FUNDACIÓN RAMÓN ARECES

Jornada: El impacto de la Nube y el Big Data en la Ciencia
Ciencias de la Vida y de la Materia
Madrid, 21 de marzo de 2013

Transformando Big Data en conocimiento: gotas de sistemas biológicos en la Nube

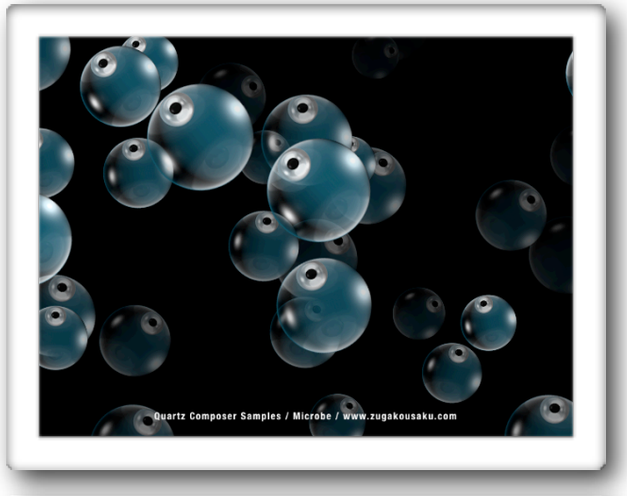
Turning Big data into knowledge: systems biology droplets in the cloud

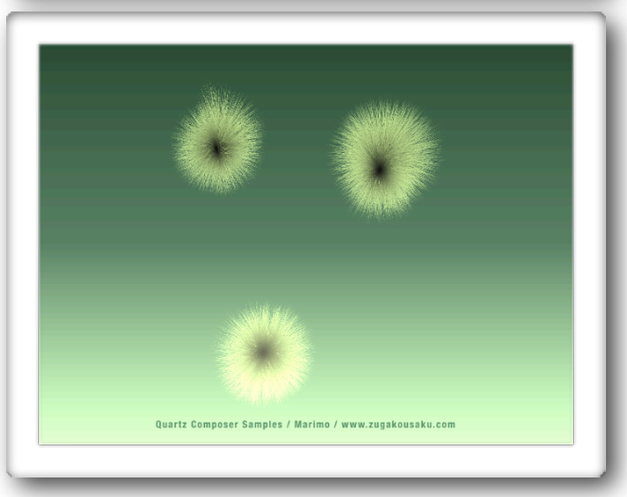
Marco Aldinucci,
Computer Science Dept., University of Torino, Italy

BioBITs

Rationale

Extract information from the apparent chaos. In parallel, (also) exploiting the cloud. Focus on app development not on middleware.





noise 70%

restored

original

M. Aldinucci et al. A parallel edge preserving algorithm for salt and pepper image denoising. Intl. IEEE Conference on Image Processing Theory Tools and Applications (IPTA). 2012

Questions answered at this point in time

- What is a cloud?
  - how a cloud look-like at the hardware level
    - which nice pictures :-)
  - what services are available in the cloud
- What is big data?
  - which scientific domains are likely to produce large amount of data
- Parallel computing
  - the urgency of novel programming models

There are not free lunches in nature

- Multiplying by N the number of virtual cores and disks does not automatically turns in dividing computing time by N

- Cloud (oversimplified)
  - **cluster** of **multi-core** and **many-core**, with an a number of storage architectures
- What is the cost, the complexity to support so many different architecture styles?
  - and what is the efficiency of the application?

More questions

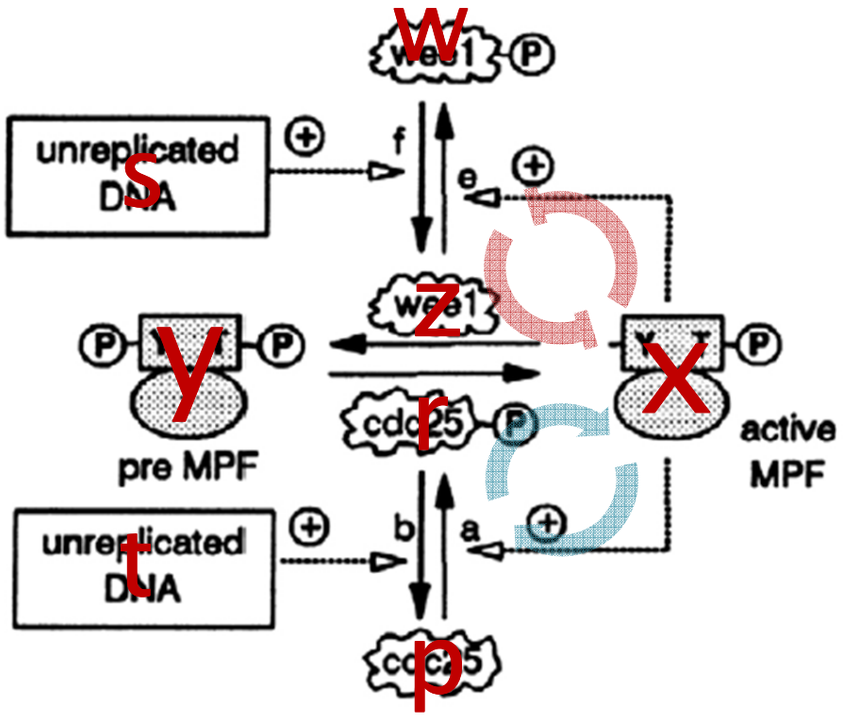
- What cloud usage model?
  - Grid-like (PaaS, IaaS)
    - not appealing for bio-scientist
    - not interactive, requires IT expertise
      - enqueue tasks, wait for your turn, run the simulation, store data, transfer data, analyze data locally, interpret results, if something wrong, restart from begin
    - Offload much of the complexity on the app programmer
  - As a service (SaaS)
    - the whole simulation-analysis pipeline should be moved to the cloud
    - The whole software pipeline and all tools should be ported and optimized for the cloud

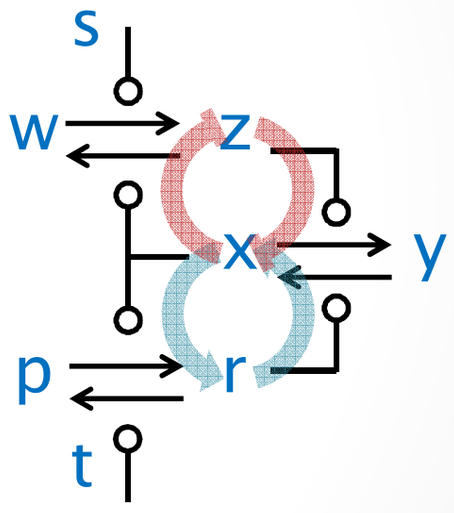
Example: systems biology

Systems Biology & Gillespie's algorithm

- Traditionally studied with continuous Ordinary Differential Equations (ODE)
  - bulk reactions, i.e. average behavior
- Alternative approach: discrete and stochastic simulation of a systems via explicit simulation of each reaction
  - Gillespie algorithm
  - More informative than ODE
    - multi-stability, divergent or rare behaviors, peaks, ...
  - More computing demanding

… a switch, when formalised

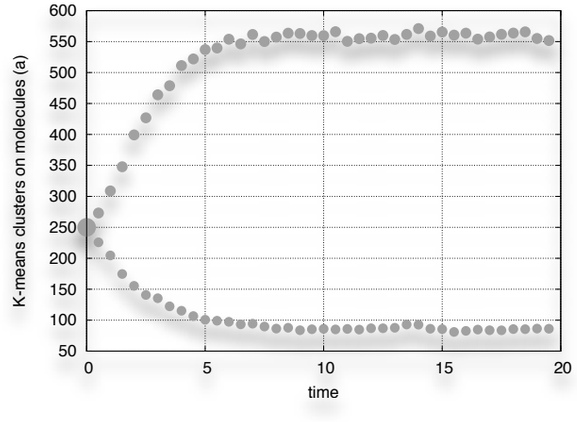




Courtesy of Luca Cardelli

On Switches and Oscillators Program Equivalence in Biology?

http://lucacardelli.name

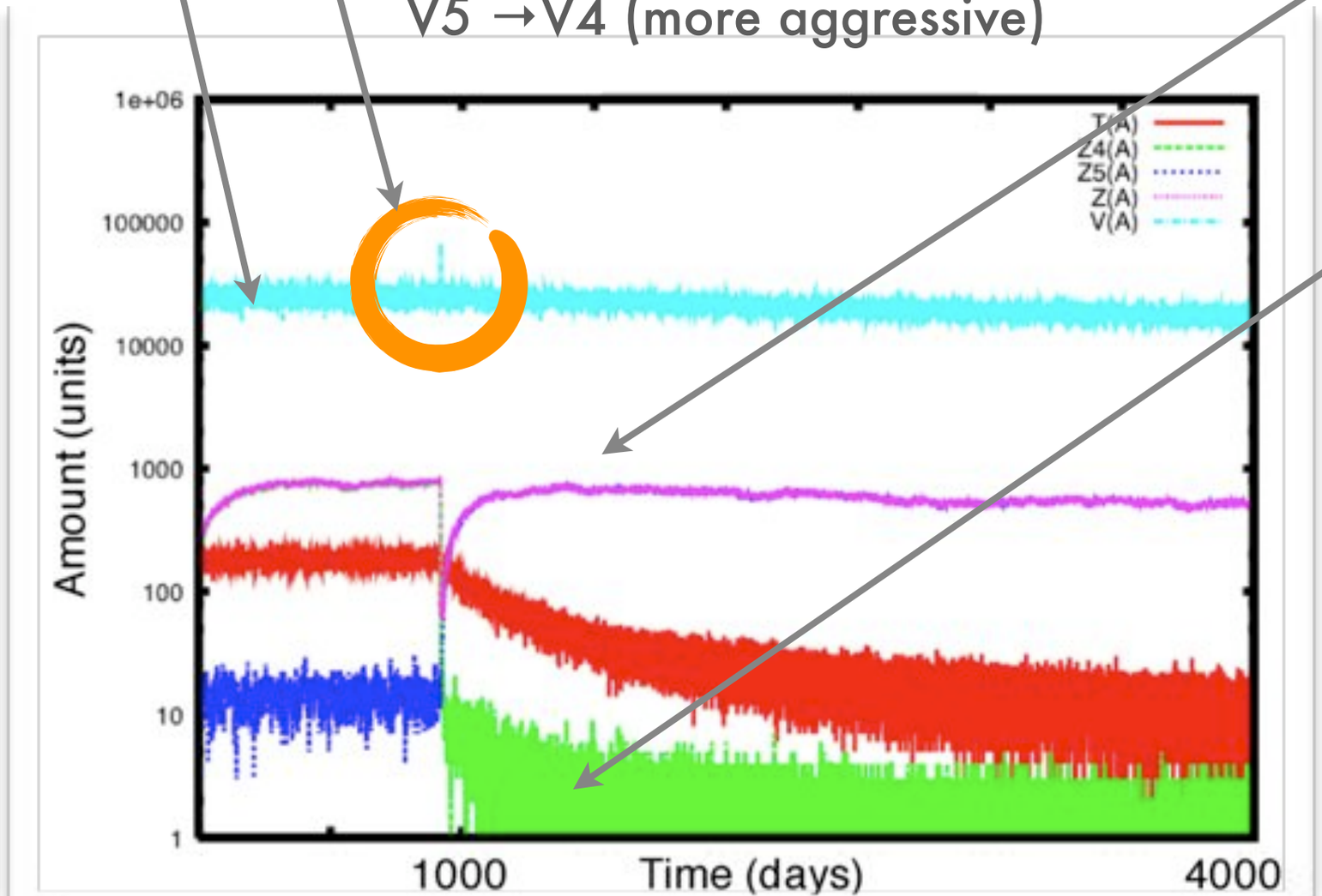


$$\top : a\ c \xmapsto{10} c\ b \qquad \top : c\ a \xmapsto{10} a\ b \qquad \top : b\ a \xmapsto{10} a\ a \qquad \top : b\ c \xmapsto{10} c\ c$$

Example: HIV and immune response (progression to AIDS)

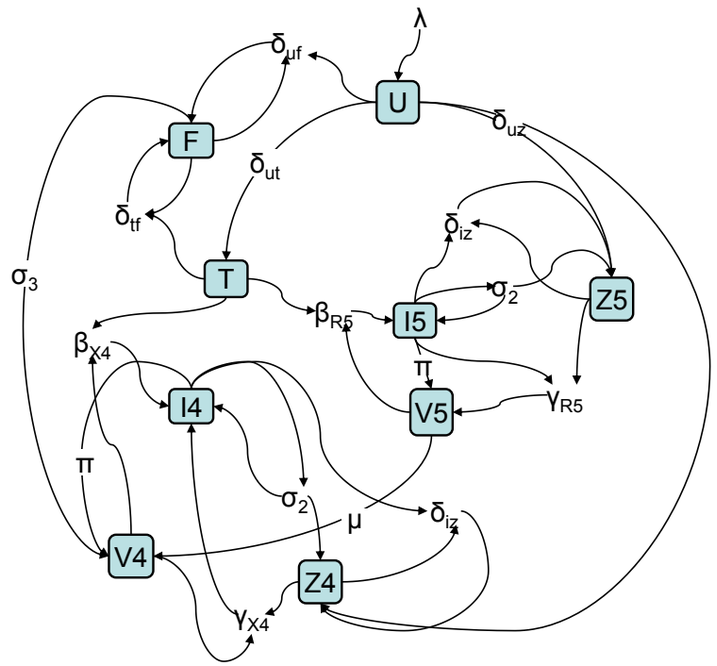
V virus, all phenotypes (V4,V5)

the spike suggests the mutation
V5 →V4 (more aggressive)



immune response Z=Z4+Z5
remain stable (but for the peak).

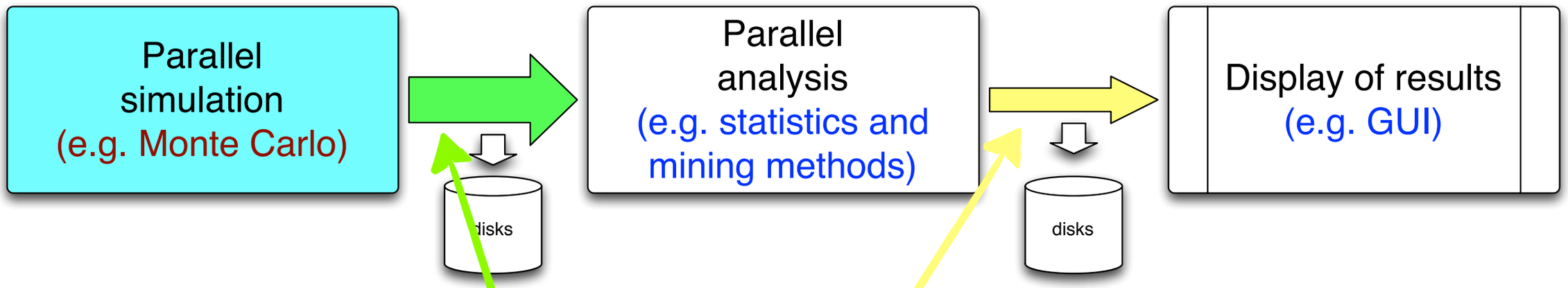
now Z4 decrease and
Z5 increase
i.e. HIV is turning into
AIDS more rapidly



M. Aldinucci, A. Bracciali, P. Liò, A. Sorathiya, and M. Torquati. StochKit-FF: Efficient systems biology on multicore architectures. In High Performance Bioinformatics and Biomedicine (HiBB), vol. 6586 of LNCS, 2011. Springer.

- Peaks are informative events,
  - virus mutation triggers AIDS progression
  - hardly detected with ODEs
- high resolution required to detect spikes,
  - each trajectory can be over 6G Bytes of data
  - The more precision the more data
- and thousands of trajectories are needed
  - compute everything, save everything, move and join all data, analyse all data, then get first results
  - often to discover parameters was wrong ...

- It is Monte Carlo,
  - well understood
  - easy to parallelize
- it is Monte Carlo w Markov Chains models (CTMC)
  - compute time ≠ simulation time
  - compute time for different trajectories heavily unbalanced
- it is Monte Carlo and data analysis
  - **data is big, analysis very expensive and it typically starts after the simulation**
  - the whole workflow is perceived too “slow” by bio-scientists to be really useful



	Simulated time	Simulation data	Seq	Parallel (16 core Intel SandyBridge)

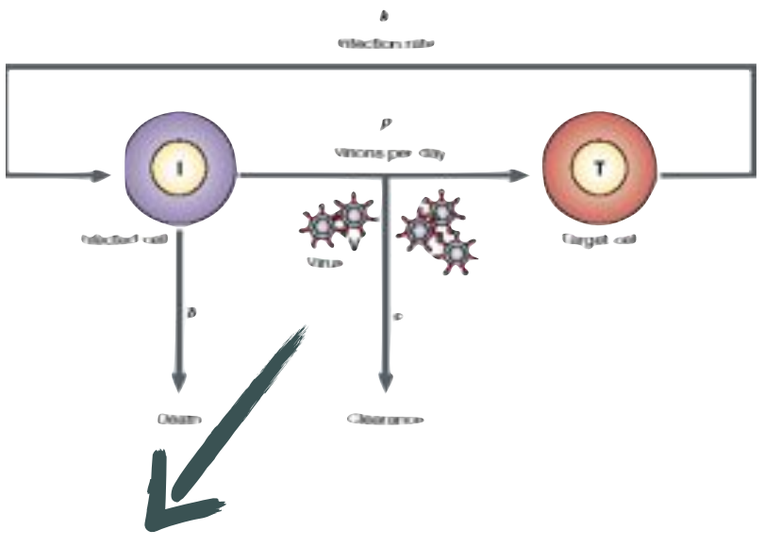
	time	resolution	raw data size	output size	MonteCarlo step latency	total time	total time	Throughput	speedup
Neurospora	1 month	~ 25 s	~8 GB	~6.5MB	600 ns	20 min	93 s	~20 MB/s	~16
Neurospora	4 days	~1 s	~80 GB	~65MB	1600 ns	60 min	~5 min	~280 MB/s	~16
Neurospora	1 month	~1 s	640 GB	520 MB	1600 ns	8 hours	~30 min	~280 MB/s	~16

what you need to store and re-read with off-line filtering

filtered data

(Neurospora) Leloup J, Gonze D, Goldbeter A: Limit cycle models for circadian rhythms based on transcriptional regulation in Drosophila and Neurospora. Journal of Biological Rhythms 1999, 14(6):433.

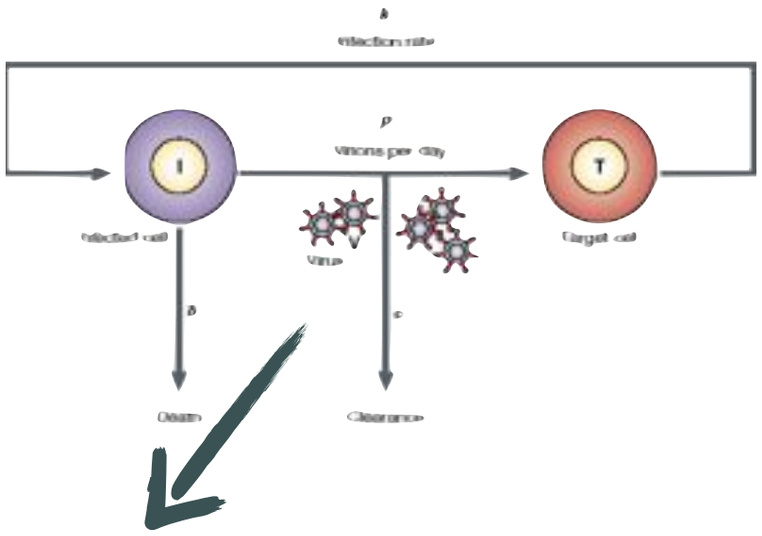
Data analysis is not for free



$$NH_3 \underset{k_2}{\overset{k_1}{\rightleftharpoons}} NH_4^+$$

$$(Fungus \rfloor NH_4^+ \;\; x) \overset{k_f}{\mapsto} NH_4^+ \;\; (Fungus \rfloor x)$$

Data analysis is not for free



In BioSims the whole trajectory is needed, easily GB of data

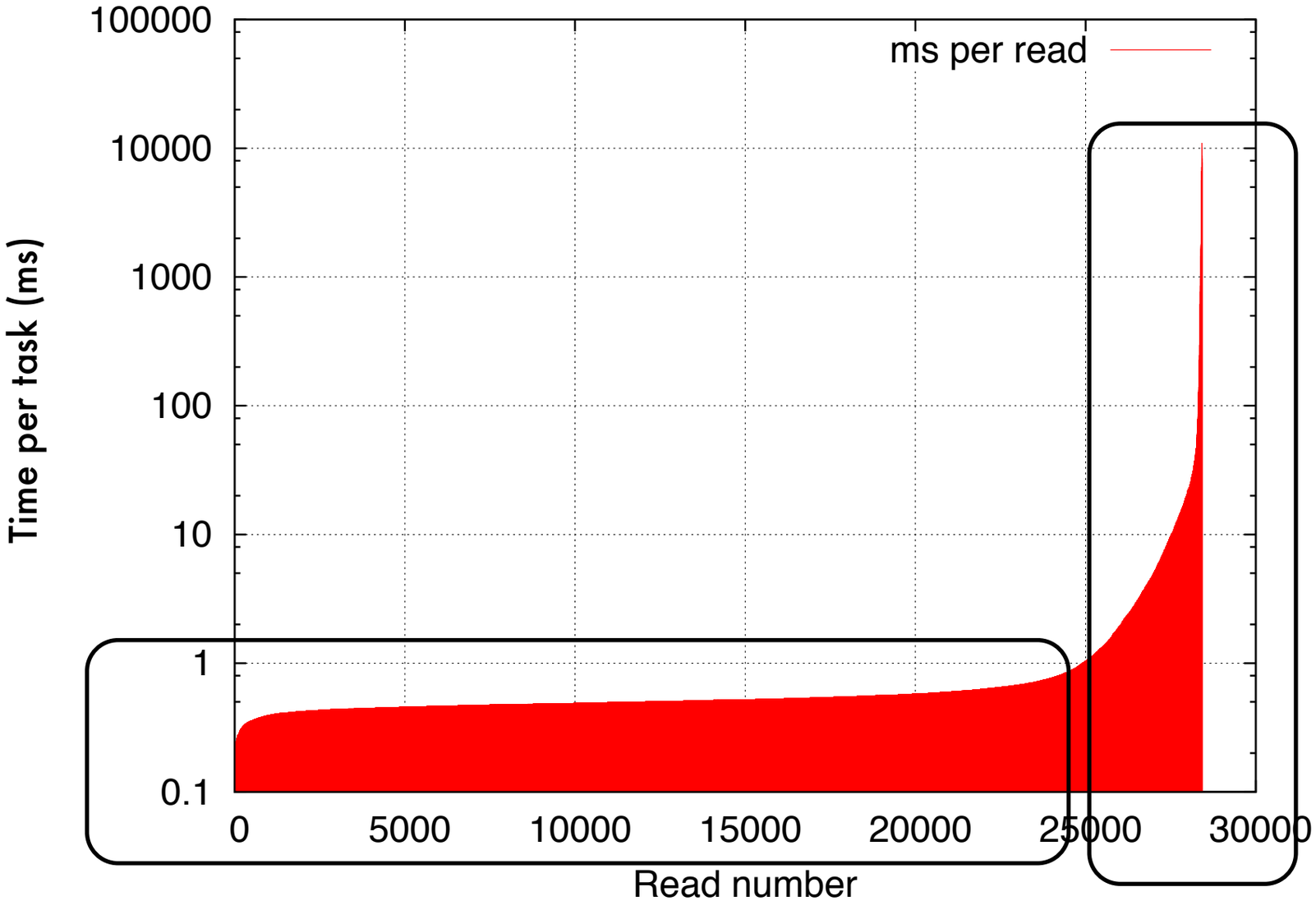
The whole dataset, TB of data, should be re-read to extract statistical estimators

Example: NGS

DNA alignment

- DNA sequencer output is composed by short unordered sequences (reads)
- Problem: mapping of read onto the right position in the DNA
- Very soon: Personalized drug design
- Several existing tools:
  - e.g. Bowtie, Shrimp, ...
  - Developed for multi-core servers, not for the cloud

DNA alignment: Load distribution (human genome mapping, real clinical data)



few medium tasks (10 s)
CPU intensive
requires load balancing

many very small tasks (<1ms)
memory intensive, disk intensive
requires fine-grain techniques

... and you don't know in advance which one you are processing

How software is build?

- Parallel computing
  - already assessed
- Typically using platform specific methodologies
  - Pthreads, MPI,
- Can this software run on the cloud?
  - Might need a complete re-design and re-development
  - Does the users trust in the correctness of the porting?

Programming for portability the FastFlow example

http://mc-fastflow.sourceforge.net/

University of Torino and Pisa, Italy

FastFlow

Applications on multicore, many core & distributed platforms of multicores
Efficient and portable - designed with high-level patterns

FastFlow

Streaming network patterns
Skeletons: pipeline, map farm, reduce, D&C, ...

Arbitrary streaming networks
Lock-free SPSC/MPMC queues + FF nodes

Arbitrary streaming networks
Collective communications + FF Dnodes

Simple streaming networks
Lock-free SPSC queues + threading model

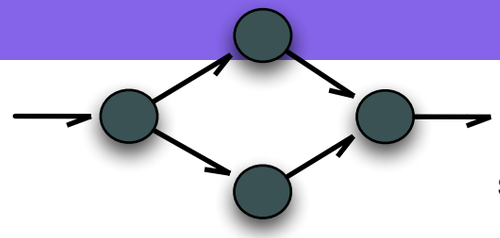
Simple streaming networks
Zero copy networking + processes model

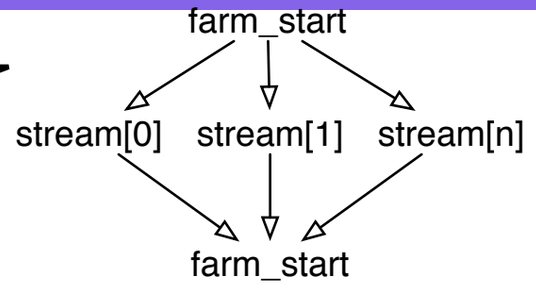
Multicore and manycore
SMP: cc-UMA & cc-NUMA

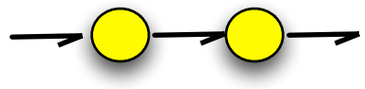
Distributed platforms
Clouds, clusters of SMPs

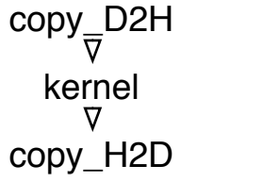
Layer 3: streaming networks patterns

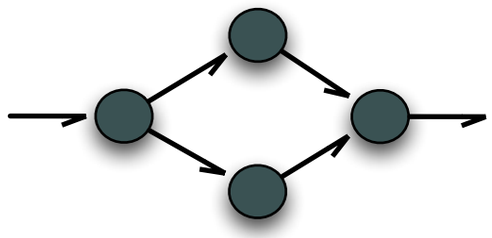
- farm
  - on CPU - master-worker - parallelism exploitation
  - on GPU - CUDA streams - automatic exploitation of asynch comm
- pipeline
  - on CPU - pipeline
  - on GPU - sequence of kernel calls or global mem synch
- map
  - on CPU - master-worker - parallelism exploitation
  - on GPU - CUDA SIMT - parallelism exploitation
- reduce
  - on CPU - master-worker - parallelism exploitation
  - on GPU - CUDA SIMT (reduction tree) - parallelism exploitation
- D&C
  - on CPU - master-worker with feedback - // exploitation
  - on GPU - working on it, maybe loop+farm

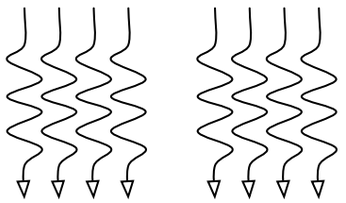


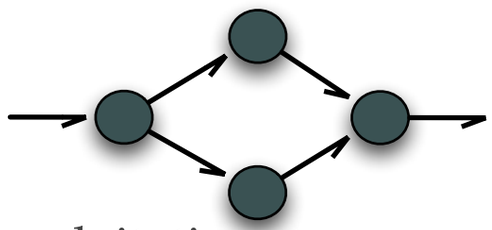


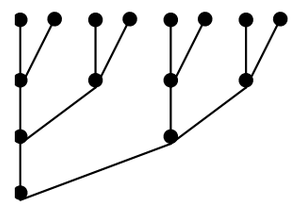


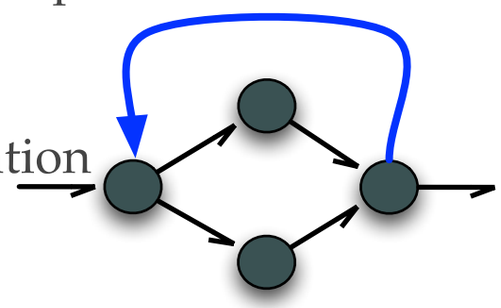










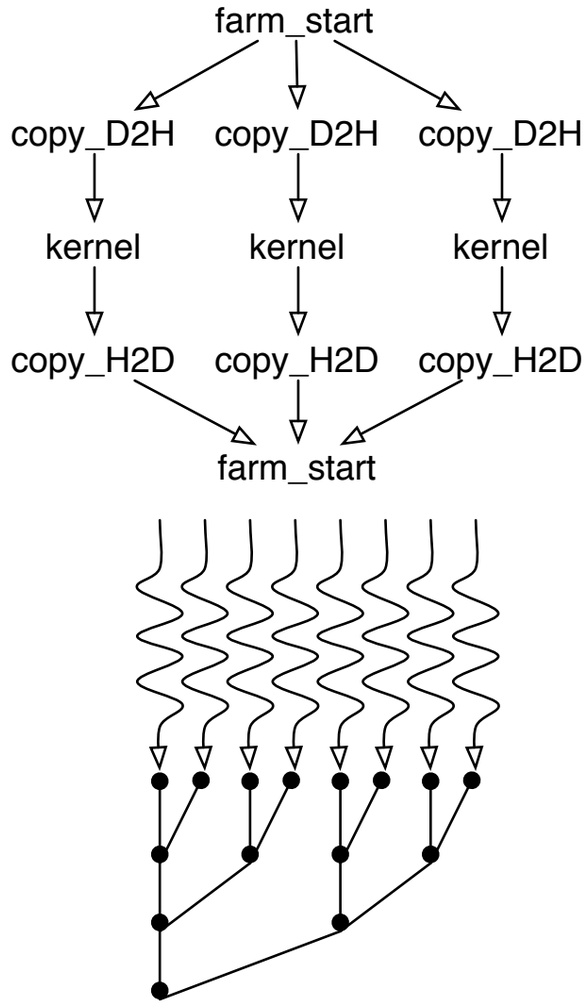


Streaming networks patterns

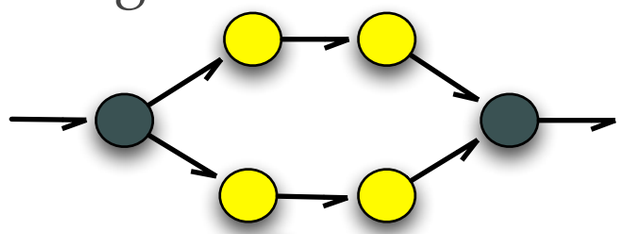
- Composition via C++ template meta-programming
  - CPU: Graph composition
  - GPU: CUDA streams
  - CPU+GPU: offloading

Multi-core & distributed

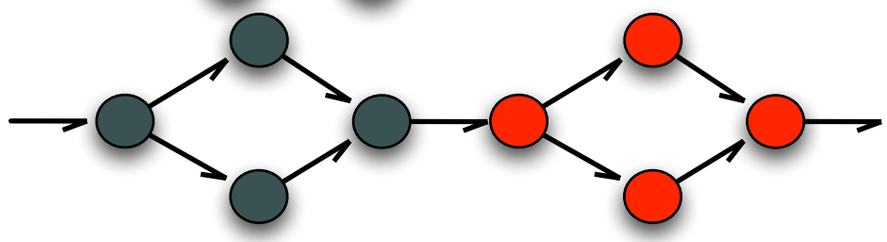
GPGPU



- farm{ pipe }

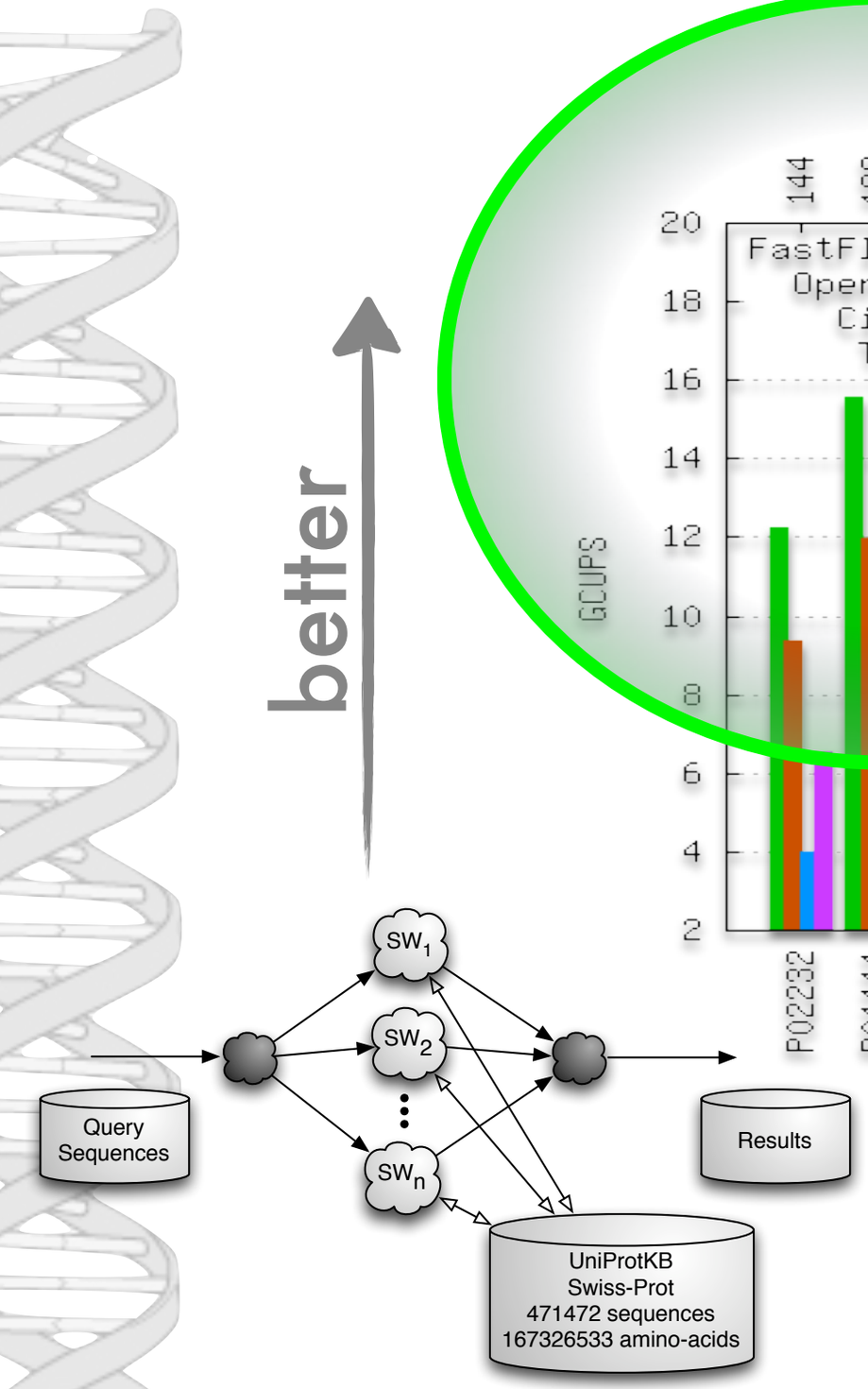


- pipe(farm, farm)



- pipe(map, reduce) i.e. Google's MapReduce
- ....

FastFlow-SWPS3 - Smith-Waterman (among the fastest SW implementation around)



A bio test case

CWC parallel simulation-analisys pipeline

http://mc-fastflow.sourceforge.net/

Monte Carlo sim for system bio

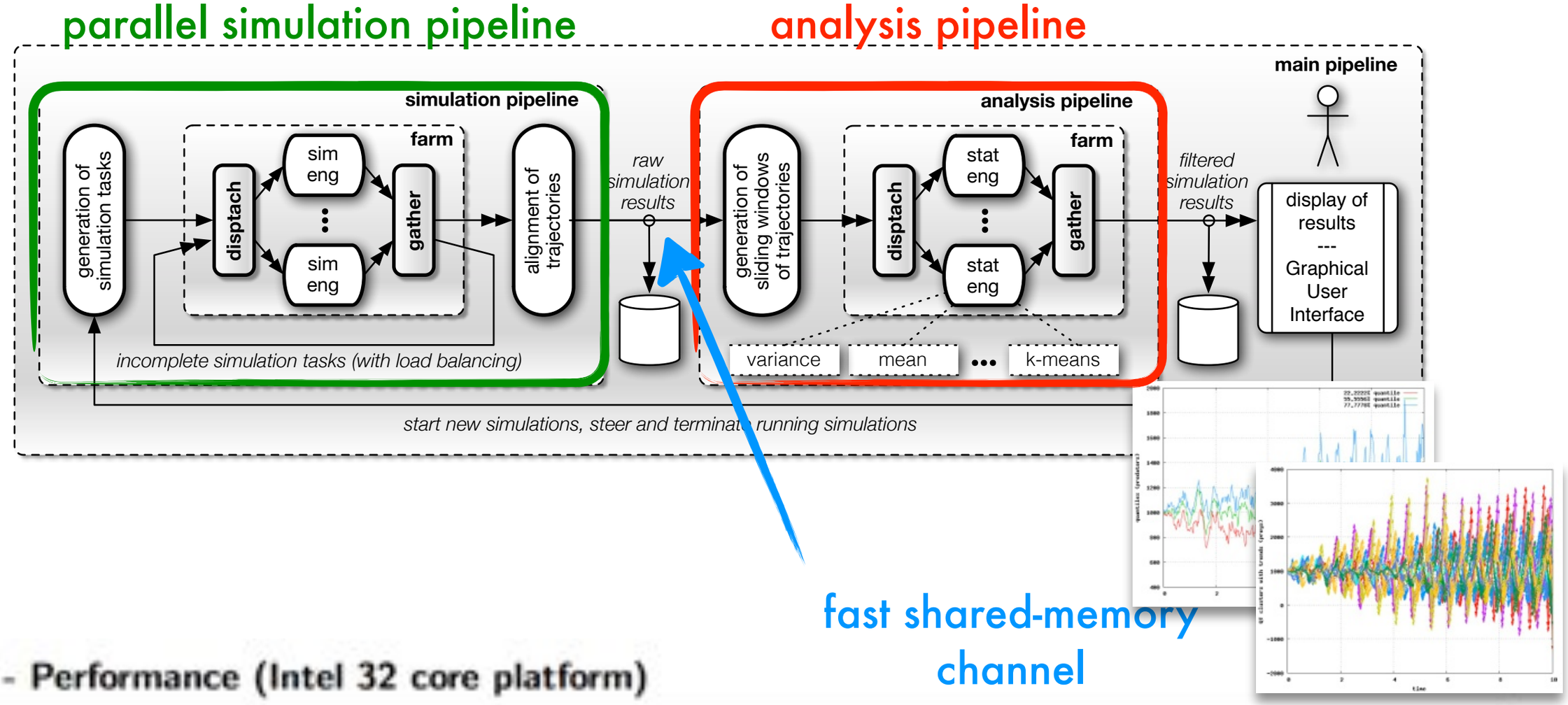
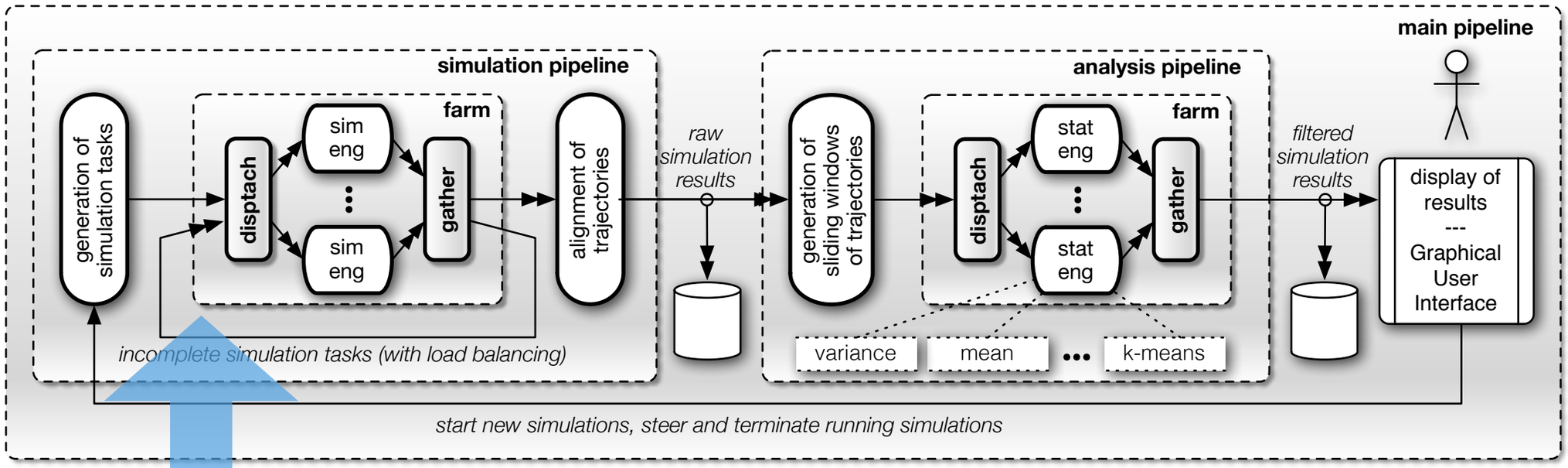
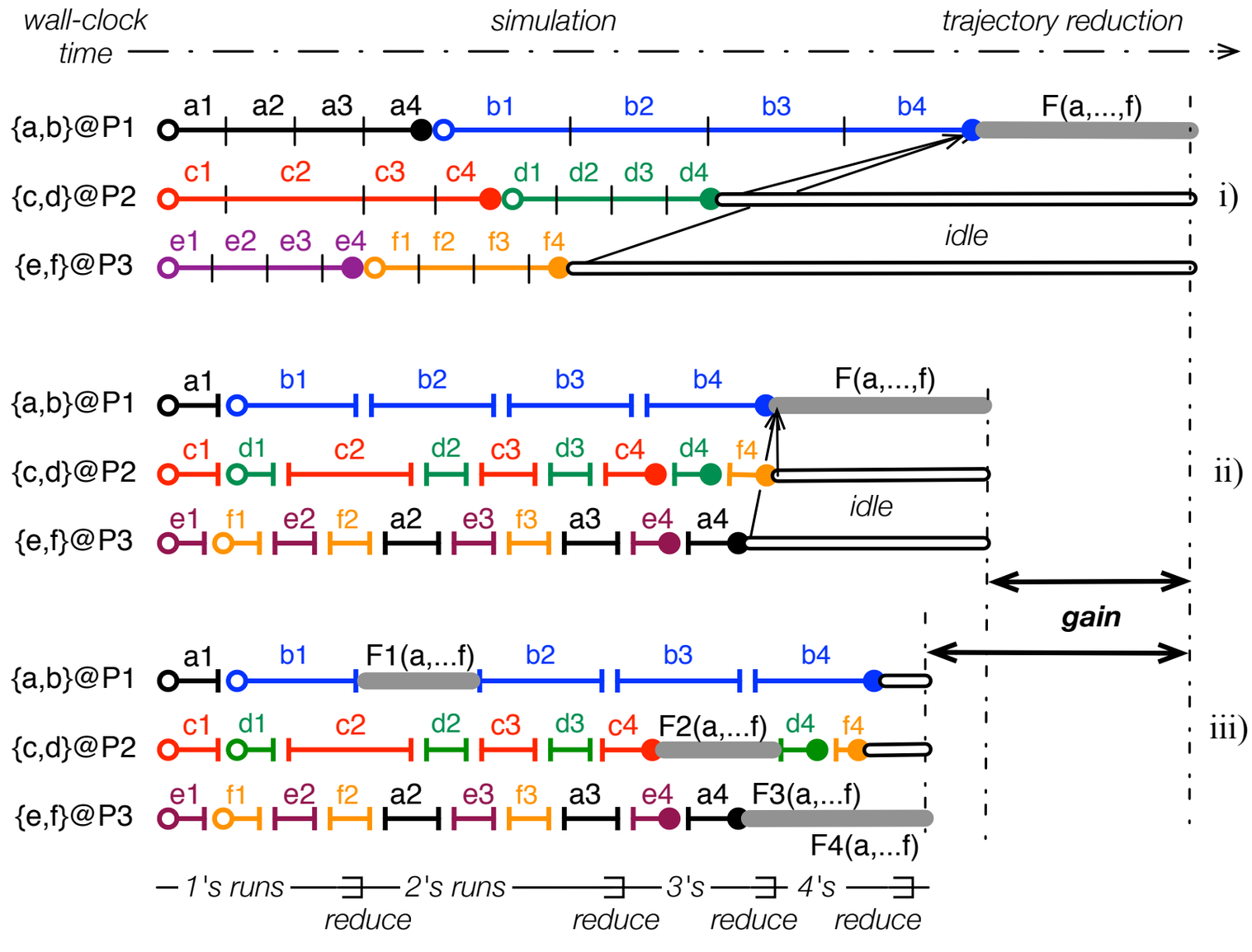


Table 2 - Performance (Intel 32 core platform)

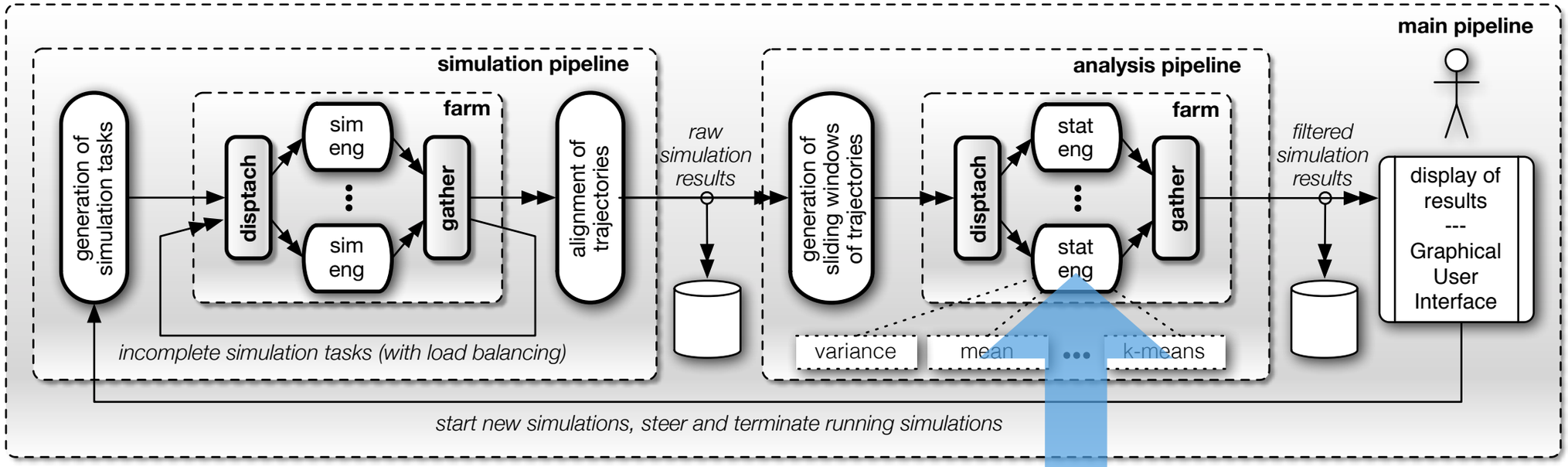
Model	Single trajectory information			Overall data (20 sim eng, 3 stat eng)		
	N. samples	Avg sim step	Sample time	Inter-arrival time	Throughput	Output size
Neurospora	10^4	7.80 μs	517.24 μs	25.86 μs	11.87 MB/s	36.62 MB
Neurospora	10^5	8.37 μs	55.51 μs	2.78 μs	11.98 MB/s	366.21 MB
Neurospora	10^6	75.63 μs	4.65 μs	232.68 ns	201.63 MB/s	3.58 GB
EColi	10^6	173.64 μs	0.58 μs	28.81 ns	257.66 MB/s	4.47 GB
Lotka-Volterra	10^6	22.86 μs	0.69 μs	34.68 ns	147.11 MB/s	2.68 GB

Aldinucci et al. IEEE PDP 2011, HiBB 2011, IEEE PDP 2013





M. Aldinucci, M. Coppo, F. Damiani, M. Drocco, M. Torquati, A. Troina. On designing multicore-aware simulators for biological systems. PDP 2011. 2011. IEEE.

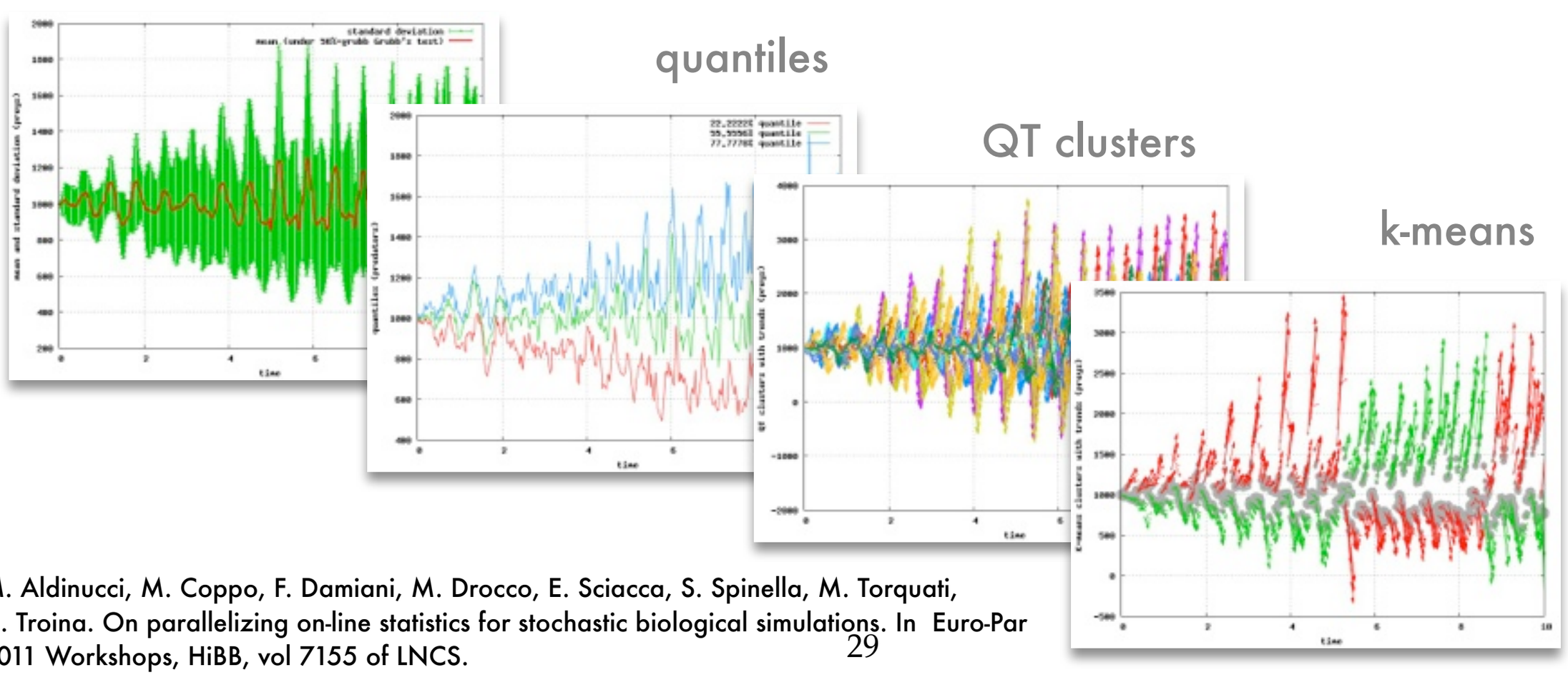


mean variance

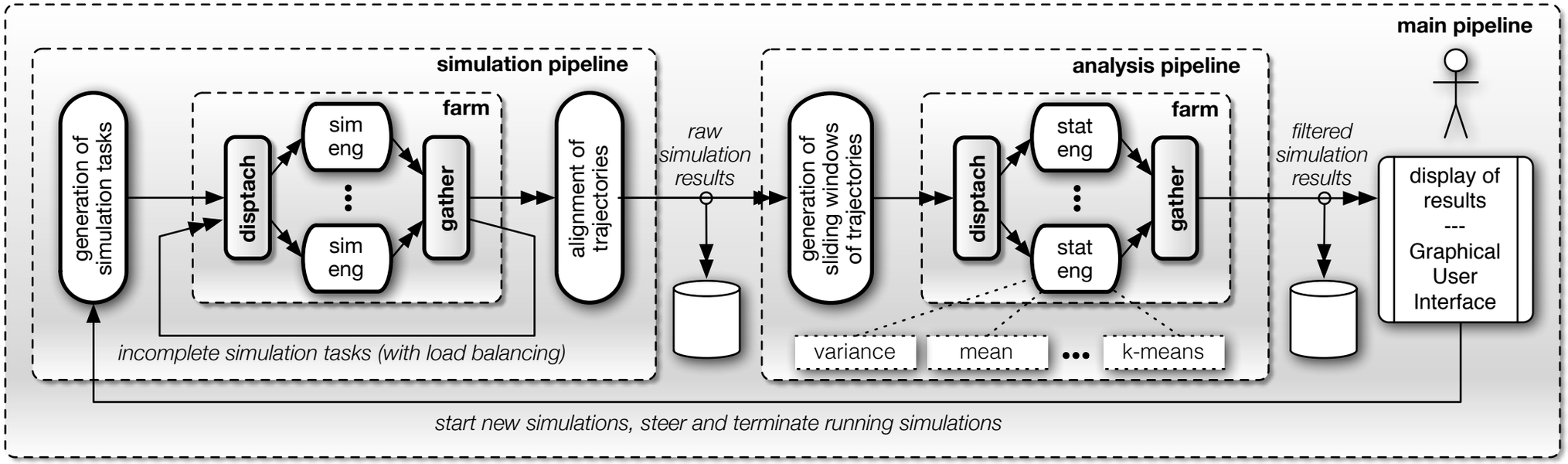
quantiles

QT clusters

k-means



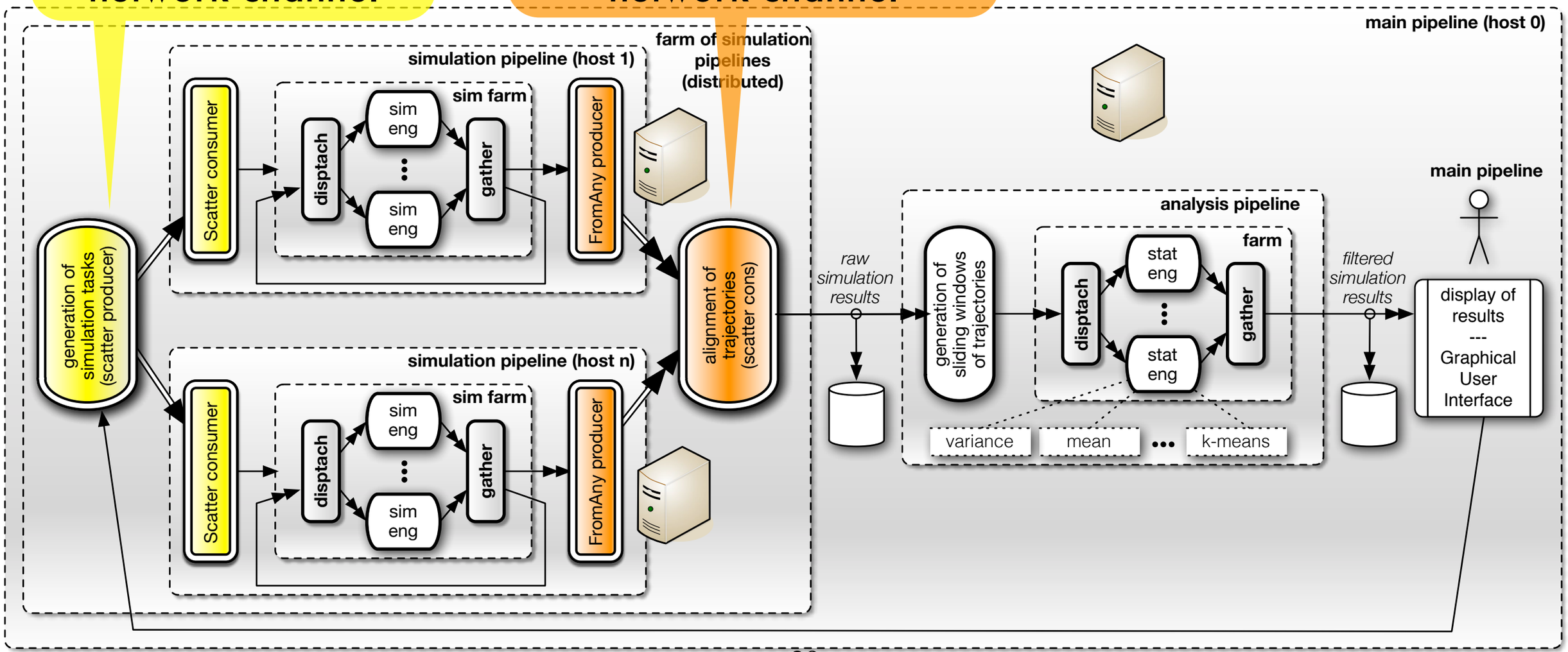
M. Aldinucci, M. Coppo, F. Damiani, M. Drocco, E. Sciacca, S. Spinella, M. Torquati, A. Troina. On parallelizing on-line statistics for stochastic biological simulations. In Euro-Par 2011 Workshops, HiBB, vol 7155 of LNCS.



stream scatter network channel

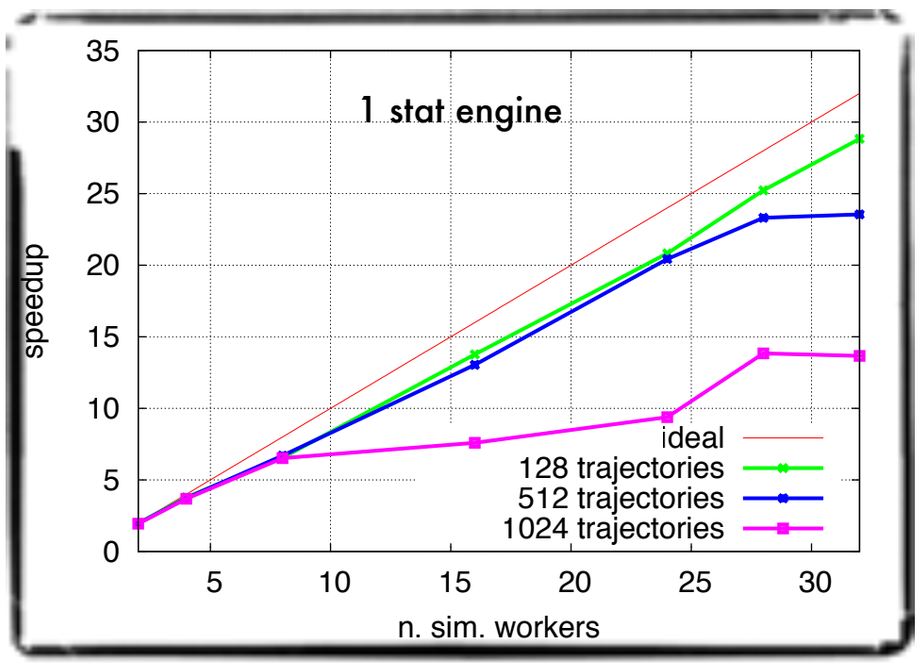
stream join network channel

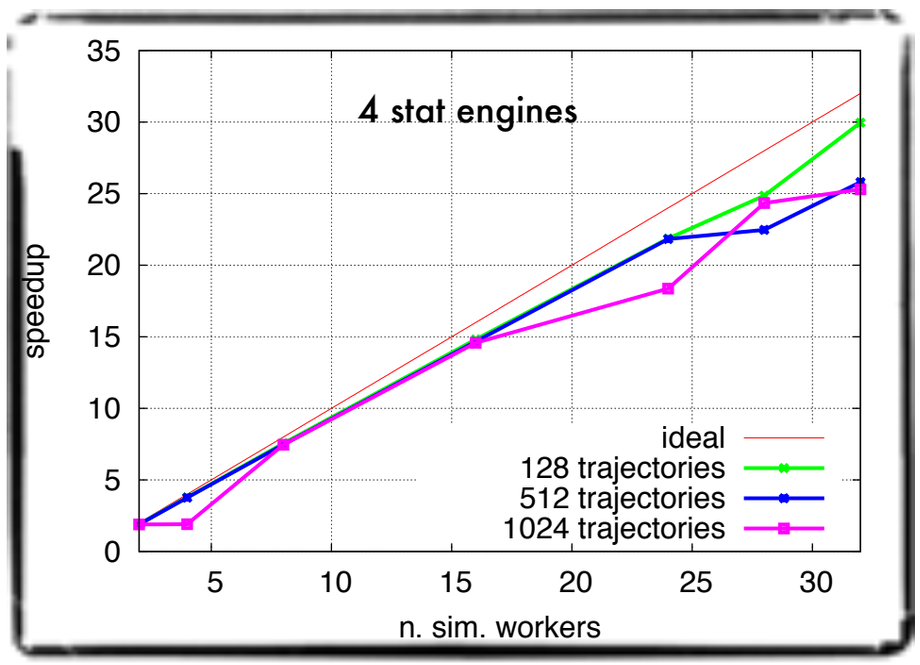
Distributed



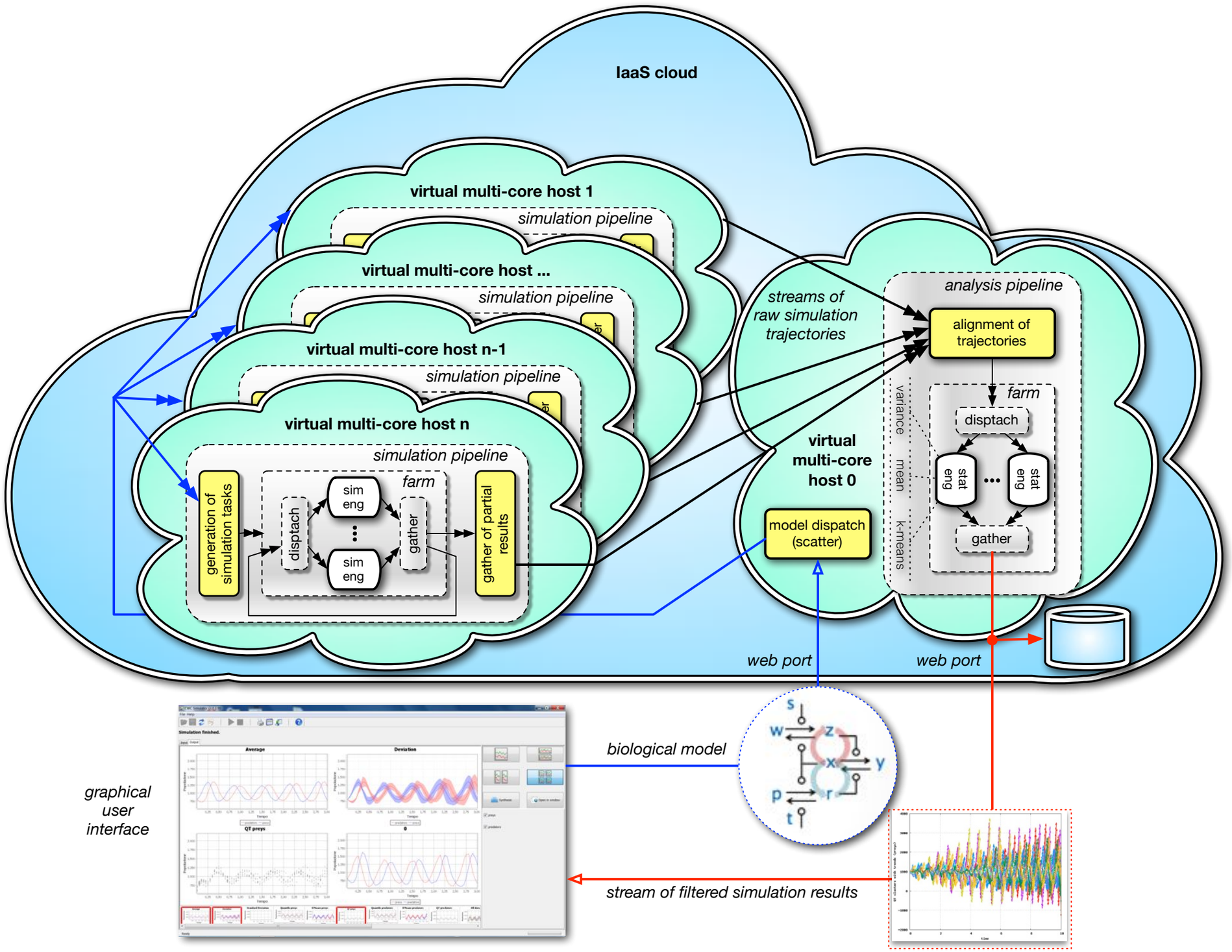
The key points

- exactly the same code: pipeline(farm,farm)
- the porting problem is moved from the programmers to development tools
- with good performance portability

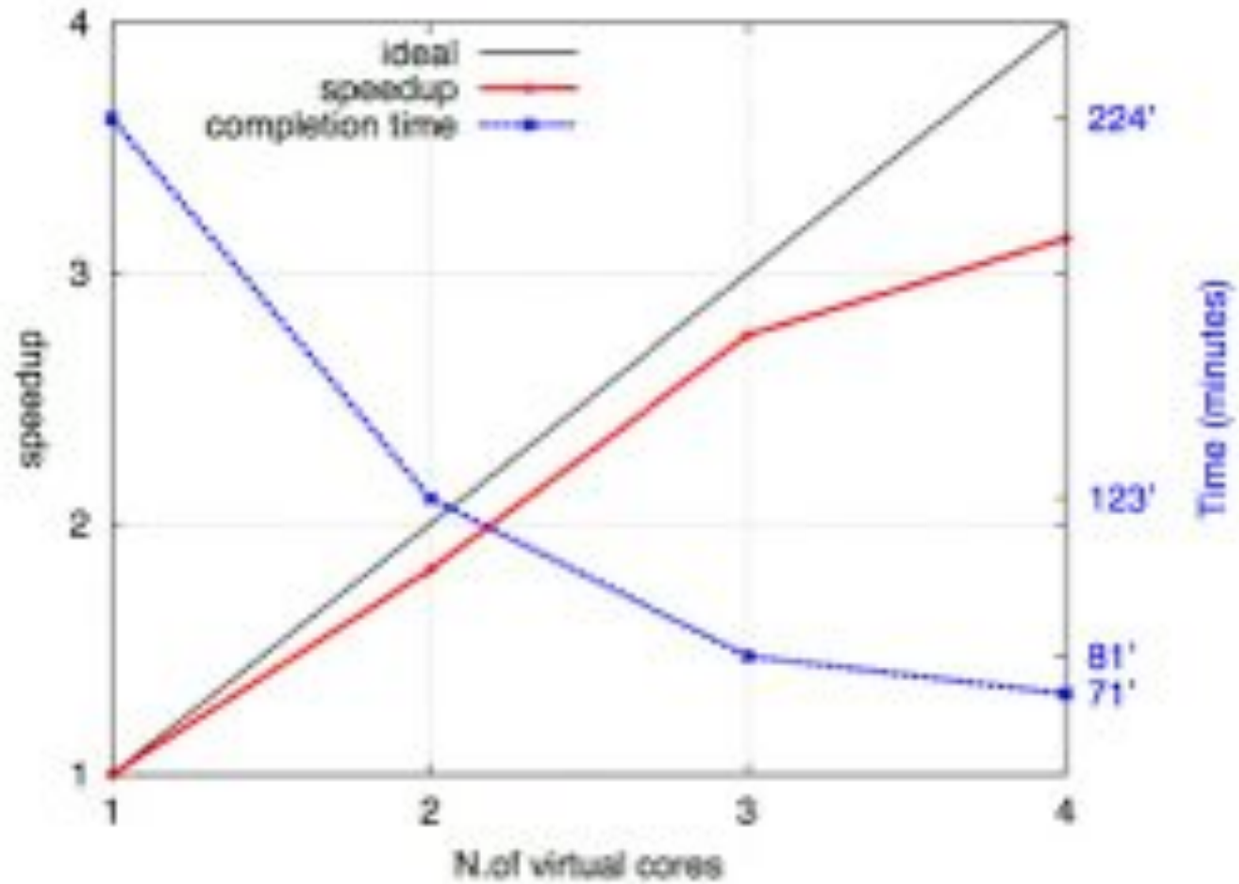


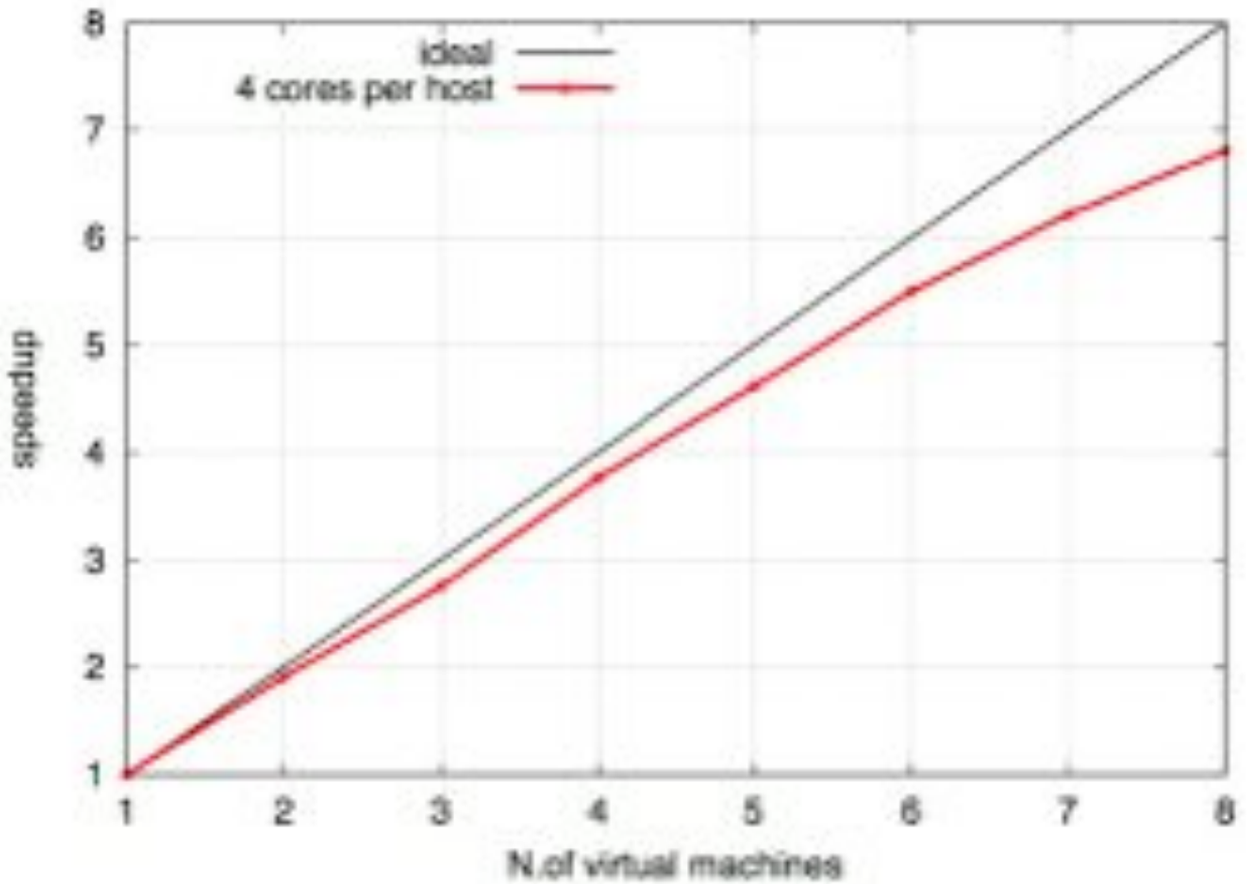


… and on the cloud running as a service (and exactly with the same code)



Amazon EC2 performances





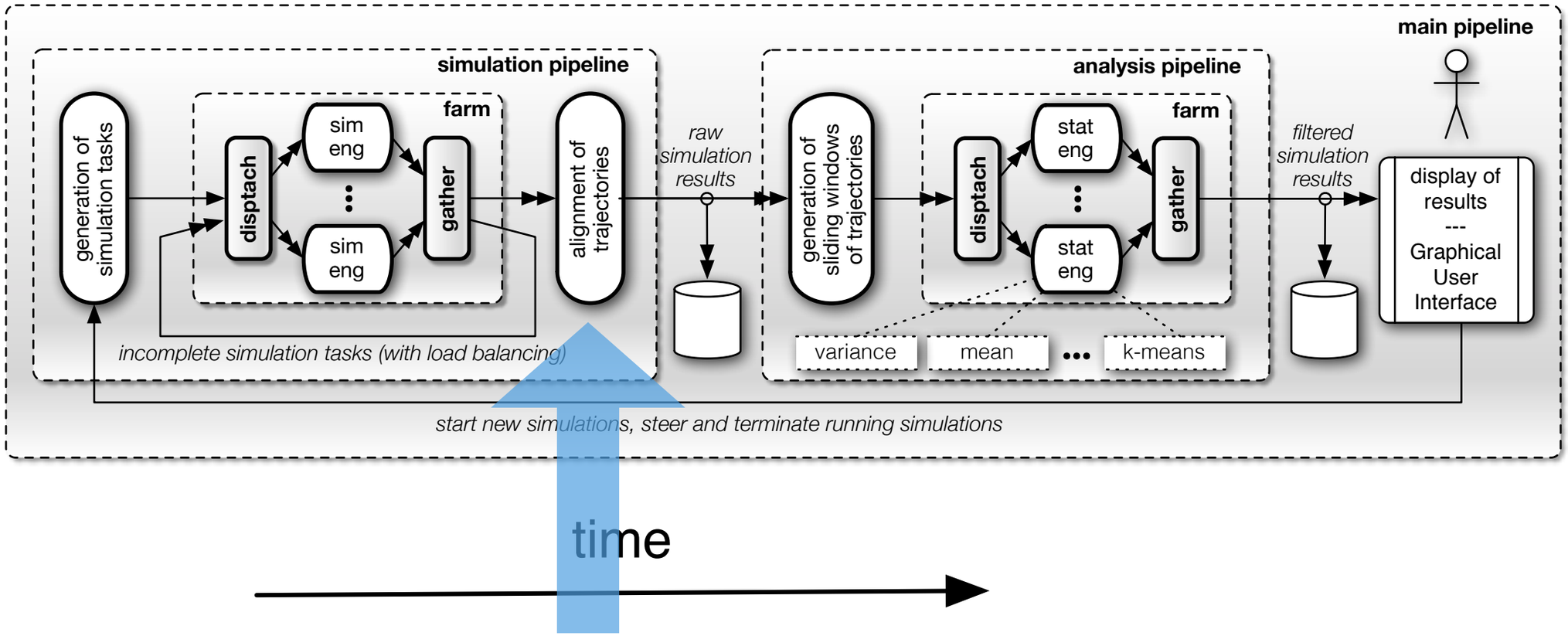
Involved data

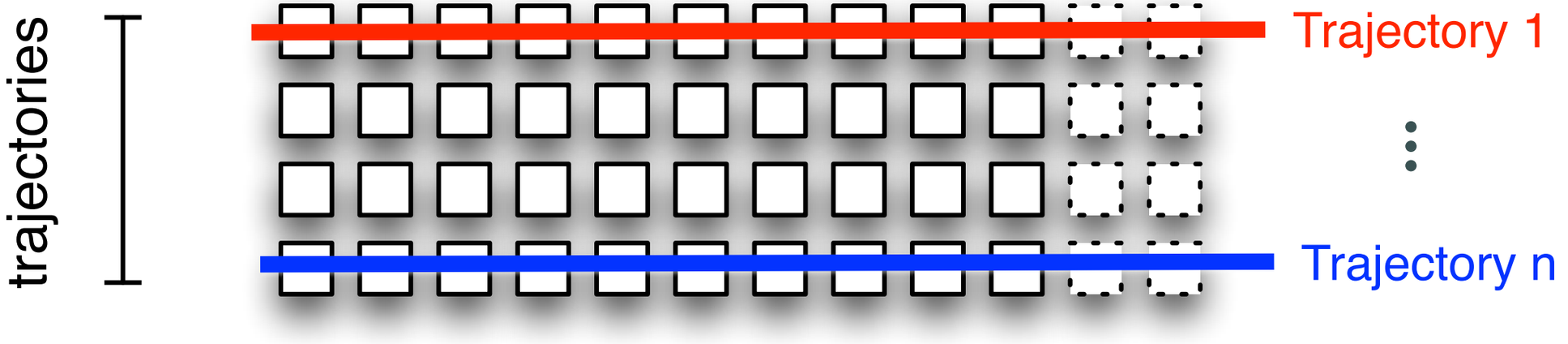
- Simple examples (neurospora, ...)
  - 2-4 double * n. of variables * n. of samples x n. of trajectories * cases in sensitivity analysis
  - e.g.4*8*4*1M*1k*8 ~ 1 TBytes
- HIV 6GB x 1024 trajectories ~ 6TB
- The more observed variables, resolution/precision, cases for sensitivity analysis, the more data

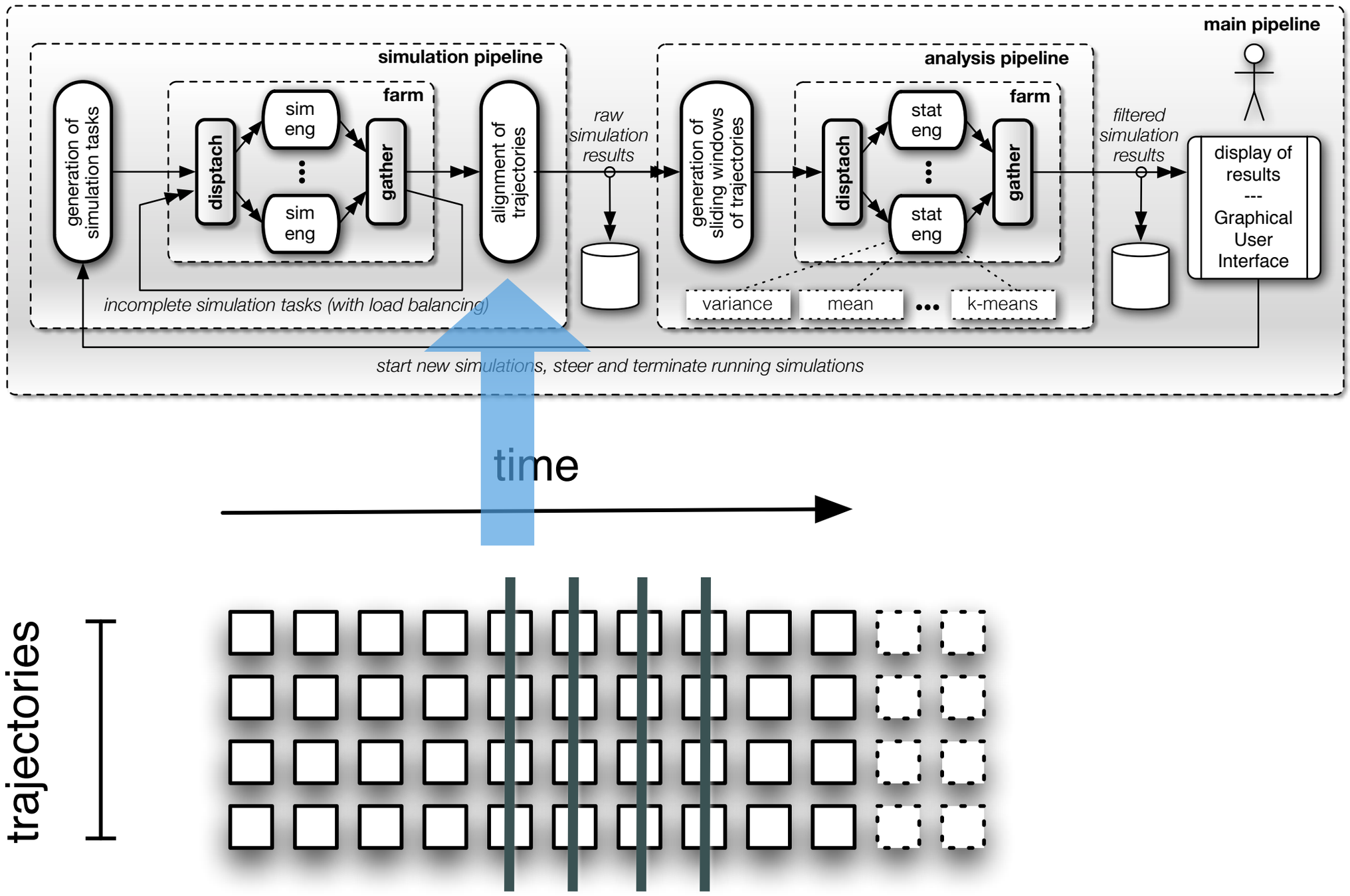
Extracting knowledge as fully on-line process

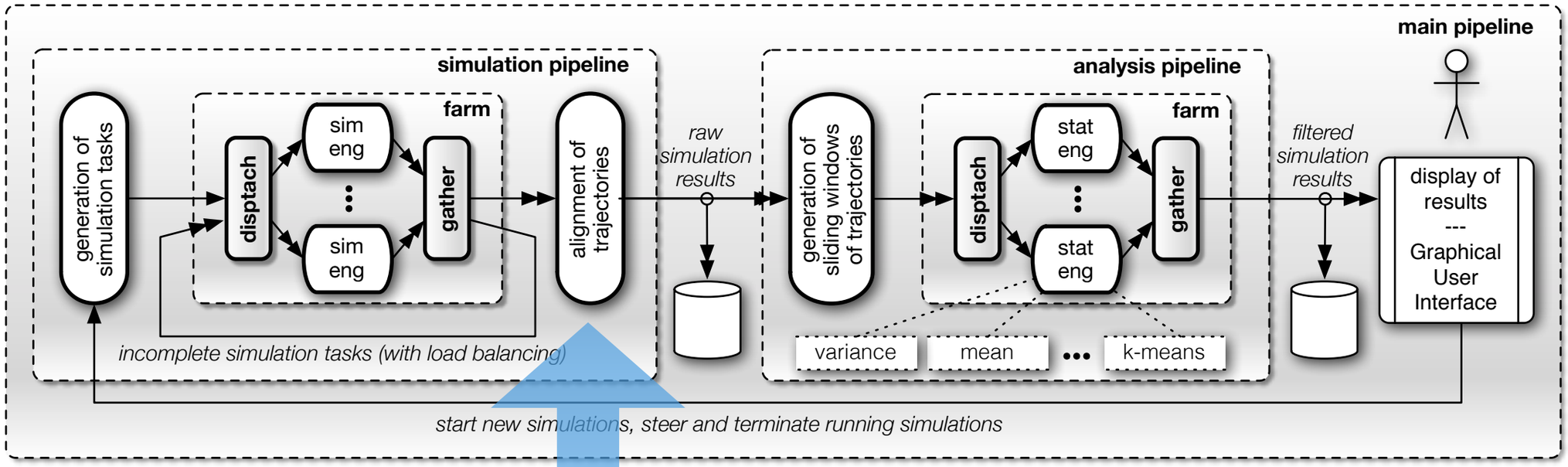
Rationale

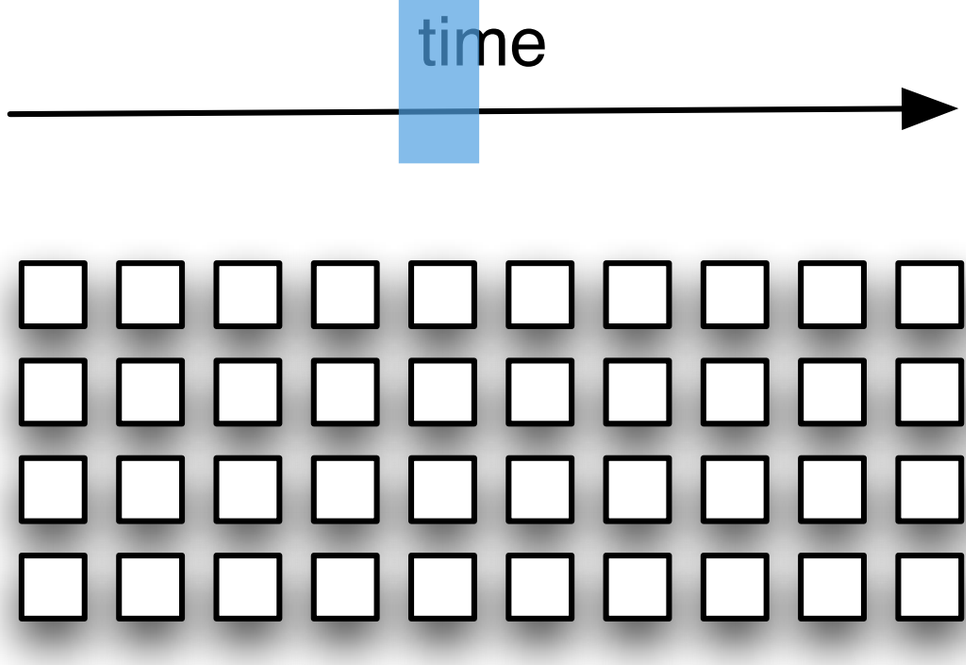
- Working on data as soon as they are produced (in pipeline)
  - immediate answer to the bio-scientist
  - when you see the first page of a Google search the searching is still on going
- Work on sliding windows of data instead of the full dataset
  - many standard mining techniques still applicable

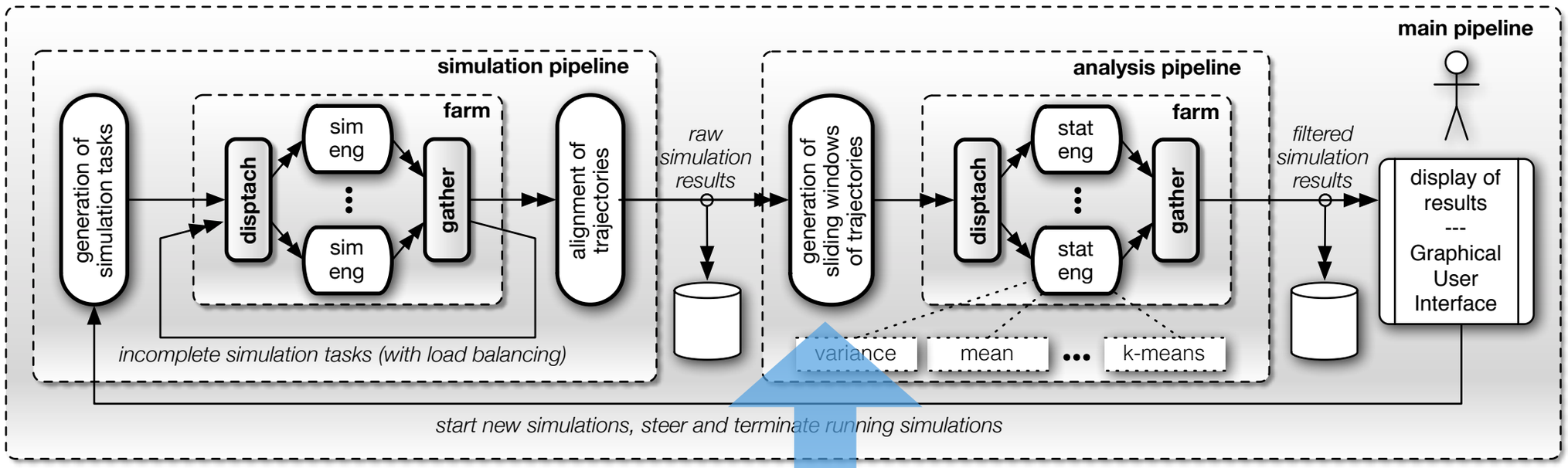

main pipeline
simulation pipeline
analysis pipeline
farm
farm
generation of simulation tasks
disptach
sim eng
sim eng
gather
alignment of trajectories
raw simulation results
generation of sliding windows of trajectories
disptach
stat eng
stat eng
gather
filtered simulation results
display of results --- Graphical User Interface
incomplete simulation tasks (with load balancing)
variance
mean
k-means
start new simulations, steer and terminate running simulations
time

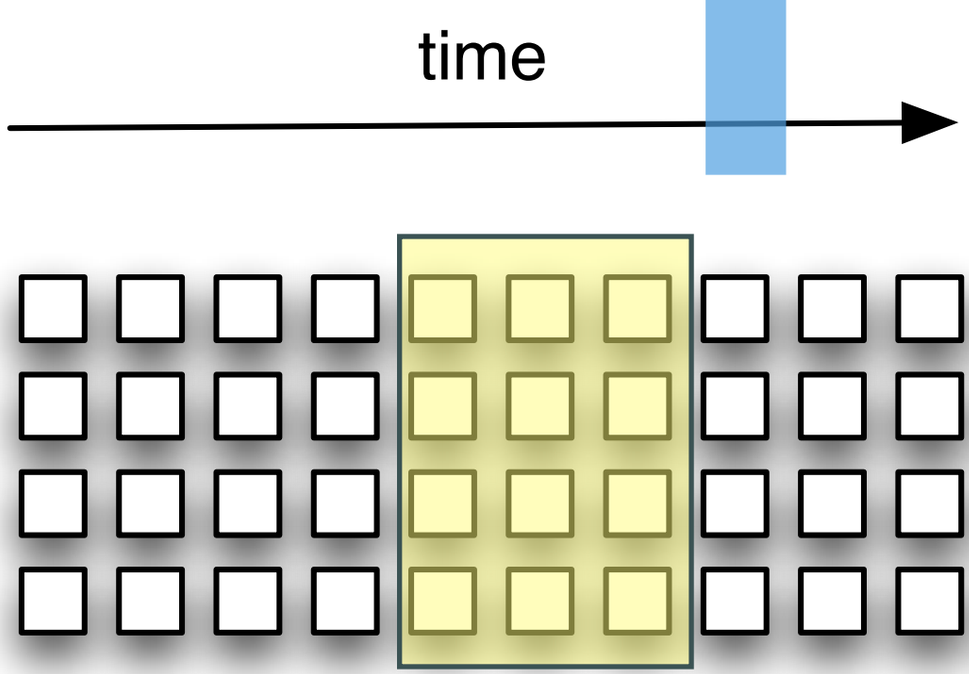

trajectories
Trajectory 1
Trajectory n

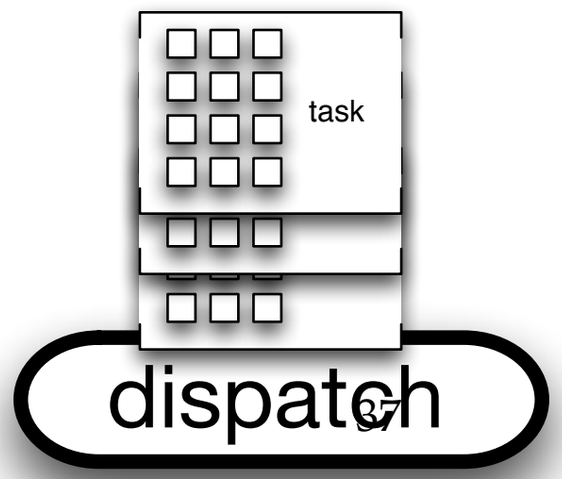

main pipeline
simulation pipeline
analysis pipeline
farm
farm
generation of simulation tasks
disptach
sim eng
sim eng
gather
alignment of trajectories
raw simulation results
generation of sliding windows of trajectories
disptach
stat eng
stat eng
gather
filtered simulation results
display of results --- Graphical User Interface
incomplete simulation tasks (with load balancing)
variance
mean
k-means
start new simulations, steer and terminate running simulations
time
trajectories
simulation-time aligned trajectories


main pipeline
simulation pipeline
analysis pipeline
farm
farm
generation of simulation tasks
disptach
sim eng
sim eng
gather
alignment of trajectories
raw simulation results
generation of sliding windows of trajectories
disptach
stat eng
stat eng
gather
filtered simulation results
display of results --- Graphical User Interface
incomplete simulation tasks (with load balancing)
variance
mean
k-means
start new simulations, steer and terminate running simulations


time
trajectories


main pipeline
simulation pipeline
analysis pipeline
farm
farm
generation of simulation tasks
disptach
sim eng
sim eng
gather
alignment of trajectories
raw simulation results
generation of sliding windows of trajectories
disptach
stat eng
stat eng
gather
filtered simulation results
display of results --- Graphical User Interface
incomplete simulation tasks (with load balancing)
variance
mean
k-means
start new simulations, steer and terminate running simulations


time
trajectories


task
dispatch

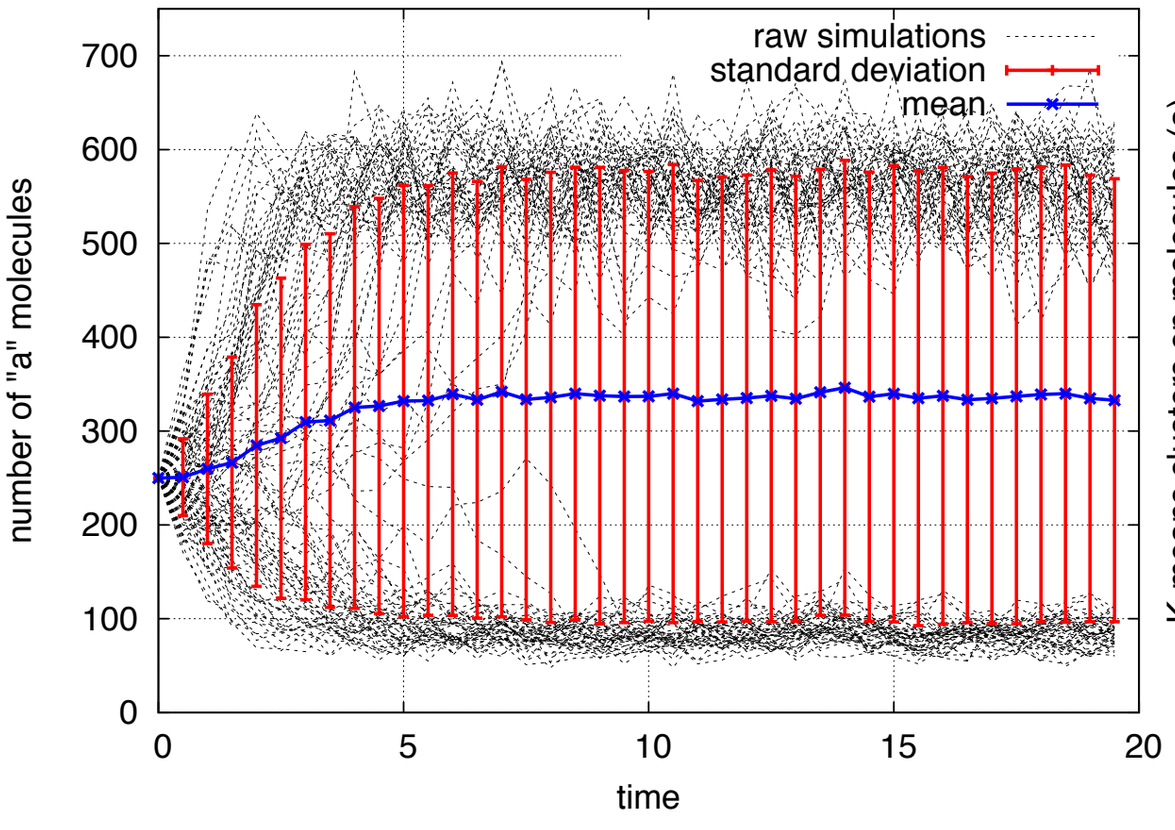
Schlögl model autocatalytic, tri-molecular reaction scheme (bistable)

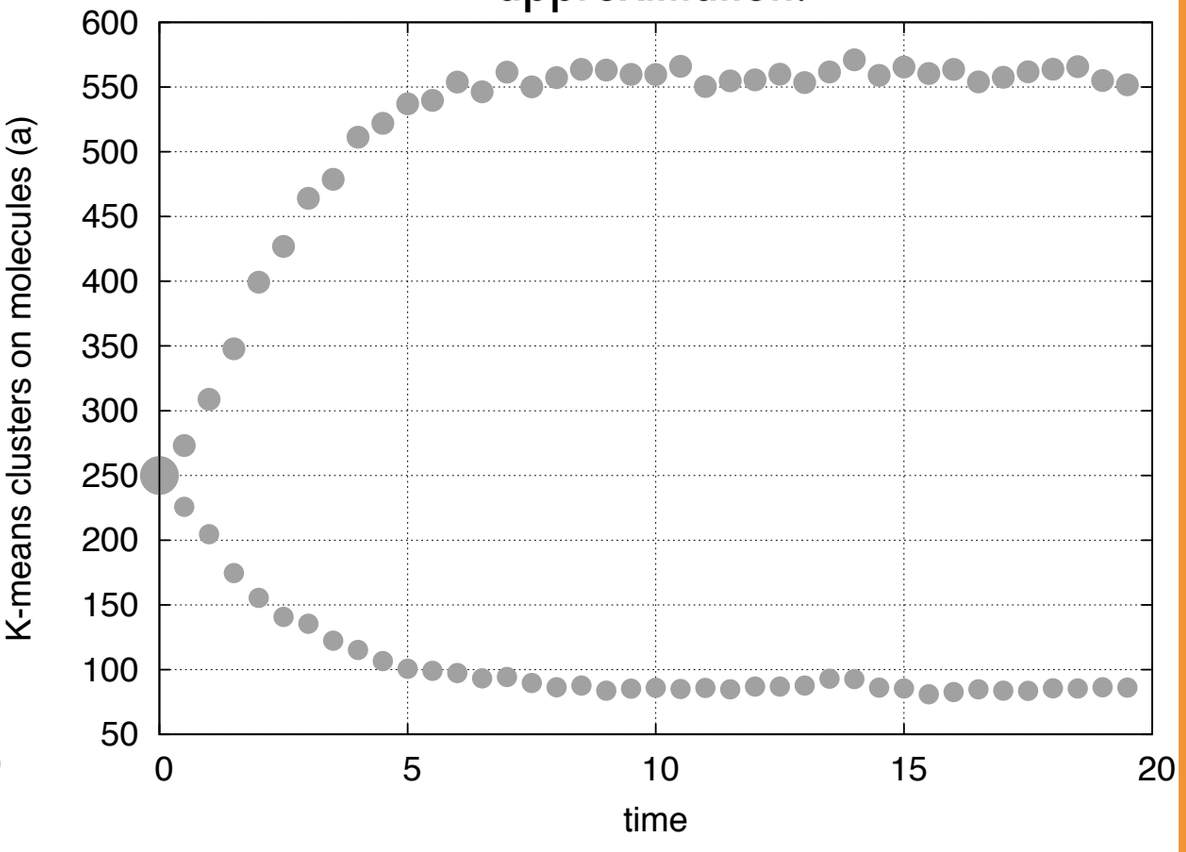
$$\mathrm{A} + 2\mathrm{X} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} 3\mathrm{X},$$

$$\mathrm{B} \underset{k_4}{\overset{k_3}{\rightleftharpoons}} \mathrm{X}.$$

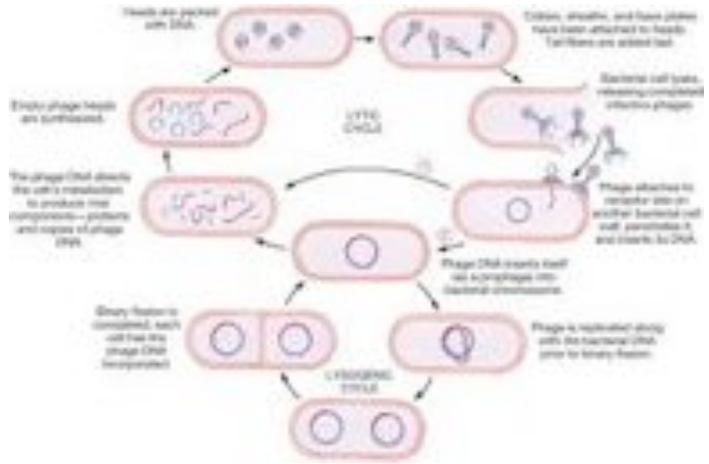
$$\top : a\ a \xmapsto{0.03} a\ a\ a \qquad \top : a\ a\ a \xmapsto{0.0001} a\ a \qquad \top : b \xmapsto{200} b\ a \qquad \top : a \xmapsto{3.5} \bullet$$

Notice this is a clustering of curves and is done while the curves are not yet fully produced. It can be done on-line with very good approximation.





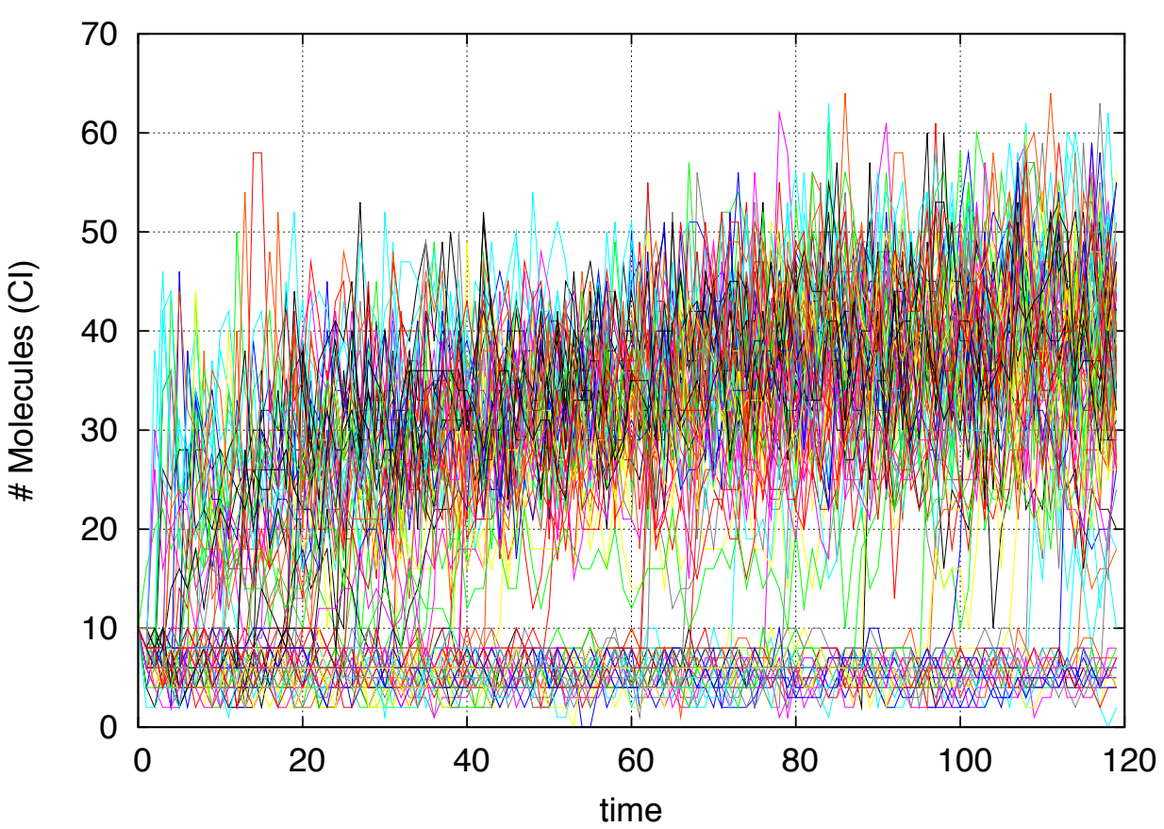
Bacteriophage λ life cycle integration of a strand of DNA in the molecule of E. coli DNA (multi-stable)

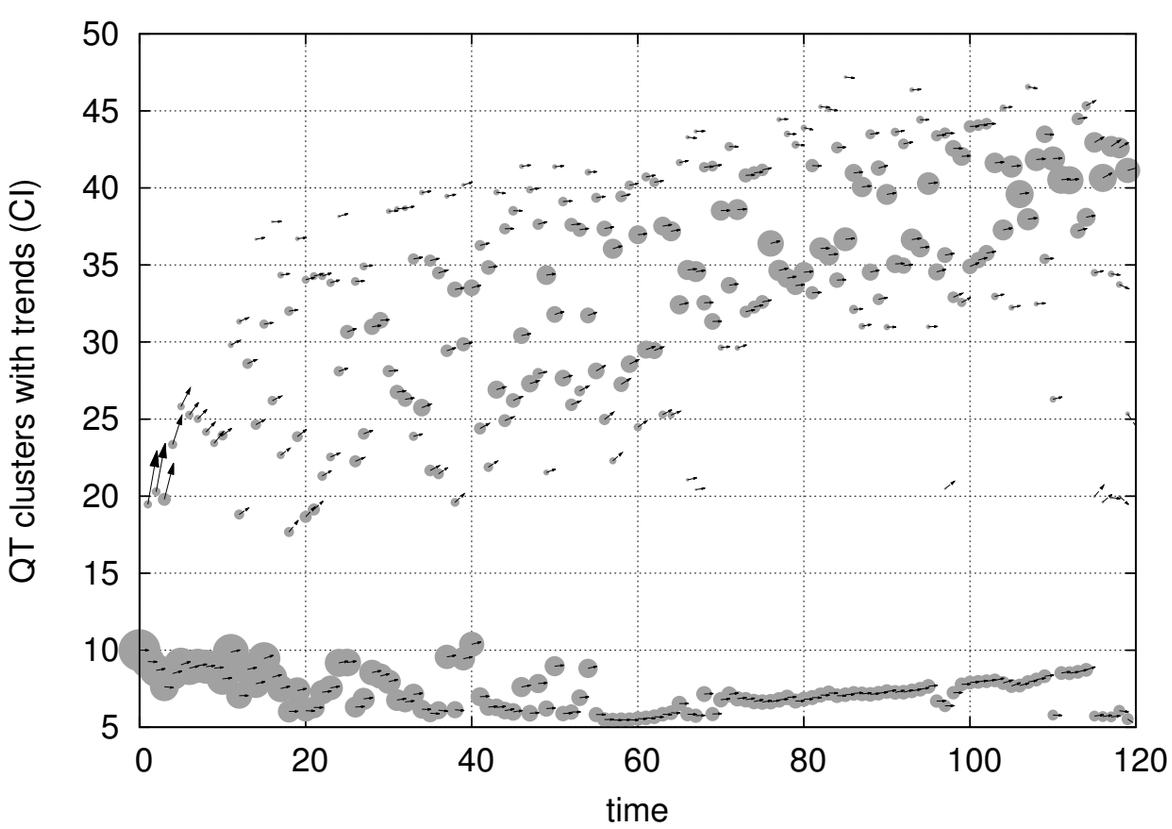


$$\top : CI\ CI \xmapsto{K_1} CI2 \qquad \top : CI2 \xmapsto{K_1} CI\ CI \qquad \top : CI2\ D \xmapsto{K_2} DCI2 \qquad \top : DCI2 \xmapsto{K_2} CI2\ D$$

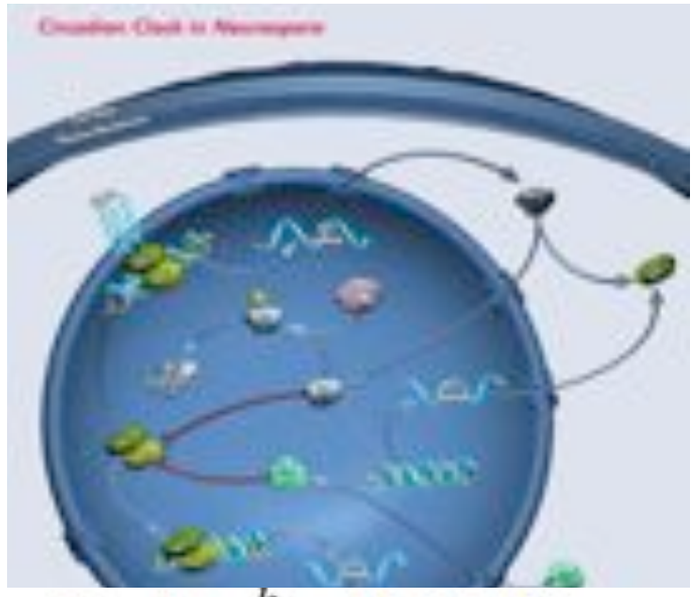
$$\top : CI2\ D \xmapsto{K_3} DCI2^* \qquad \top : DCI2^* \xmapsto{K_3} CI2\ D \qquad \top : DCI2\ CI2 \xmapsto{K_4} DCI2CI2$$

$$\top : DCI2CI2 \xmapsto{K_4} DCI2\ CI2 \qquad \top : DCI2\ P \xmapsto{k_t} DCI2\ P\ CI2\ CI2 \qquad CI \xmapsto{k_d} \bullet$$





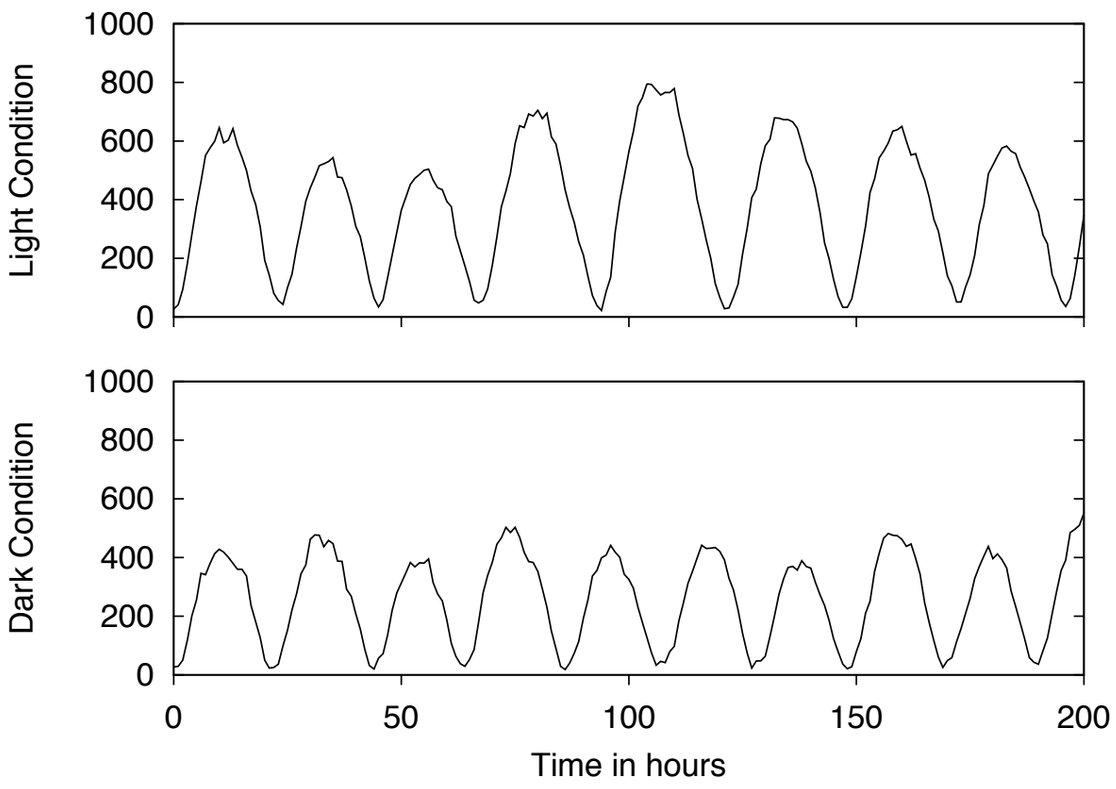
Transcriptional regulation in Neurospora (circadian clock period detection)

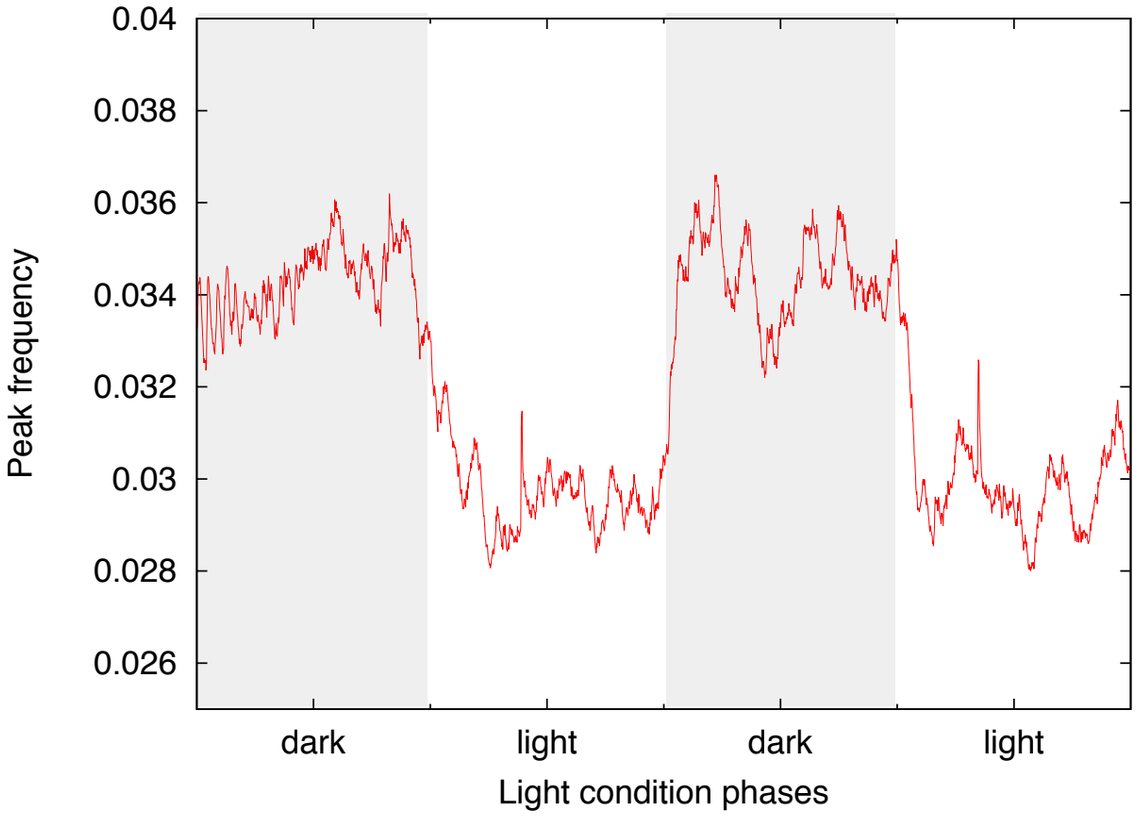


$$\top : (x \rfloor FRQ\ X)^{\eta} \xrightarrow{f_{FRQ}(t)} (x \rfloor FRQ\ X)^{\eta}\ M \qquad \top : M \xrightarrow{k_s} M\ FRQ$$

$$\top : M \xrightarrow{f_M} \bullet \qquad \top : FRQ \xrightarrow{f_d} \bullet$$

$$\top : FRQ(x \rfloor X)^{\eta} \xrightarrow{k_1} (x \rfloor FRQ\ X)^{\eta} \qquad \top : (x \rfloor FRQ\ X)^{\eta} \xrightarrow{k_2} FRQ(x \rfloor X)^{\eta}$$





Conclusions

- Cloud will have a huge impact on other sciences
- The more clouds, the more rain fall onto programmers
  - programming models can hardly dominate heterogeneity
  - performance portability is still a big issue
- Design with high-level approach
  - help the porting from multi-core to cloud
    - with performance portability
  - MapReduce is an example, not the only one
- Data analysis is more difficult than simulation
  - not embarrassingly parallel
  - MapReduce will not be enough, we need a programming model

Acknowledgements

- Projects
  - Paraphrase: Parallel Patterns for Adaptive Heterogeneous Multicore Systems (EU-FP7 STREP, 2011-2014, 3.5 M€)
  - IMPACT: Innovative Methods for Particle Colliders at the Terascale (2012-2015, 400 K €)
  - HiPEAC: High Performance and Embedded Architecture and Compilation (EU Network of Excellence, 2012-2016)
  - BETTY: Behavioral Types for Reliable Large-Scale Software Systems (EU Cost Action, 2012-2016)
  - BioBITS: Converging technologies: Biotechnology-ICT (2008-2012)
- Contributions from my group
  - Guilherme Peretti Pezzi, Irfan Uddin, Fabio Tordini, Claudia Misale, Maurizio Drocco
- Contributions for HPC from
  - Massimo Torquati (Pisa), Massimiliano Meneghin (IBM Research), Marco Danelutto (Pisa), Peter Kilpatrick (Belfast),
- Contributions for biological models from
  - Luca Cardelli (Microsoft), Pietro Liò (Cambridge), Andrea Bracciali (Stirling), Eva Sciacca, Salvatore Spinella, Mario Coppo, Angelo Troina, Ferruccio Damiani, Cristina Calcagno (Torino)