GECCO 2009

Montréal July 8-12, 2009

www.sigevo.org/gecco-2009/

Data-Intensive Computing for Competent Genetic Algorithms: A Pilot Study using Meandre

Xavier Llorà

National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
Urbana, Illinois, 61801

xllora@ncsa.illinois.edu
http://www.ncsa.illinois.edu/~xllora

Outline

- Data-intensive computing and HPC?
- Is this related at all to evolutionary computation?
- Data-intensive computing with Meandre
- GAs and competent GAs
- Data-intensive computing for GAs

2 Minute HPC History

- The eighties and early nineties picture
  - Commodity hardware rare, slow, and costly
  - Supercomputers were extremely expensive
  - Most of them hand crafted and only few units
  - Two competing families
    - CISC (e.g. Cray C90 with up to 16 processors)
    - RISC (e.g. Connection Machine CM-5 with up 4,096 processors)
- Late nineties commodity hardware hit main stream
  - Start becoming popular, cheaper, and faster
  - Economy of scale
  - Massive parallel computers build from commodity components become a viable option

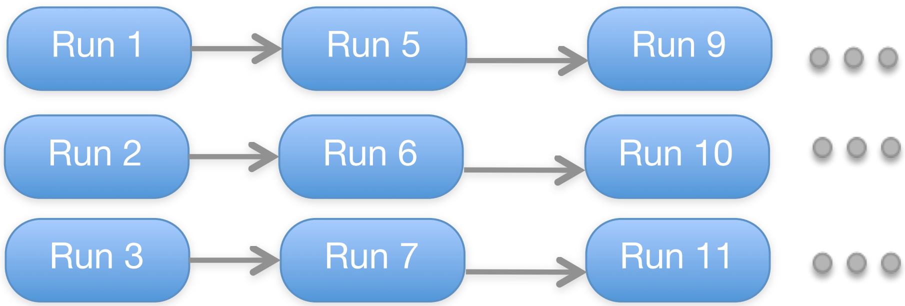
Two Visions

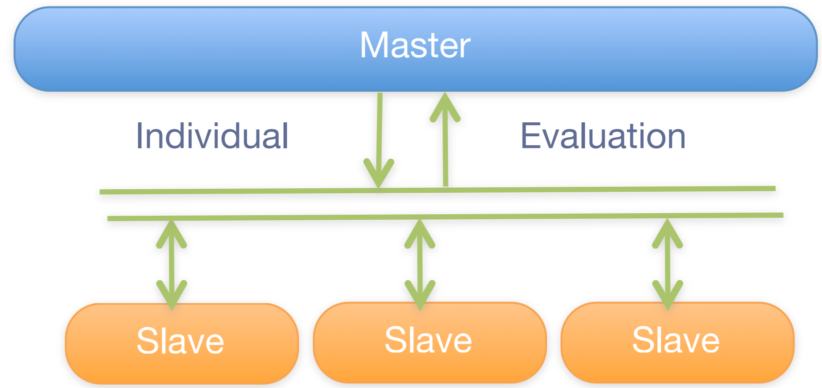
- C90 like supercomputers were like a comfy pair of trainers
  - Oriented to scientific computing
  - Complex vector oriented supercomputers
  - Shared memory (lots of them)
  - Multiprocessor enabled via some intercommunication networks
  - Single system image
- CM5 like computers did not get massive traction, but a bit
  - General purpose (as long as you can chop the work in simple units)
  - Lots of simple processors available
  - Distributed memory pushed new programming models (message passing)
  - Complex interconnection networks
- NCSA have shared memory, distributed memory, and gpgpu based

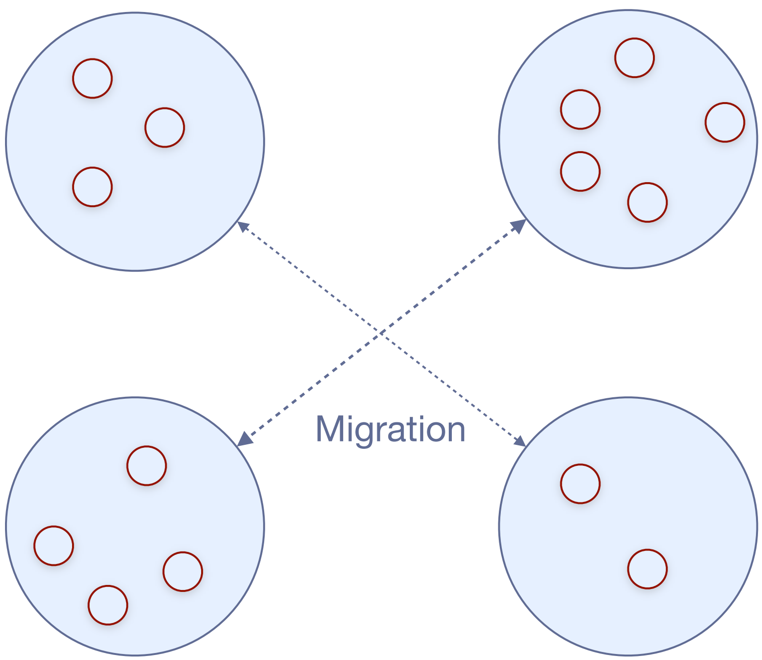
Miniaturization Building Bridges

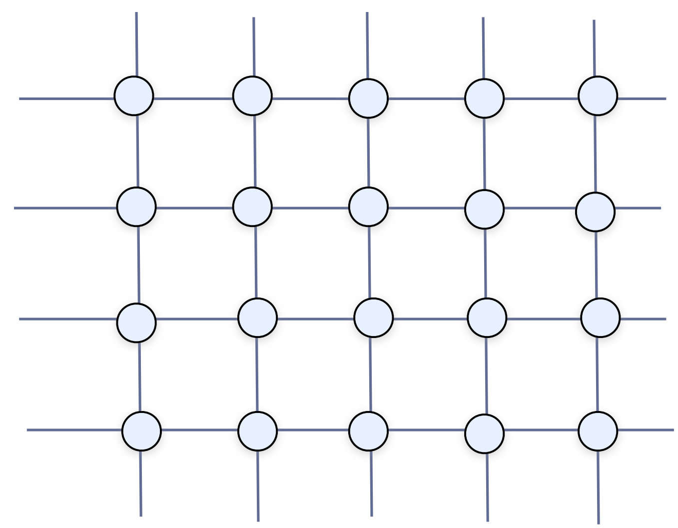
- Multicores and gpgpus are reviving the C90 flavor
- The CM-5 flavor now survives as distributed clusters of not so simple units

Control Models of Parallelization in EC







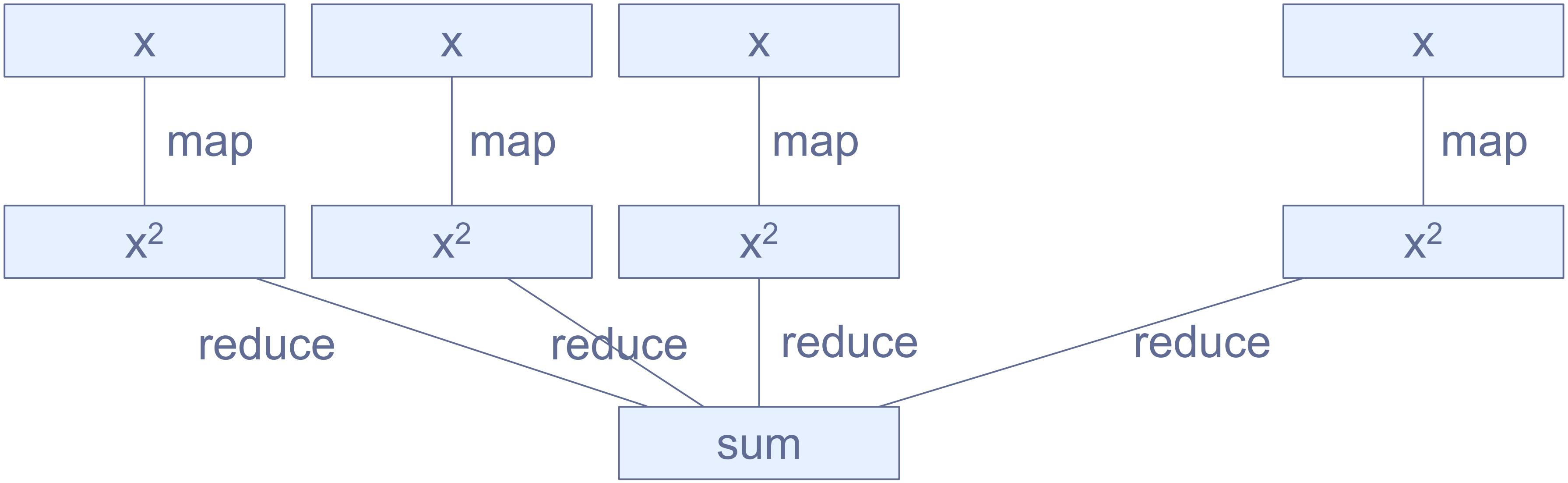


But Data is also Part of the Equation

- Google and Yahoo! revived an old route
- Usually refers to:
  - Infrastructure
  - Programming techniques/paradigms
- Google made it main stream after their MapReduce model
- Yahoo! provides and open source implementation
  - Hadoop (MapReduce)
  - HDFS (Hadoop distributed filesystem)
- Store petabytes reliably on commodity hardware (fault tolerant)
- Programming model
  - Map: Equivalent to the map operation on functional programming
  - Reduce: The reduction phase after maps are computed

A Simple Example

$$\sum_{i=0}^{n} x^2 \rightarrow reduce(map(x, sqr), sum)$$



Is This Related to EC?

- How can we easily benefit of the current core race painlessly?
- NCSA's Blue Waters estimated may top on 100K
- Yes on several facets
  - Large optimization problems need to deal with large population sizes (Sastry, Goldberg & Llorà, 2007)
  - Large-scale data mining using genetic-based machine learning (Llorà et al. 2007)
  - Competent GAs model building extremely costly and data rich (Pelikan et al. 2001)
- The goal?
  - Rethink parallelization as data flow processes
  - Show that traditional models can be map to data-intensive computing models
  - Foster you curiosity

Data-Intensive Computing with Meandre

The Meandre Infrastructure Challenges

- NCSA infrastructure effort on data-intensive computing
- Transparency
  - From a single laptop to a HPC cluster
  - Not bound to a particular computation fabric
  - Allow heterogeneous development
- Intuitive programming paradigm
  - Modular Components assembled into Flows
  - Foster Collaboration and Sharing
- Open Source
- Service Orientated Architecture (SOA)

Basic Infrastructure Philosophy

- Dataflow execution paradigm
- Semantic-web driven
- Web oriented
- Facilitate distributed computing
- Support publishing services
- Promote reuse, sharing, and collaboration
- More information at http://seasr.org/meandre

Data Flow Execution in Meandre

- A simple example $c \leftarrow a+b$
- A traditional control-driven language

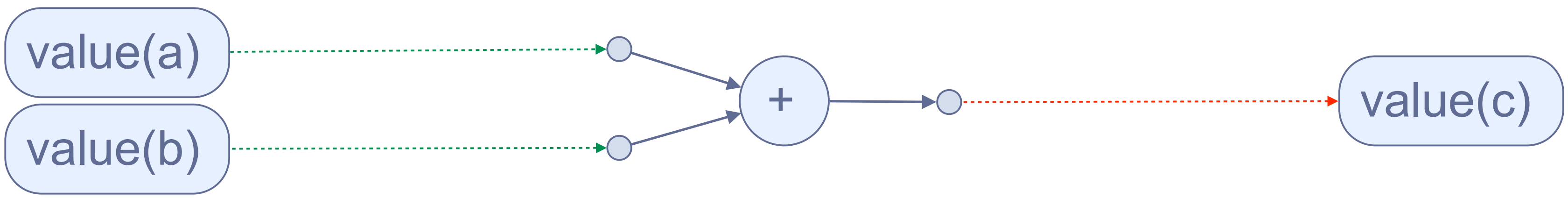
```
a = 1
b = 2
c = a+b
```

- Execution following the sequence of instructions
- One step at a time
  - a+b+c+d requires 3 steps
  - Could be easily parallelized

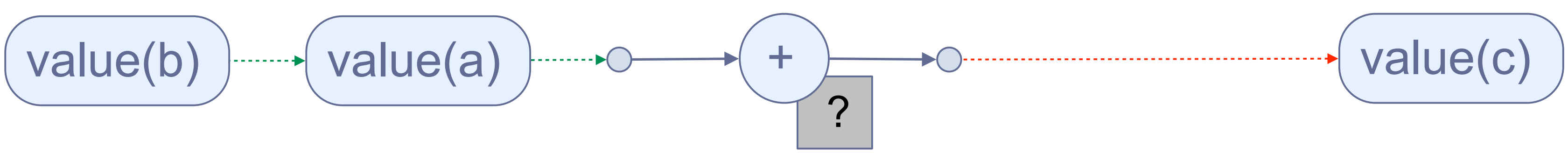
Data Flow Execution in Meandre

- Data flow execution is driven by data
- The previous example may have 2 possible data flow versions

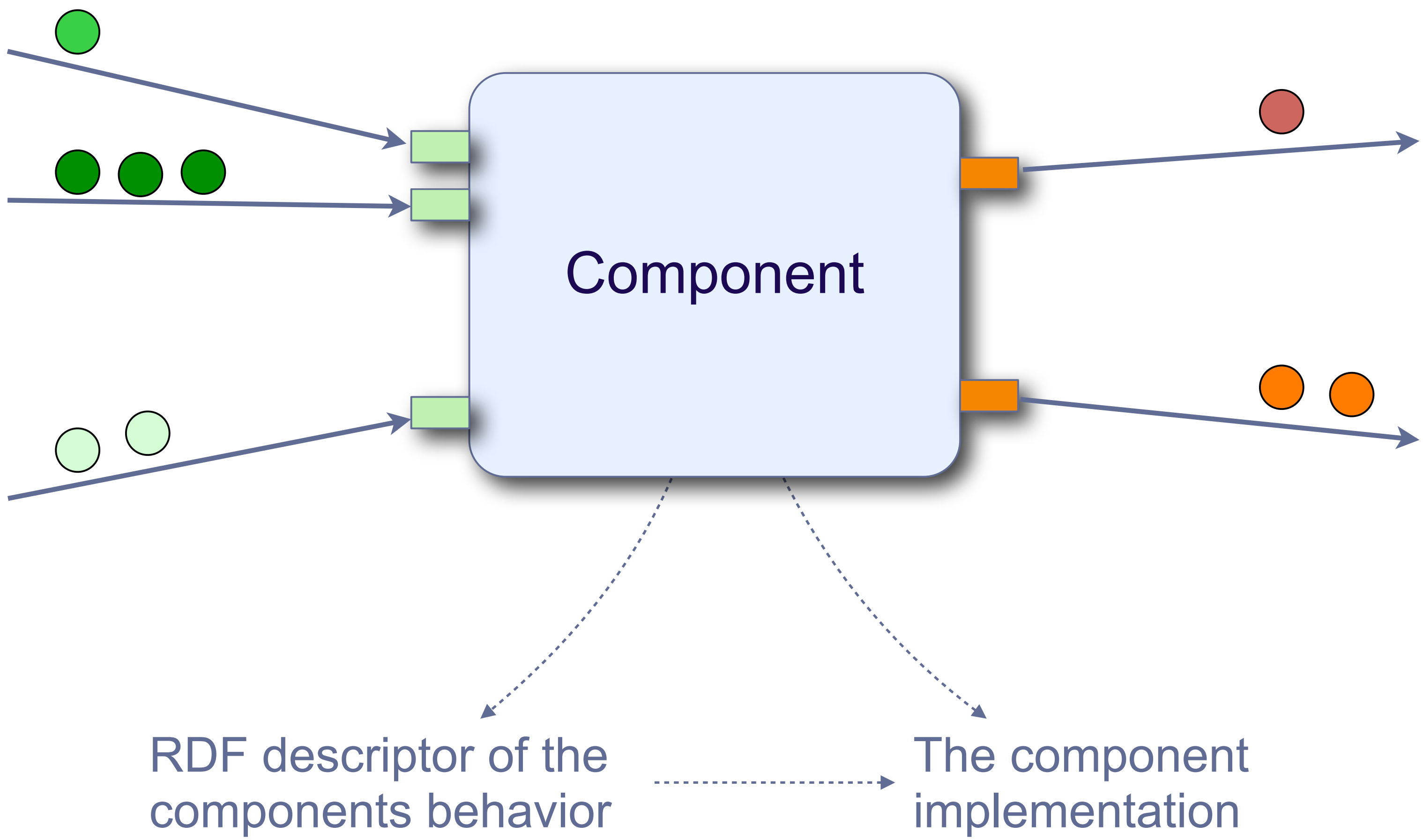
Stateless data flow



State-based data flow

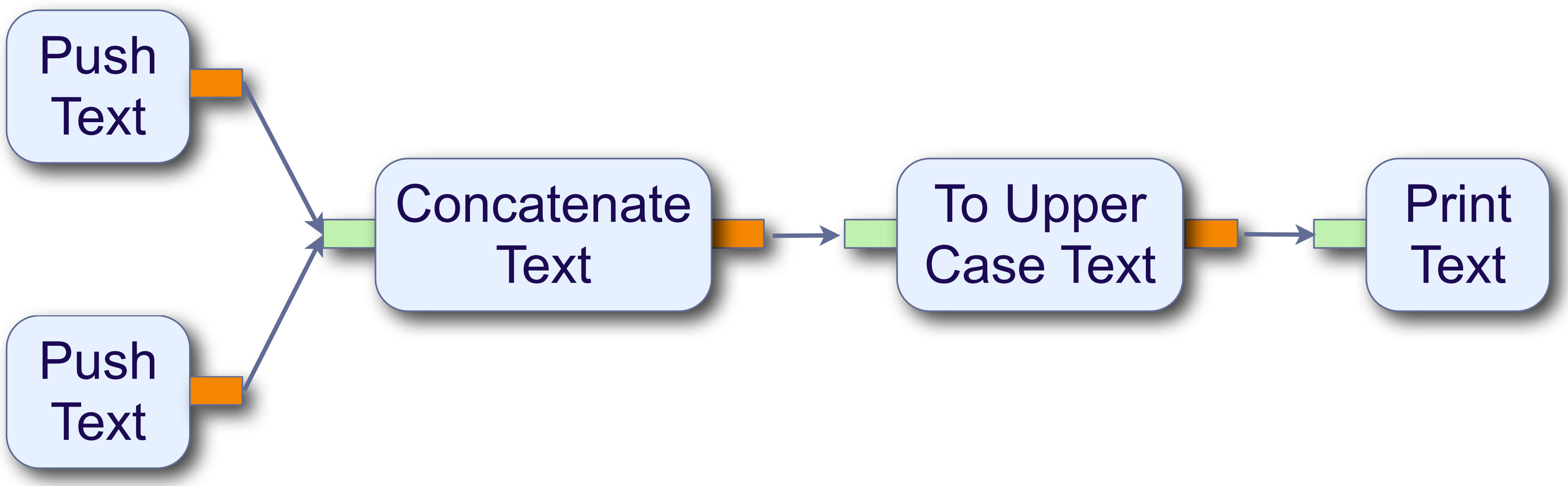


The Basic Building Blocks: Components



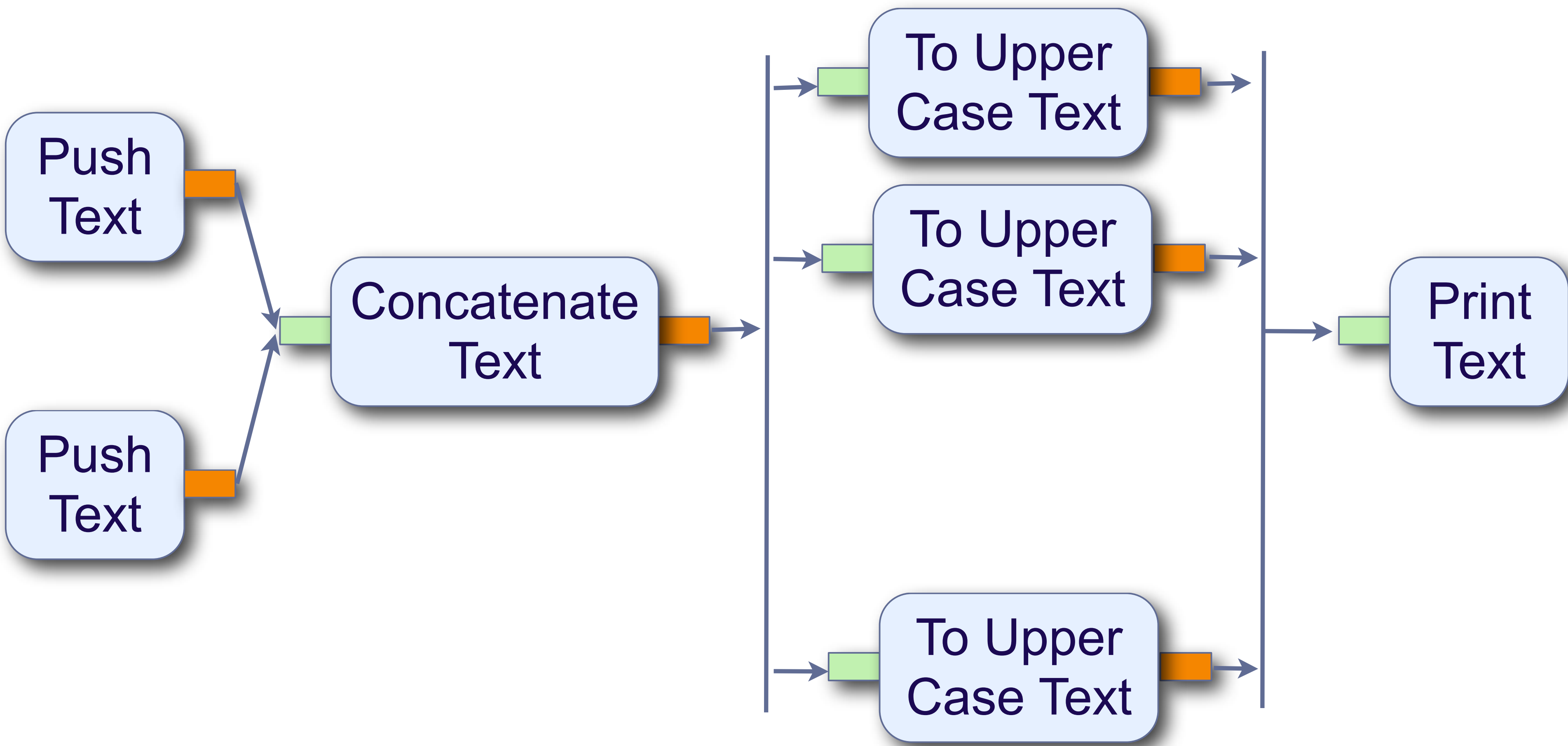
Go with the Flow: Creating Complex Tasks

- Directed multigraph of components creates a flow



Automatic Parallelization: Speed and Robustness

- Meandre ZigZag language allow automatic parallelization



GAs and Competent GAs

Selectorecombinative GAs

1. Initialize the population with random individuals
2. Evaluate the fitness value of the individuals
3. Select good solutions by using *s*-wise tournament selection without replacement *(Goldberg, Korb & Deb, 1989)*
4. Create new individuals by recombining the selected population using uniform crossover (Sywerda, 1989)
5. Evaluate the fitness valueof all offspring
6. Repeat steps 3-5 until convergence criteria are met

Extended Compact Genetic Algorithm

- Harik et al. 2006
- Initialize the population (usually random initialization)
- Evaluate the fitness of individuals
- Select promising solutions (e.g., tournament selection)
- Build the probabilistic model
  - Optimize structure & parameters to best fit selected individuals
  - Automatic identification of sub-structures
- Sample the model to create new candidate solutions
  - Effective exchange of building blocks
- Repeat steps 2–7 till some convergence criteria are met

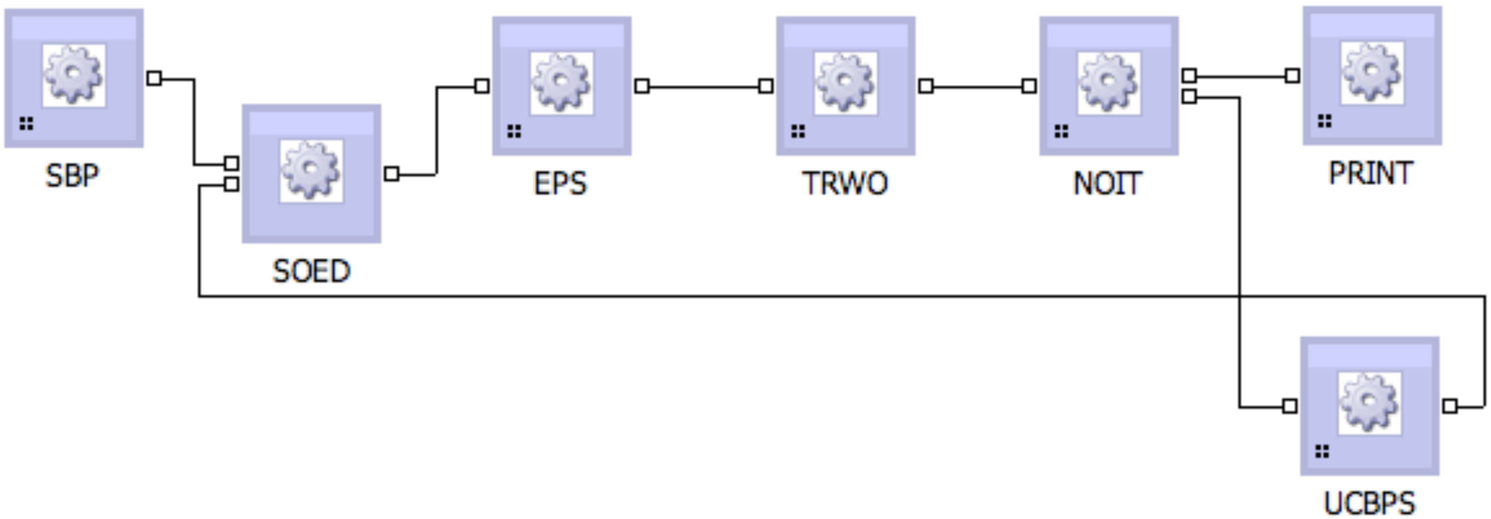
eCGA Model Building Process

- Use model-building procedure of extended compact GA
  - Partition genes into (mutually) independent groups
  - Start with the lowest complexity model
  - Search for a least-complex, most-accurate model

Model Structure	Metric
$[X_0]$ $[X_1]$ $[X_2]$ $[X_3]$ $[X_4]$ $[X_5]$ $[X_6]$ $[X_7]$ $[X_8]$ $[X_9]$ $[X_{10}]$ $[X_{11}]$	1.0000
$[X_0]$ $[X_1]$ $[X_2]$ $[X_3]$ $[X_4X_5]$ $[X_6]$ $[X_7]$ $[X_8]$ $[X_9]$ $[X_{10}]$ $[X_{11}]$	0.9933
$[X_0]$ $[X_1]$ $[X_2]$ $[X_3]$ $[X_4X_5X_7]$ $[X_6]$ $[X_8]$ $[X_9]$ $[X_{10}]$ $[X_{11}]$	0.9819
$[X_0]$ $[X_1]$ $[X_2]$ $[X_3]$ $[X_4X_5X_6X_7]$ $[X_8]$ $[X_9]$ $[X_{10}]$ $[X_{11}]$	0.9644
…	
$[X_0]$ $[X_1]$ $[X_2]$ $[X_3]$ $[X_4X_5X_6X_7]$ $[X_8X_9X_{10}X_{11}]$	0.9273
…	
$[X_0X_1X_2X_3]$ $[X_4X_5X_6X_7]$ $[X_8X_9X_{10}X_{11}]$	0.8895

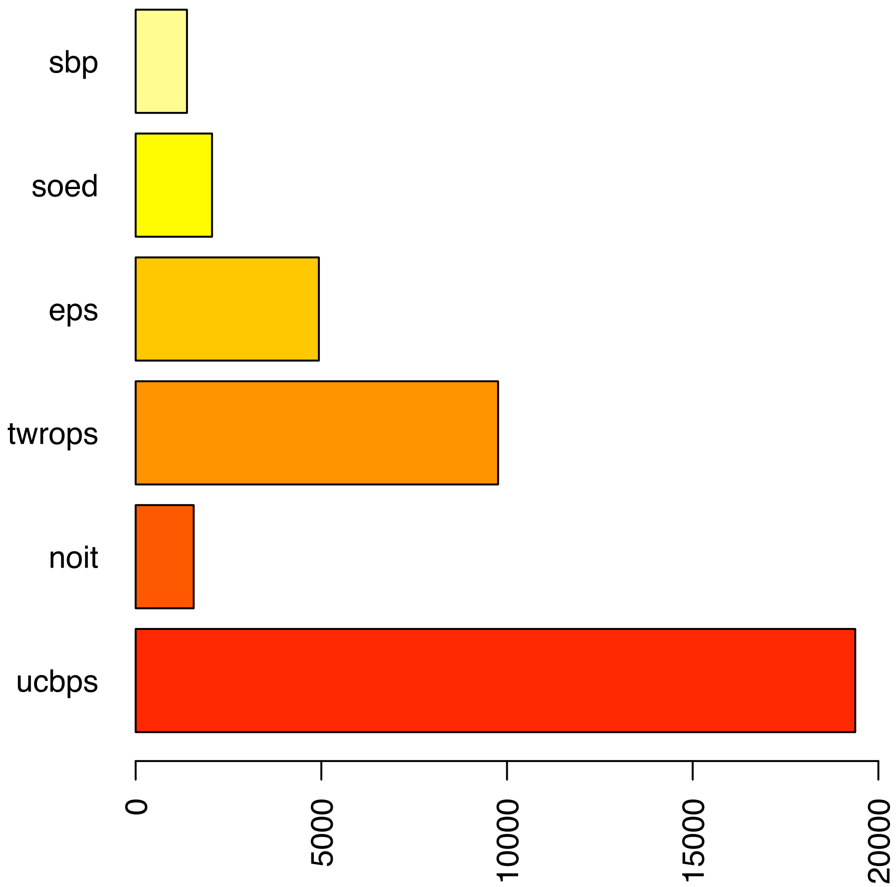
Data-Intensive Flows for Competent GAs

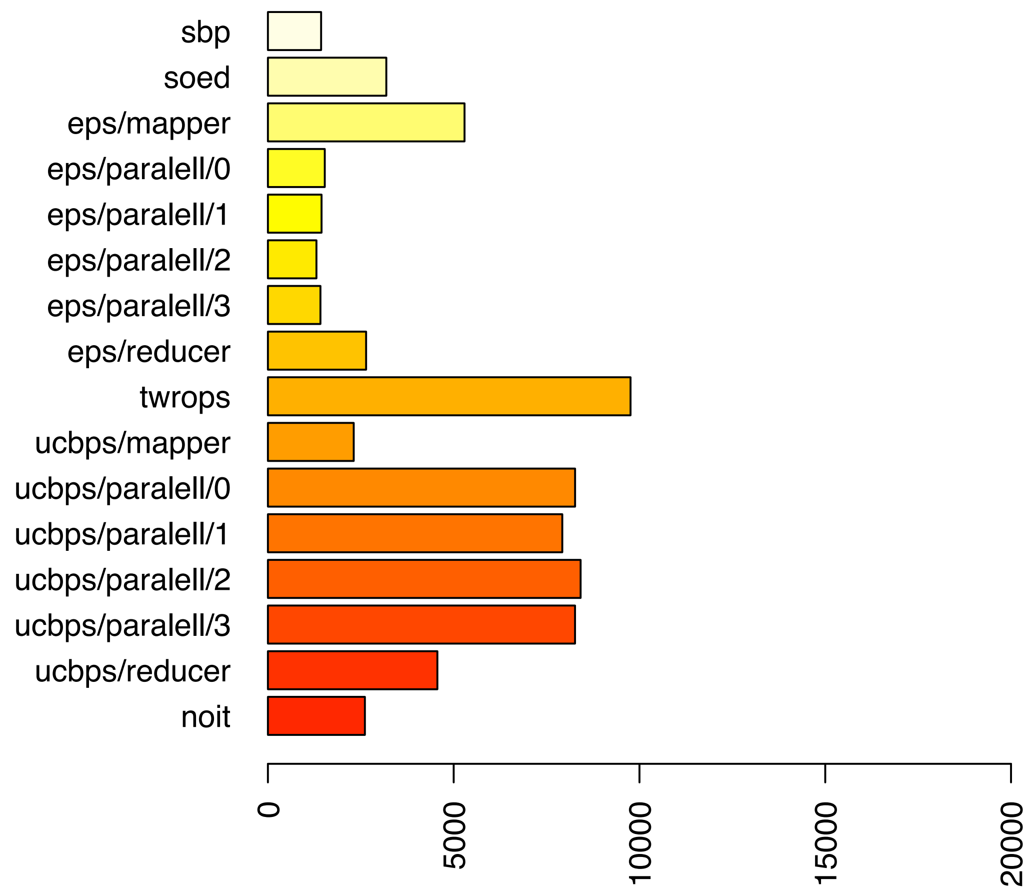
Selectorecombinative GA



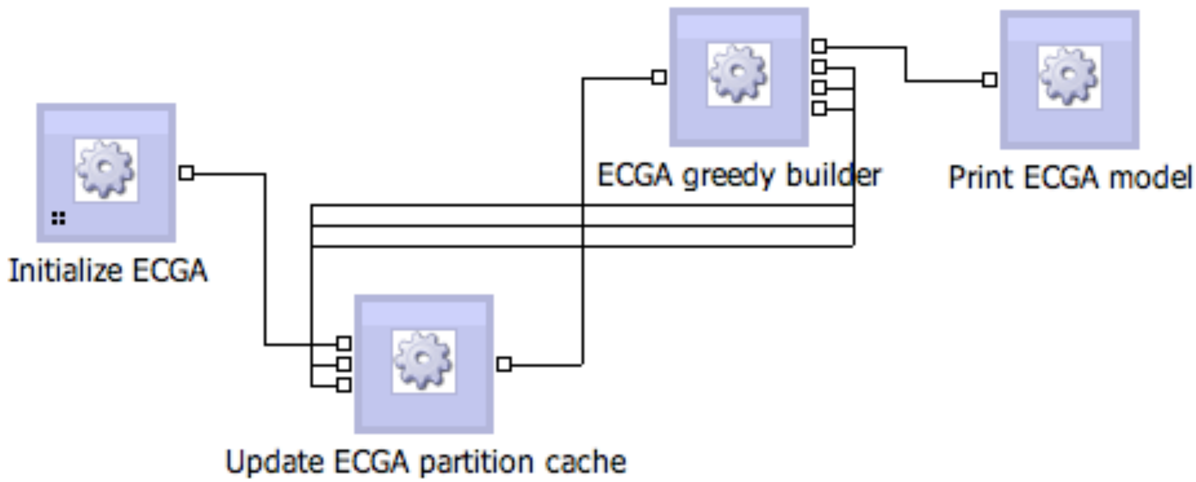
sGAs Execution Profile and Parallelization

- Intel 2.8Ghz QuadCore, 4Gb RAM. Average of 20 runs.



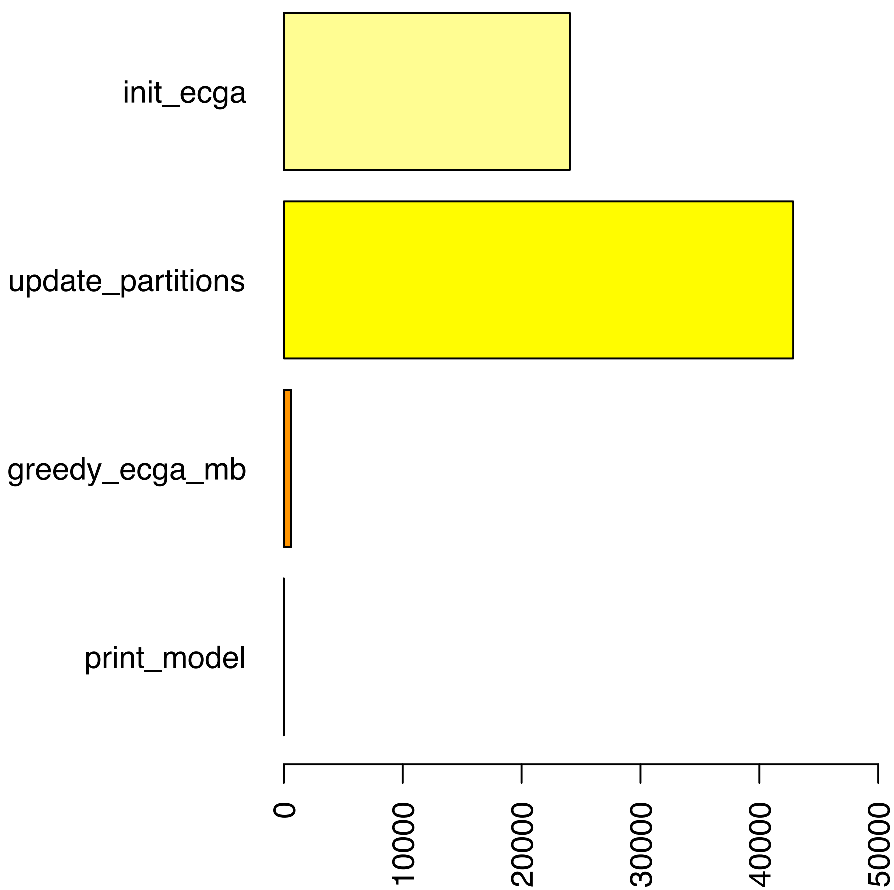


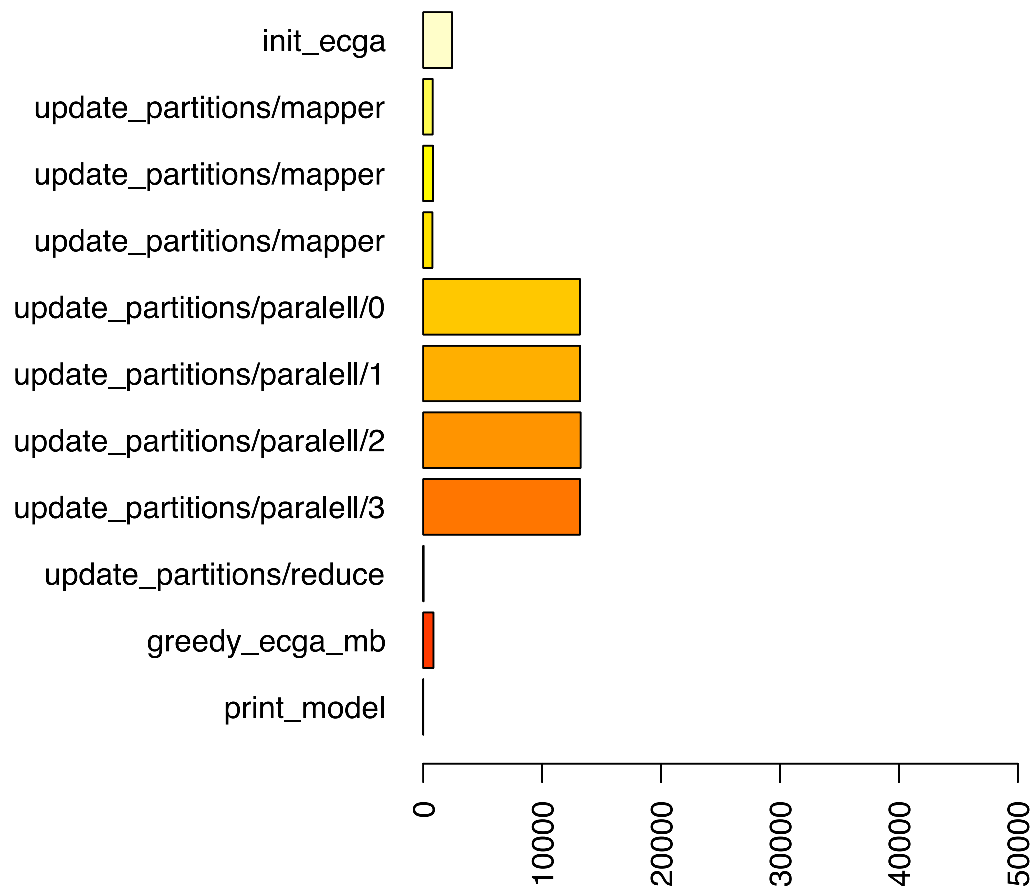
eCGA Model Model building



eCGA Execution Profile and Parallelization

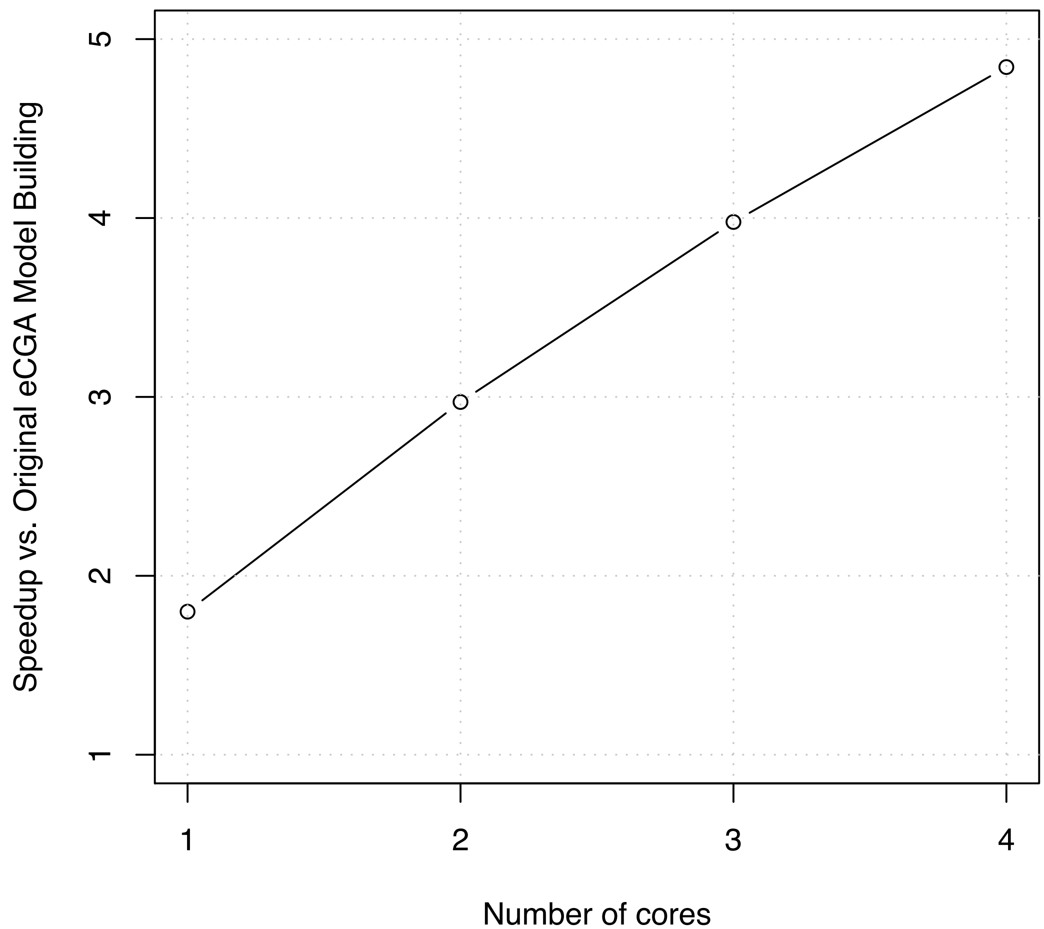
- Intel 2.8Ghz QuadCore, 4Gb RAM. Average of 20 runs.





eCGA Model Building Speedup

- Intel 2.8Ghz QuadCore, 4Gb RAM. Average of 20 runs.
- Speedup against original eCGA model building



Scalability on NUMA Systems

- Run on NCSA's SGI Altix Cobalt
- 1,120 processors and up to 5 TB of RAM
- SGI NUMAlink
- NUMA architecture
- Test for speedup behavior
- Average of 20 independent runs
- Automatic parallelization of the partition evaluation
- Results still show the linear trend (despite the NUMA)
  - 16 processors, speedup = 14.01
  - 32 processors, speedup = 27.96

Wrapping Up

Summary

- Evolutionary computation is data rich
- Data-intensive computing can provide to EC:
  - Tap into parallelism quite painless
  - Provide a simple programming and modeling
  - Boost reusability
  - Tackle otherwise intractable problems
- Shown that equivalent data-intensive computing versions of traditional algorithms exist
- Linear parallelism can be tap transparently

GECCO 2009
Montréal July 8-12, 2009
www.sigevo.org/gecco-2009/

Data-Intensive Computing for Competent Genetic Algorithms: A Pilot Study using Meandre

Xavier Llorà

National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
Urbana, Illinois, 61801

xllora@ncsa.illinois.edu
http://www.ncsa.illinois.edu/~xllora