed!

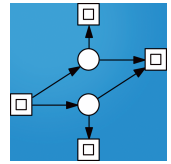
Tool comparison



Selection criteria

1. *Stochastic simulation tool.* There are different approaches to simulate biochemical reaction networks, such as deterministic or hybrid simulation, but we are only interested in stochastic simulation tools.
2. *Algorithms used by the Simulator.* There are a couple of algorithms for stochastic simulation, such as *direct method* [Gillespie 1977], first reaction method [Gillespie 1976], next reaction method [Gibson et al. 2000] and τ-leaping [Gillespie 2001]. The stochastic simulation tool must support at least the Gillespie's direct method.
3. *License type.* The use of the simulation tool must be free of charge.
4. *SBML support.* The tool must support SBML (Level 2), either directly, or indirectly by connection to a tool supporting SBML.

Tool comparison

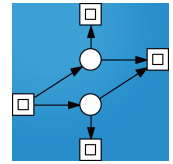


Candidates

1. Cain
2. Copasi
3. Dizzy
4. Marcie
5. Snoopy
6. StochKit2
7. StochPy

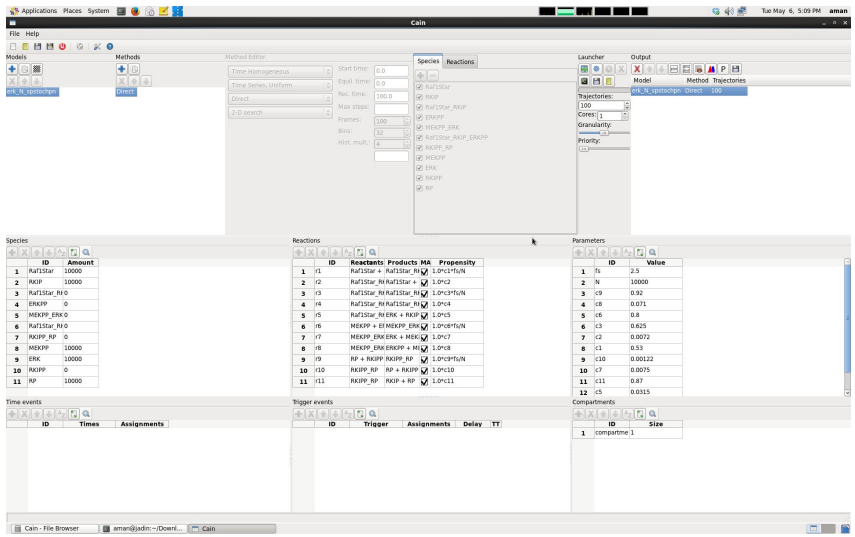
$\rightarrow$ may be extended

Candidates



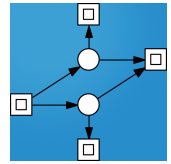
Cain

- Cain performs stochastic and deterministic simulations of chemical reactions.
- Caltech Center for Advanced Computing Research, Pasadena, California, US.
- Modelling paradigms: stochastic, deterministic.
- Simulation algorithm: direct method, next reaction method, τ-leaping.
- Interface: graphical user interface tool.
- File formats: SBML.



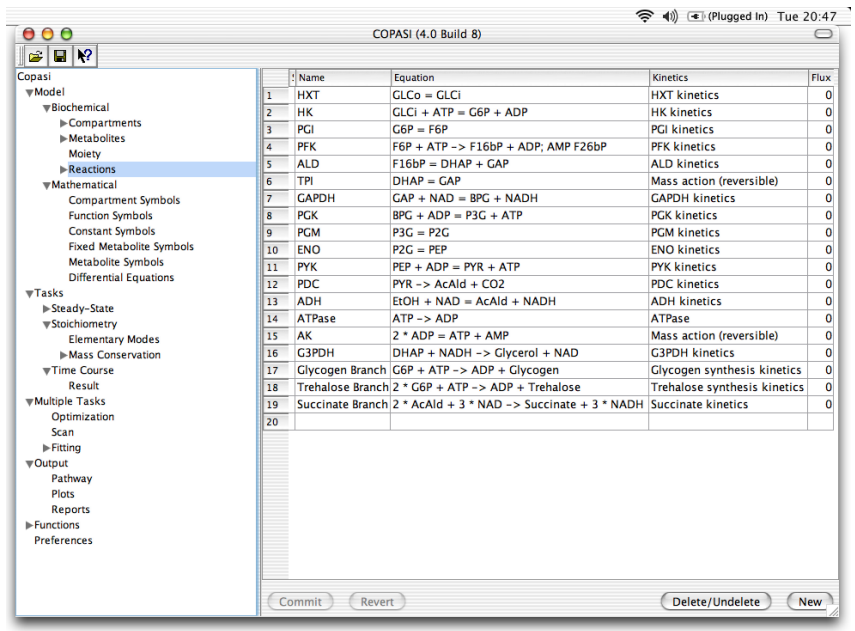
`http://cain.sourceforge.net/`

Candidates



Copasi

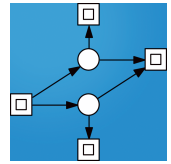
- COPASI is a software application for simulation and analysis of biochemical networks and their dynamics.
- Virginia Bioinformatics Institute, University of Heidelberg, University of Manchester.
- Modelling paradigms: stochastic, deterministic.
- Simulation algorithm: direct method, next reaction method, τ-leaping.
- Interface: GUI & CL tool.
- File formats: SBML L2.



	Name	Equation	Kinetics	Flux
1	HXT	GLCo = GLCi	HXT kinetics	0
2	HK	GLCi + ATP = G6P + ADP	HK kinetics	0
3	PGI	G6P = F6P	PGI kinetics	0
4	PFK	F6P + ATP -> F16bP + ADP; AMP F26bP	PFK kinetics	0
5	ALD	F16bP = DHAP + GAP	ALD kinetics	0
6	TPI	DHAP = GAP	Mass action (reversible)	0
7	GAPDH	GAP + NAD = BPG + NADH	GAPDH kinetics	0
8	PGK	BPG + ADP = P3G + ATP	PGK kinetics	0
9	PGM	P3G = P2G	PGM kinetics	0
10	ENO	P2G = PEP	ENO kinetics	0
11	PYK	PEP + ADP = PYR + ATP	PYK kinetics	0
12	PDC	PYR -> AcAld + CO2	PDC kinetics	0
13	ADH	EtOH + NAD = AcAld + NADH	ADH kinetics	0
14	ATPase	ATP -> ADP	ATPase	0
15	AK	2 * ADP = ATP + AMP	Mass action (reversible)	0
16	G3PDH	DHAP + NADH -> Glycerol + NAD	G3PDH kinetics	0
17	Glycogen Branch	G6P + ATP -> ADP + Glycogen	Glycogen synthesis kinetics	0
18	Trehalose Branch	2 * G6P + ATP -> ADP + Trehalose	Trehalose synthesis kinetics	0
19	Succinate Branch	2 * AcAld + 3 * NAD -> Succinate + 3 * NADH	Succinate kinetics	0
20				

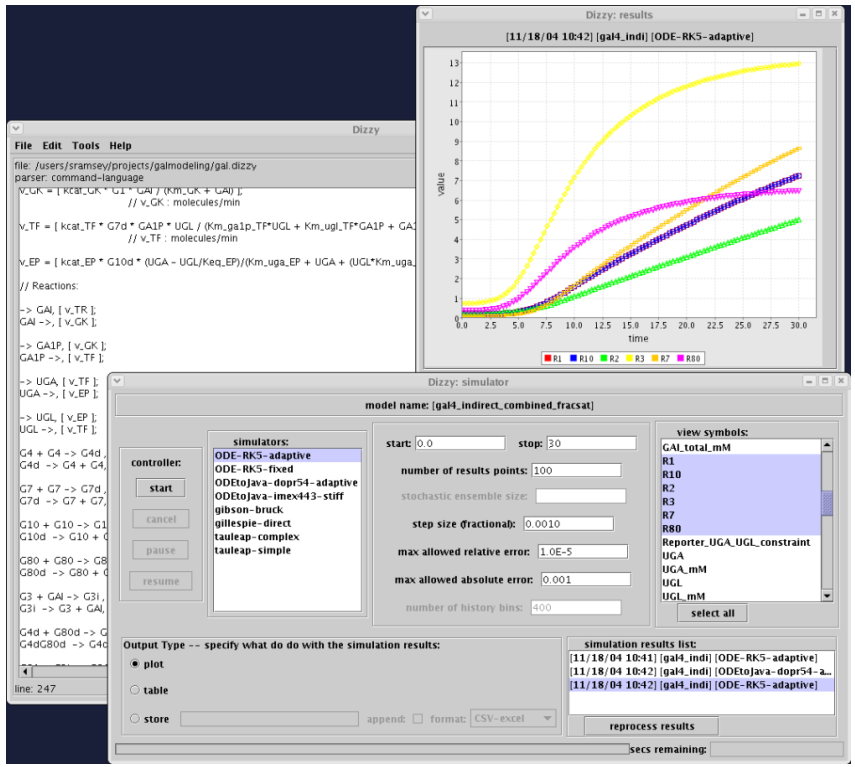
`http://www.copasi.org`

Candidates



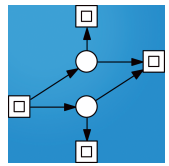
Dizzy

- Dizzy is a chemical kinetics stochastic simulation software package written in Java.
- Institute for System Biology, Seattle, Washington, US.
- Modelling paradigms: stochastic, deterministic.
- Simulation algorithm: direct method, next reaction method, τ-leaping.
- Interface: graphical user interface tool.
- File formats: SBML L1V2.



`http://magnet.systemsbiology.net/software/Dizzy/`

Candidates

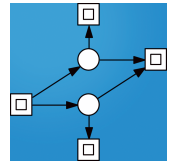


Marcie

- Marcie is a tool for qualitative and quantitative analysis of Generalized Stochastic Petri nets with extended arcs.
- BTU Cottbus, Germany.
- Modelling paradigms: stochastic.
- Simulation algorithm: direct method, next reaction method, τ-leaping.
- Interface: command line tool.
- File formats: ANDL, PNML.

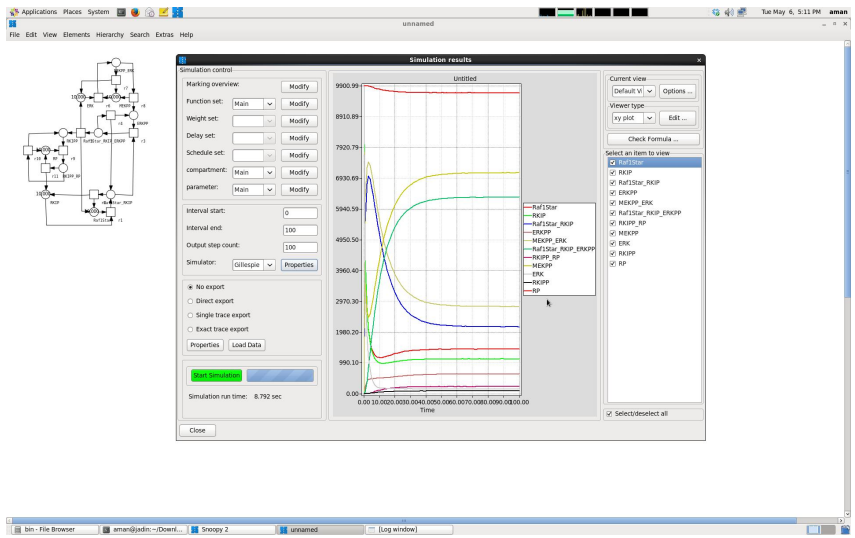
`http://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Marcie`

Candidates



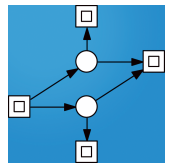
Snoopy

- Snoopy is a software tool to design, animate and simulate hierarchical graphs, among others Petri nets.
- BTU Cottbus, Germany.
- Modelling paradigms: stochastic, deterministic and hybrid.
- Simulation algorithm: direct method.
- Interface: graphical user interface tool.
- File formats: SBML L2, ANDL.



`http://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Snoopy`

Candidates

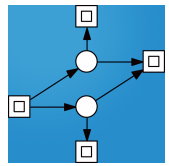


StochKit2

- StochKit is an extensible stochastic simulation framework developed in C++.
- University of California, Santa Barbara, US.
- Modelling paradigms: stochastic.
- Simulation algorithm: direct method, next reaction method, τ-leaping.
- Interface: command line tool.
- File formats: SBML L2.

`http://engineering.ucsb.edu/~cse/StochKit`

Candidates

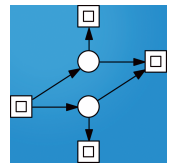


StochPy

- StochPy is a versatile modeling package for stochastic simulation.
- VU University Amsterdam, NL.
- Modelling paradigms: stochastic.
- Simulation algorithm: direct method, next reaction method, τ-leaping.
- Interface: command line tool.
- File formats: SBML L2.

`http://stochpy.sourceforge.net/`

Test suite



- (colored) stochastic Petri net
- standard arcs, no read, inhibitor, modifier arcs
- mass-action kinetics
- scalable

stochastic Petri nets

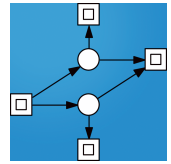
- RKIP inhibited ERK Pathway
- Mitogen-activated Protein Kinase
- Angiogenesis

colored stochastic Petri nets

- Prey Predator
- Gradient
- Repressilator

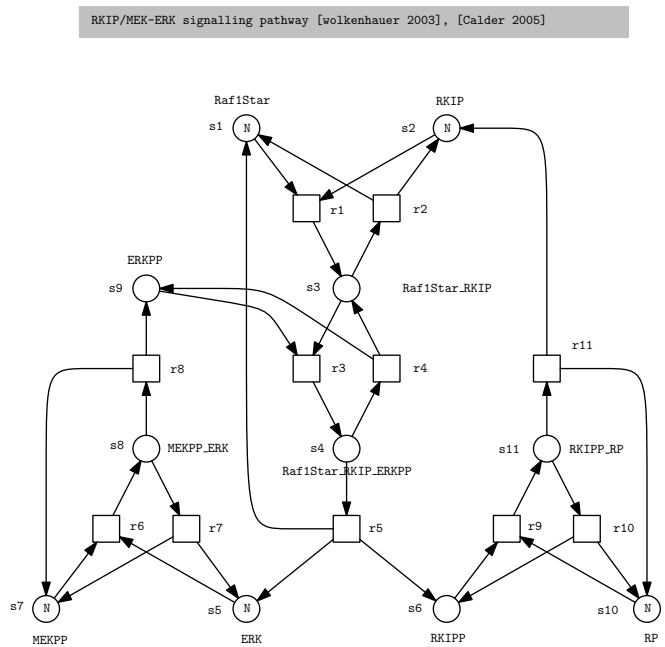
$\rightarrow$ may be extended

Test suite

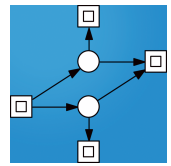


RKIP inhibited ERK Pathway [Gilbert et al. 2006]

- # places: 11
- # transitions: 11
- # arcs: 34
- scalable in N – the initial number of tokens on the places ERK, MEKPP, Raf1Star, RKIP and RP

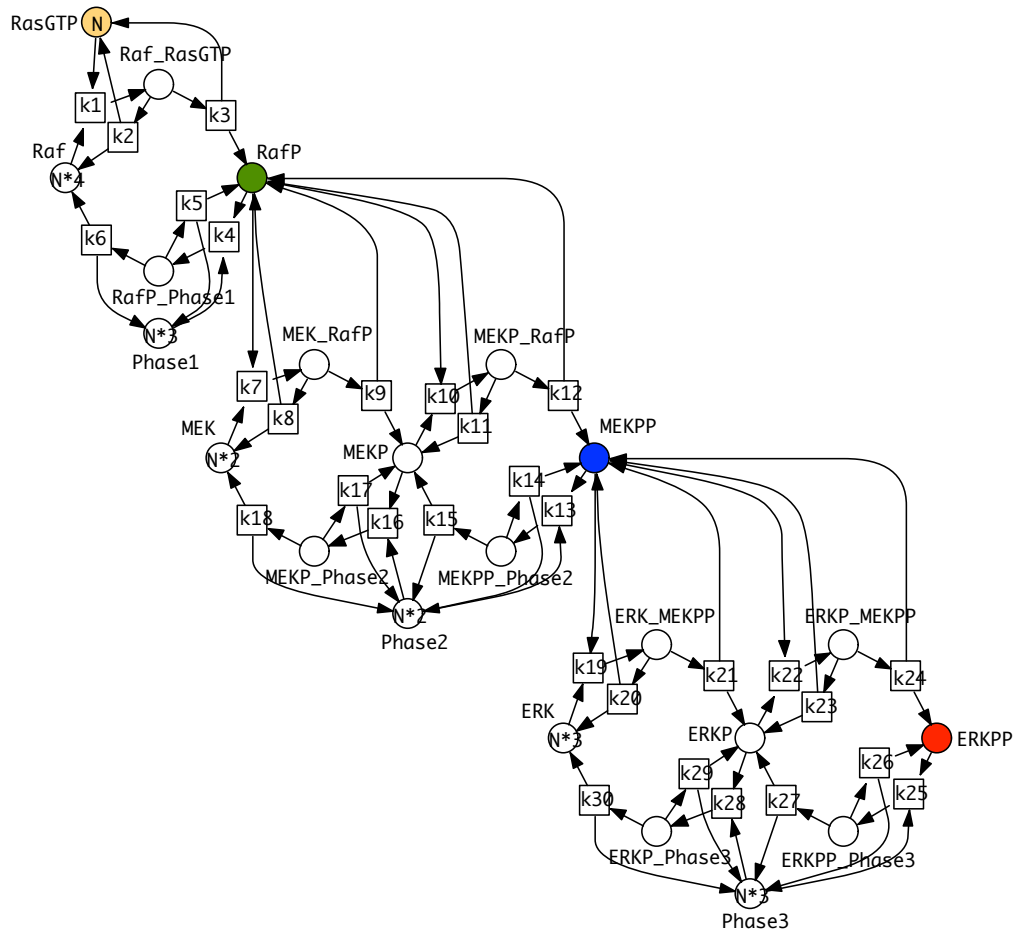


Test suite

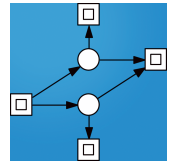


Mitogen-activated Protein Kinase [Levchenko et al. 2000], [Heiner et al. 2008]

- # places: 22
- # transitions: 30
- # arcs: 90
- scalable in N – the multiplier of initial number of tokens on the places Raf, RasGTP, RafP_Phase1, MEKP_Phase2, ERk, ERKP_Phase3

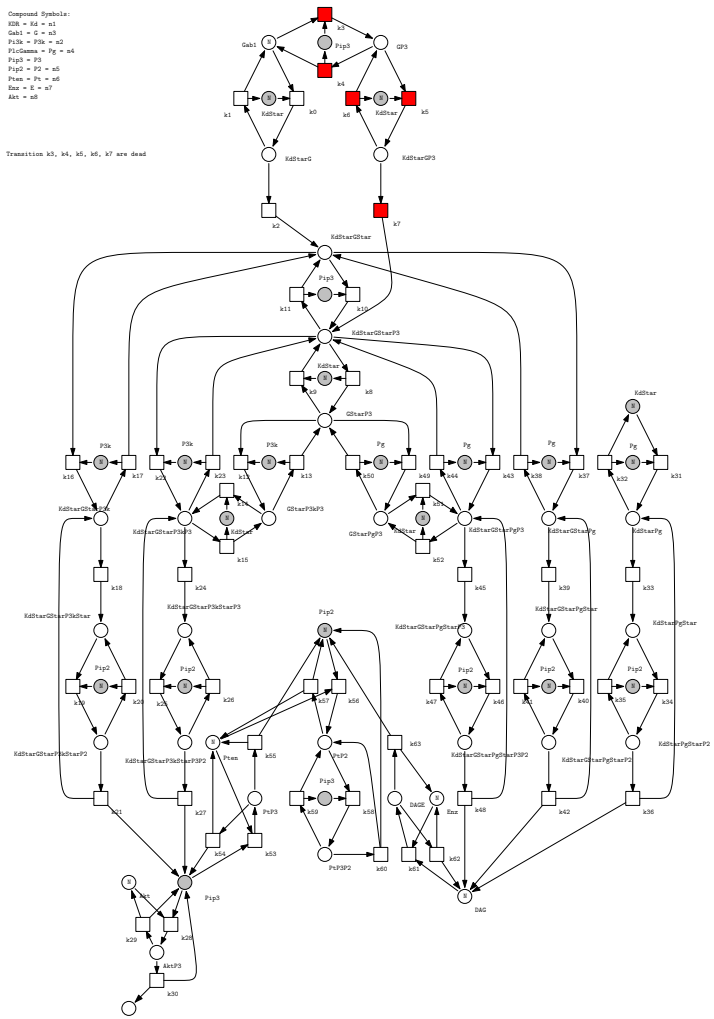


Test suite

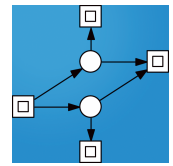


Angiogenesis [Napione et al. 2009]

- # places: 39
- # transitions: 64
- # arcs: 185
- scalable in N – the initial number of tokens on the places Gab1, KdStar, P3k,Pg, Pip2, DAG, Enz and Akt
- *dead states*

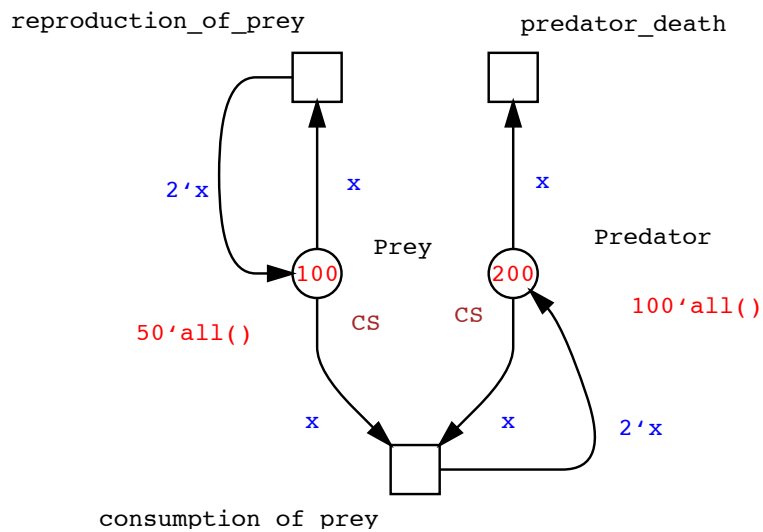


Test suite

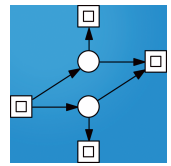


Prey Predator

- # places: $2 * N$
- # transitions: $3 * N$
- # arcs: $6 * N$
- scalable by # of places, transitions and arcs
- *unbounded*

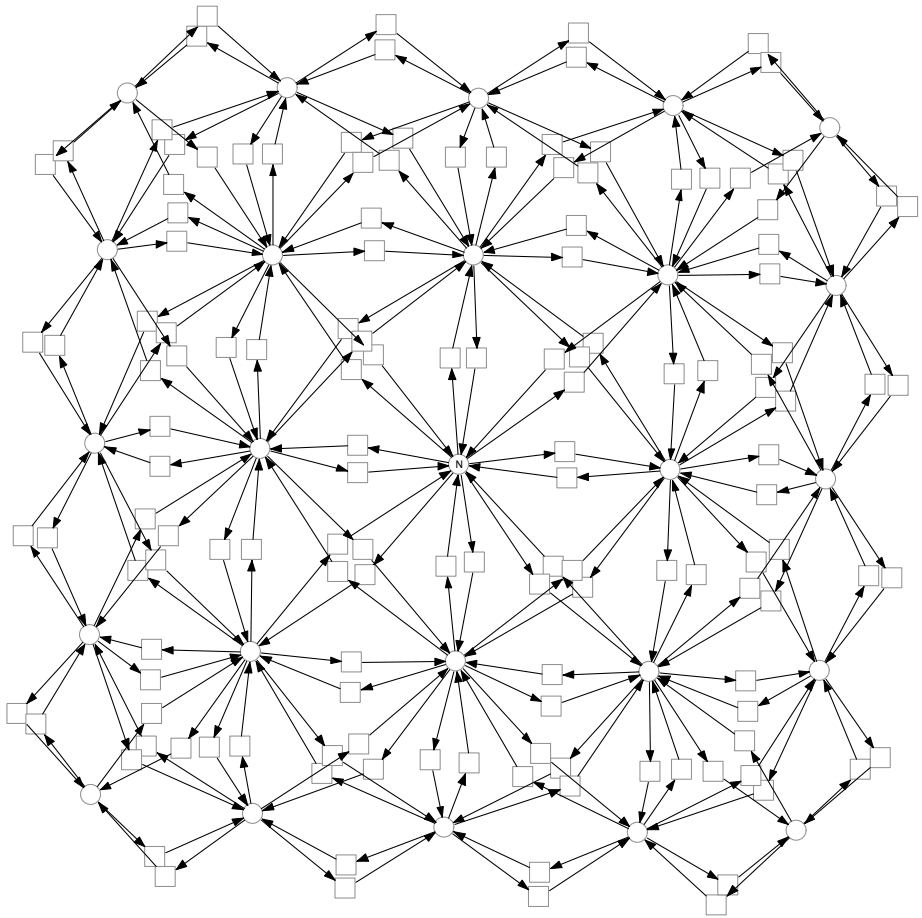


Test suite

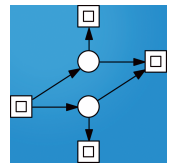


Gradient [Gilbert et al. 2013]

- # places: N^2
- # transitions: $8 * N^2 - 12 * N + 4$
- # arcs: $2 * |T|$
- scalable by # of places, transitions and arcs

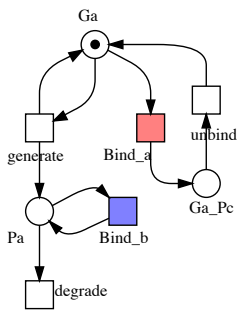


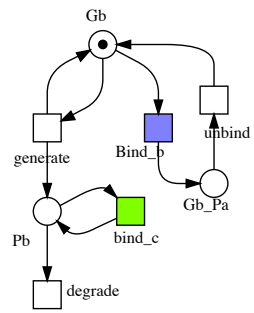
Test suite

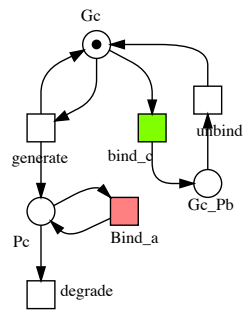


Repressilator [Liu 2012]

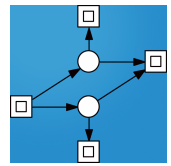
- # places: $3 * N$
- # transitions: $4 * N$
- # arcs: $10 * N$
- scalable by # of places, transitions and arcs







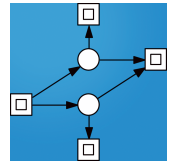
Simulation parameters



The assumptions and constraints while performing simulation are:

- *Simulation algorithm:* direct method.
- *Simulation results:* mean token value of the places (species).
- *Simulation time:* from 0 time units to t time units (model specific).
- *Simulation runs:* 1; 100; 10 000; 1 000 000
- *Threshold:* max. 1h run time
- *Threads:* 1 and 4 threads
- *Model parameter:* 4 different values of N
- *Experiment:* a particular value of N, number of runs and threads
  $\rightarrow$ each experiment performed 10 times
  $\rightarrow$ 320 experiments per model and tool
  $\rightarrow$ worst case 320h, usually 12h

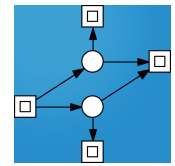
Comparison criteria



Performance measures

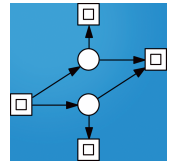
- average run time per experiment
- peak memory consumption per experiment
- simulation results
- multi-threading coefficient

Setup



- Dell Precision T7400
- CPU: Intel® Xeon® CPU E5440 @ 2.83GHz
- RAM: 4x1024MB DDR2 FB-DIMM @ 667MHz
- OS: CentOS release 6.5 (64bit)

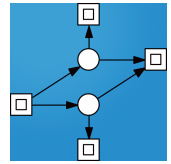
Results

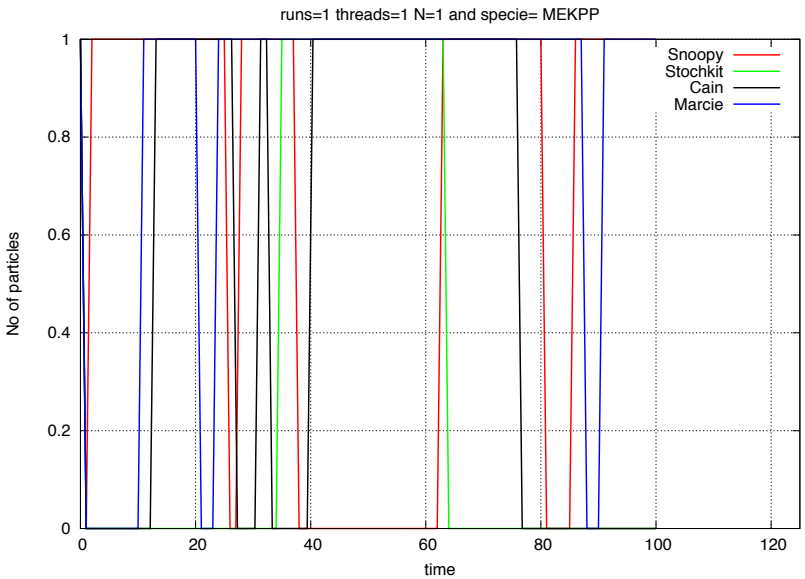


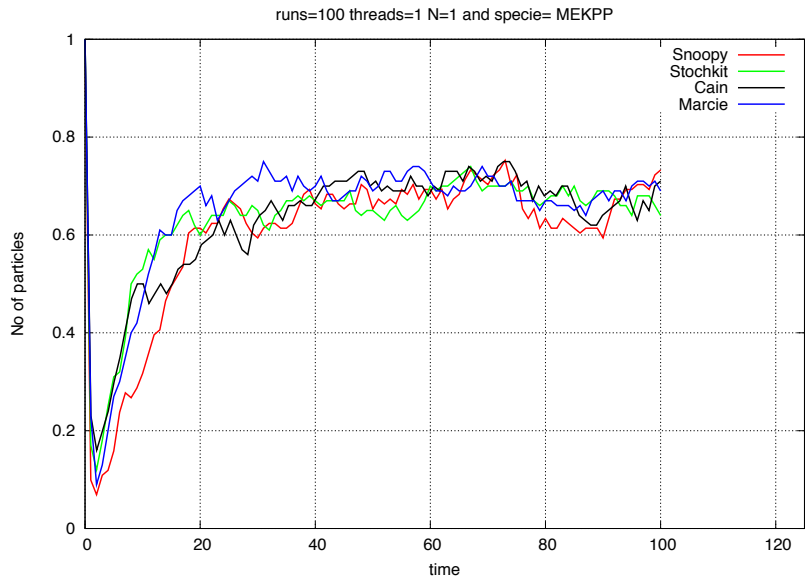
RKIP inhibited ERK Pathway

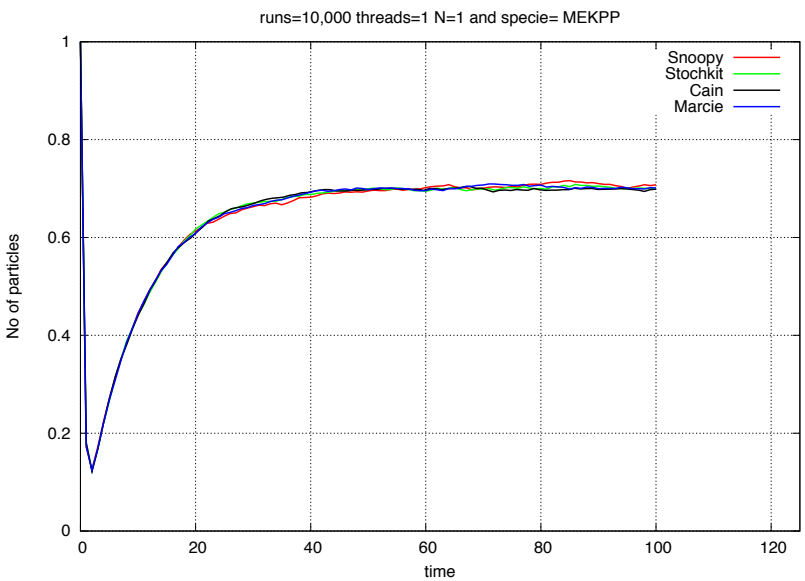
- *Model parameter:* $N \in \{1, 100, 10000, 1000000\}$
- *Tools:* Cain, Marcie, Snoopy, StochKit2

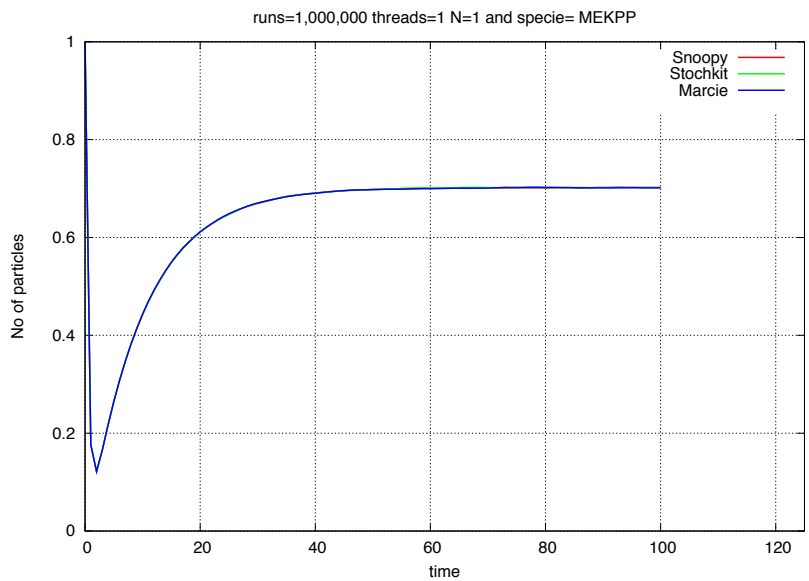
Results – ERK



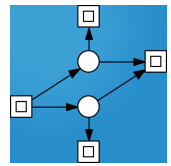

runs=1 threads=1 N=1 and specie= MEKPP
Snoopy
Stochkit
Cain
Marcie
No of particles
time
0
0.2
0.4
0.6
0.8
1
0
20
40
60
80
100
120


runs=100 threads=1 N=1 and specie= MEKPP
Snoopy
Stochkit
Cain
Marcie
No of particles
time
0
0.2
0.4
0.6
0.8
1
0
20
40
60
80
100
120


runs=10,000 threads=1 N=1 and specie= MEKPP
Snoopy
Stochkit
Cain
Marcie
No of particles
time
0
0.2
0.4
0.6
0.8
1
0
20
40
60
80
100
120


runs=1,000,000 threads=1 N=1 and specie= MEKPP
Snoopy
Stochkit
Marcie
No of particles
time
0
0.2
0.4
0.6
0.8
1
0
20
40
60
80
100
120

Results

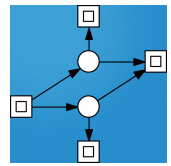


Snoopy, average runtime (in sec) for ERK.

N	threads	runs			
		1	100	10,000	1,000,000
1	1	0.0002	0.0018	0.1658	16.8030
	4	1.0003	1.0014	1.0079	4.9084
100	1	0.0009	0.0989	8.9830	884.4062
	4	1.0005	1.0084	3.0072	233.9180
10,000	1	0.0898	8.8005	875.7929	†
	4	1.0004	3.0085	231.8182	†
1,000,000	1	9.0178	889.899	†	†
	4	9.4008	243.3228	†	†

† runtime > 1h

Results

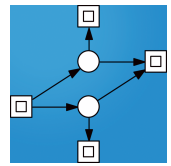


Snoopy, peak memory consumption (in KB) for ERK.

N	threads	runs			
		1	100	10,000	1,000,000
1	1	37,576	37,752	37,904	37,988
	4	37,512	37,767	37,824	39,604
100	1	37,580	39,748	39,507	39,968
	4	40,196	40,160	39,900	39,932
10,000	1	39,980	39,752	41,928	†
	4	40,032	40,112	39,568	†
1,000,000	1	39,600	39,460	†	†
	4	39,796	39,004	†	†

† runtime > 1h

Results

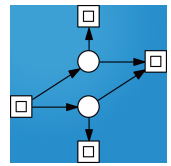


Marcie, average runtime (in sec) for ERK.

N	threads	runs			
		1	100	10,000	1,000,000
1	1	0.0	0.0	0.0	13.0
	4	0.0	0.0	0.0	3.0
100	1	0.0	0.0	6.0	617.6
	4	0.0	0.0	1.0	162.5
10,000	1	0.0	6.0	620.3	†
	4	0.0	1.0	161.8	†
1,000,000	1	6.0	621.2	†	†
	4	6.0	163.5	†	†

† runtime > 1h

Results

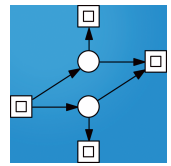


Marcie, peak memory consumption (in KB) for ERK.

N	threads	runs			
		1	100	10,000	1,000,000
1	1	0,000	0,000	5,172	3,128
	4	3,126	0,000	5,324	5,324
100	1	0,000	5,172	3,132	3,156
	4	0,000	5,324	5,328	5,360
10,000	1	3,124	3,152	3,156	†
	4	3,128	5,328	5,356	†
1,000,000	1	3,148	3,160	†	†
	4	3,128	5,364	†	†

† runtime > 1h

Results



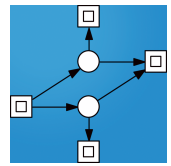
CAIN, average runtime (in sec) for ERK.

N	threads	runs			
		1	100	10,000	1,000,000
1	1	0.0222	0.1160	8.1655	$*$
	4	0.0231	0.0943	4.2285	$*$
100	1	0.0231	0.1426	10.9276	$*$
	4	0.0239	0.1048	5.1622	$*$
10,000	1	0.0315	1.8656	183.8095	$*$
	4	0.0338	0.6837	60.1041	$*$
1,000,000	1	1.7569	173.7179	†	$*$
	4	1.7658	56.9456	†	$*$

† runtime > 1h

$*$ tool crash while performing simulation

Results



CAIN, peak memory consumption (in KB) for ERK.

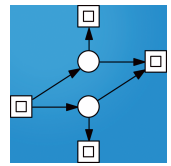
Table: CAIN peak memory consumption (in KB) for ERK.

N	threads	runs			
		1	100	10,000	1,000,000
1	1	110,048	113,544	201,795	$*$
	4	110,520	117,912	109,125	$*$
100	1	112,196	112,960	199,904	$*$
	4	110,944	117,316	203,208	$*$
10,000	1	110,080	114,324	199,728	$*$
	4	110,468	118,568	202,600	$*$
1,000,000	1	111,200	115,408	†	$*$
	4	111,792	118,192	†	$*$

† runtime > 1h

$*$ tool crash while performing simulation

Results

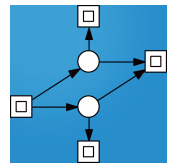


StochKit2, average runtime (in sec) for ERK.

N	threads	runs			
		1	100	10,000	1,000,000
1	1	0.0080	0.0134	0.4421	42.7381
	4	0.0080	0.0172	0.1284	11.5534
100	1	0.0080	0.0425	3.3970	338.0346
	4	0.0083	0.0276	0.9099	88.4614
10,000	1	0.0380	2.9934	294.4733	†
	4	0.0381	0.8301	77.7298	†
1,000,000	1	2.9665	296.6058	†	†
	4	2.9649	80.5857	†	†

† runtime > 1h

Results

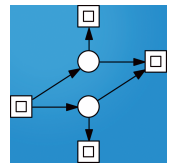


StochKit2, peak memory consumption (in KB) for ERK.

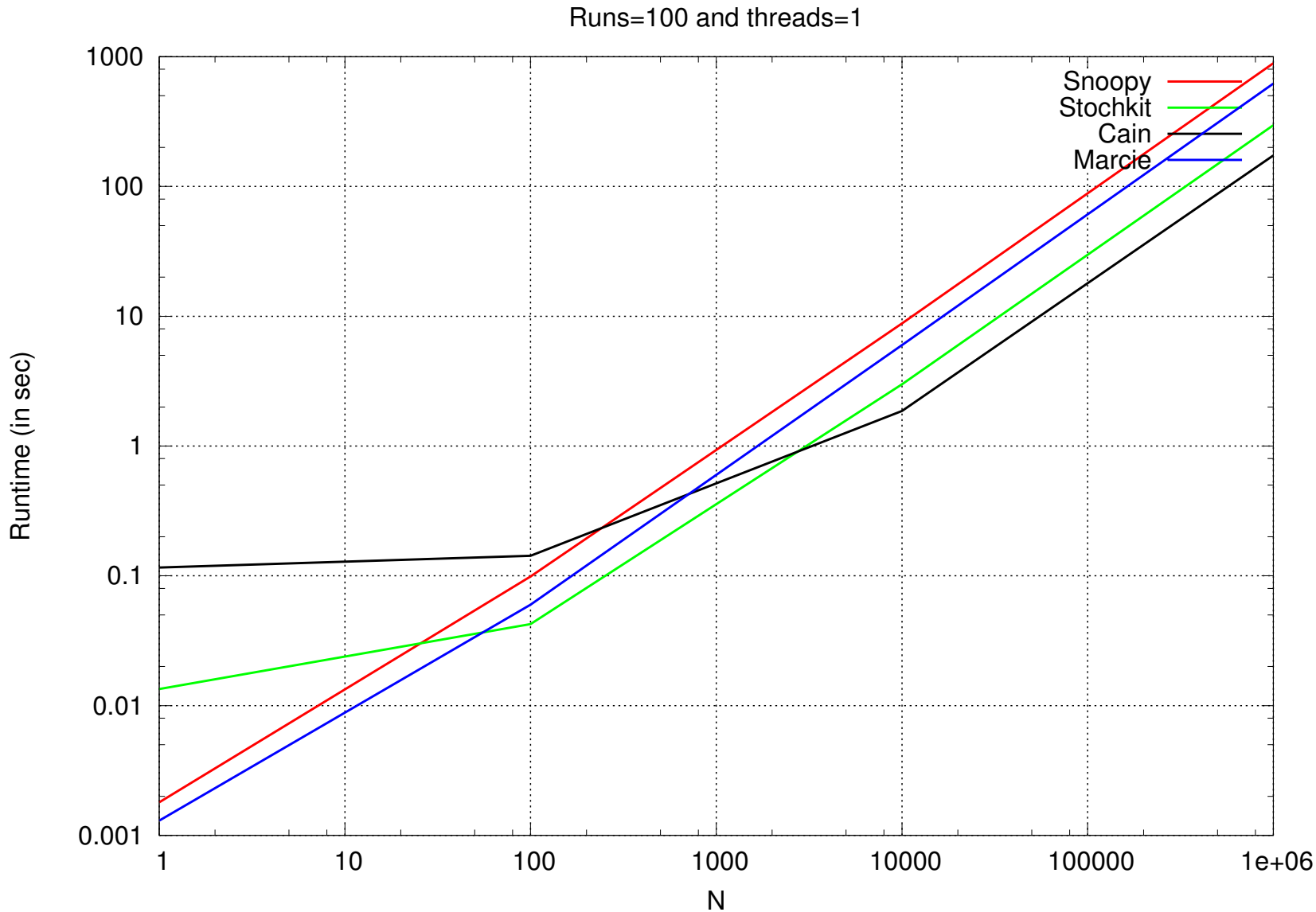
N	threads	runs			
		1	100	10,000	1,000,000
1	1	2,748	2,752	9,904	9,924
	4	2,748	2,748	20,560	24,800
100	1	2,748	2,748	9,982	9,924
	4	2,748	2,748	24,786	26,832
10,000	1	2,748	9,904	9,932	†
	4	2,748	24,800	24,876	†
1,000,000	1	9,964	9,968	†	†
	4	9,916	24,920	†	†

† runtime > 1h

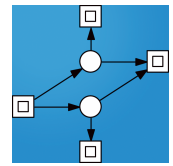
Result



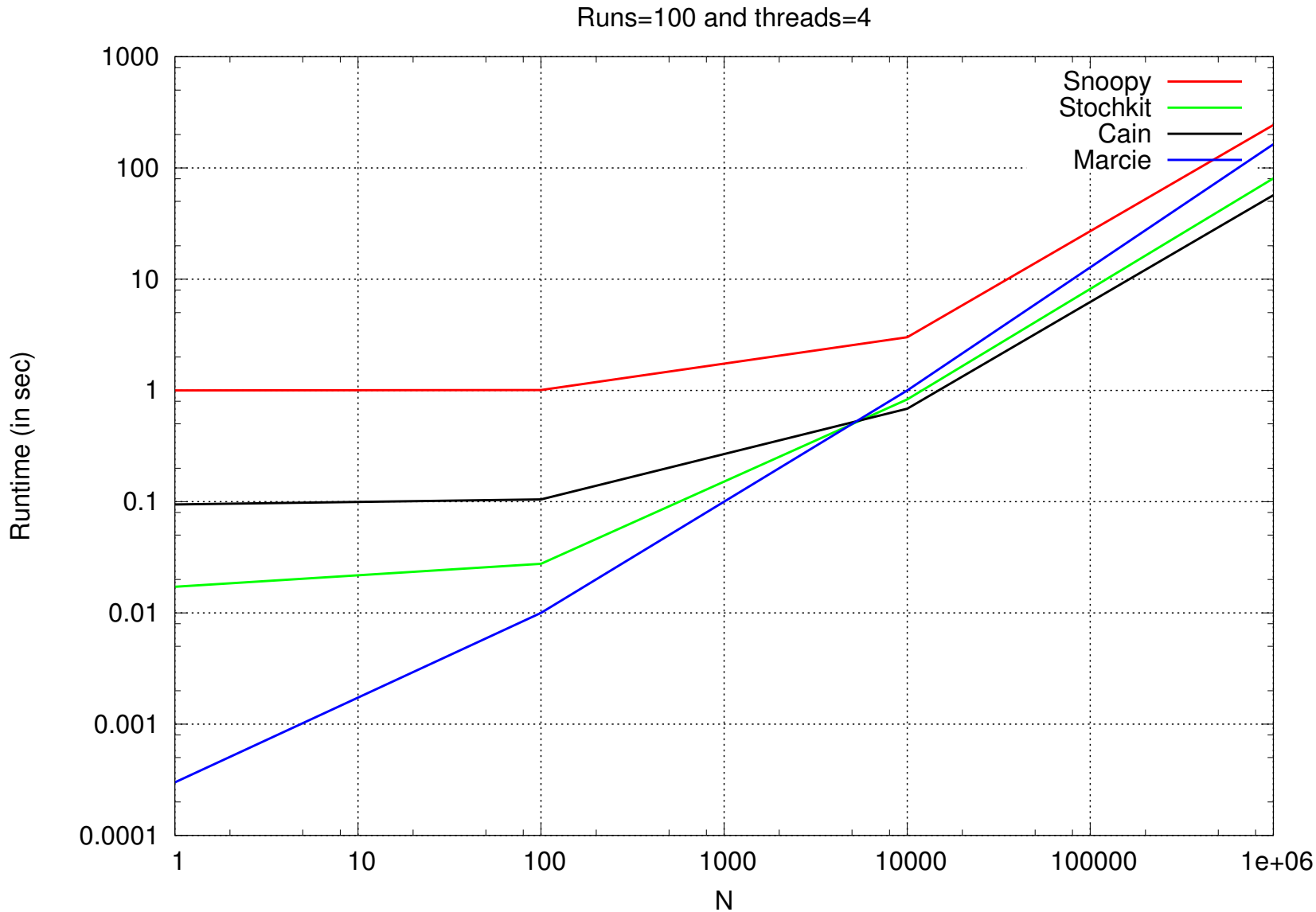
Comparison: average run time



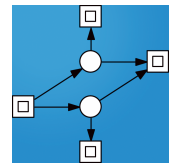
Result



Comparison: average run time



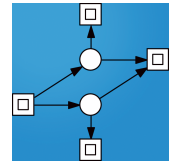
Results



Multithreading Coefficient

Tool	Threading Coefficient	Variation
CAIN	3.0544	0.01
MARCIE	3.8112	0.02
SNOOPY	3.7387	0.07
STOCHKIT	3.7474	0.07

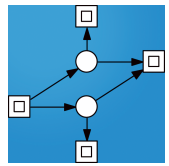
Conclusions



So far

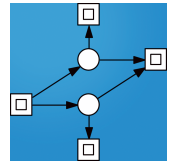
1. *CAIN* fastest, highest memory consumption (leak?)
2. *StochKit2* very fast
3. *Marcie* fast, lowest memory consumption
4. *Snoopy*

Milestones

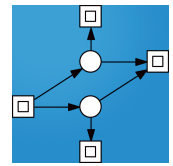


	2013				2014				2015			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
WP1		M1										
WP2				M2								
WP3						M3						
WP4								M4				
WP5								M4				
WP6												M5
WP7												M6

Next steps...



- finish tool comparison
- go on with WP 4 & 5



Thank you for your attention!