A Lightweight, Scalable Grid Computing Framework for Parallel Bioinformatics Applications

Hans De Sterck[1], Rob Markel[2], and Rob Knight[3]

[1]Department of Applied Mathematics, University of Waterloo (hdesterck@uwaterloo.ca)

[2]Scientific Computing Division, National Center for Atmospheric Research, Boulder, Colorado, USA

[3]Department of Chemistry and Biochemistry, University of Colorado at Boulder, USA

First SHARCNET/HPCVL Joint Symposium on High Performance Computing and Applications, Ryerson University, Toronto, Tuesday, October 11, 2005

< > - +

Outline

(1) Introduction: why parallel bioinformatics on computational grids

(2) The TaskSpaces framework for grid computing

(3) Application: finding correctly folded active RNA motifs

(4) Conclusions and future work

(1) Introduction: why parallel bioinformatics

- bioinformatics: large data sets
  - genomics: 28.5 billion basepairs in Genbank (2002 release), 200.000 species
  - proteomics: millions of data files per sample

- bioinformatics: computationally expensive algorithms
  - phylogenetics: exponential in number of species
  - given a random RNA molecule, what is the probability that it has a specific chemical activity? (e.g. binding Isoleucine) $\Rightarrow$ need to fold billions of short randomized RNA sequences

$\Rightarrow$ 1 CPU is not enough, need distributed/parallel computing for bioinformatics

Introduction: why grid computing?

- where can I find computational power?
  - my desktop machine (1 CPU)
  - other computers in my lab ($\sim$ 10 CPUs)
  - linux cluster at CS department ($\sim$ 100 CPUs), SHARCNET
  - superclusters at SHARCNET ($\sim$ 1000 CPUs, at Waterloo)
  - parallel supercomputers at SHARCNET ($\sim$ at McMaster)
- ‘dream’: depending on problem size, I want to be able to run the problem on any of these machines
  (or all at the same time when I have a very large problem, or when I have to beat a deadline!)

why grid computing?

- problems with this ‘dream’:
  - variety of hardware
  - variety of operating systems, OS versions
  - variety of software versions
  - variety of queueing systems
  - need to install and maintain code on all those machines?
  - do I distribute work/data by hand? scripts? how are results centralized?
  - what happens to scripts when machines are added, removed?
  - . . .

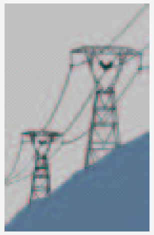
University of Waterloo

< > - +

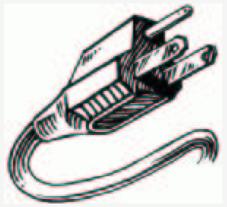
grid computing

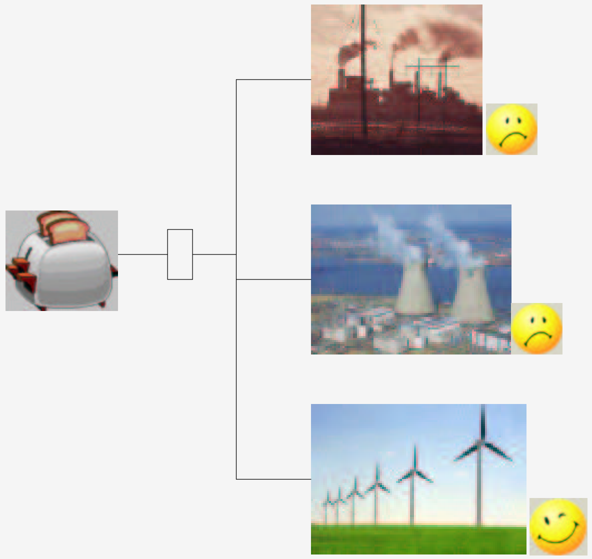
- an answer: concept of ‘grid computing’
- analogy: power grid
  - user wants electrical power
  - user doesn’t care where the power is produced

    ‘electrical power is interchangeable commodity’

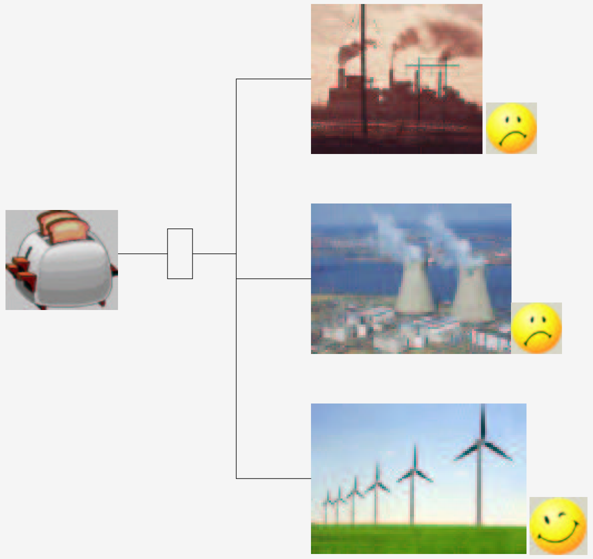


(1) standard interface: electrical plug





grid computing



(2) scalable:

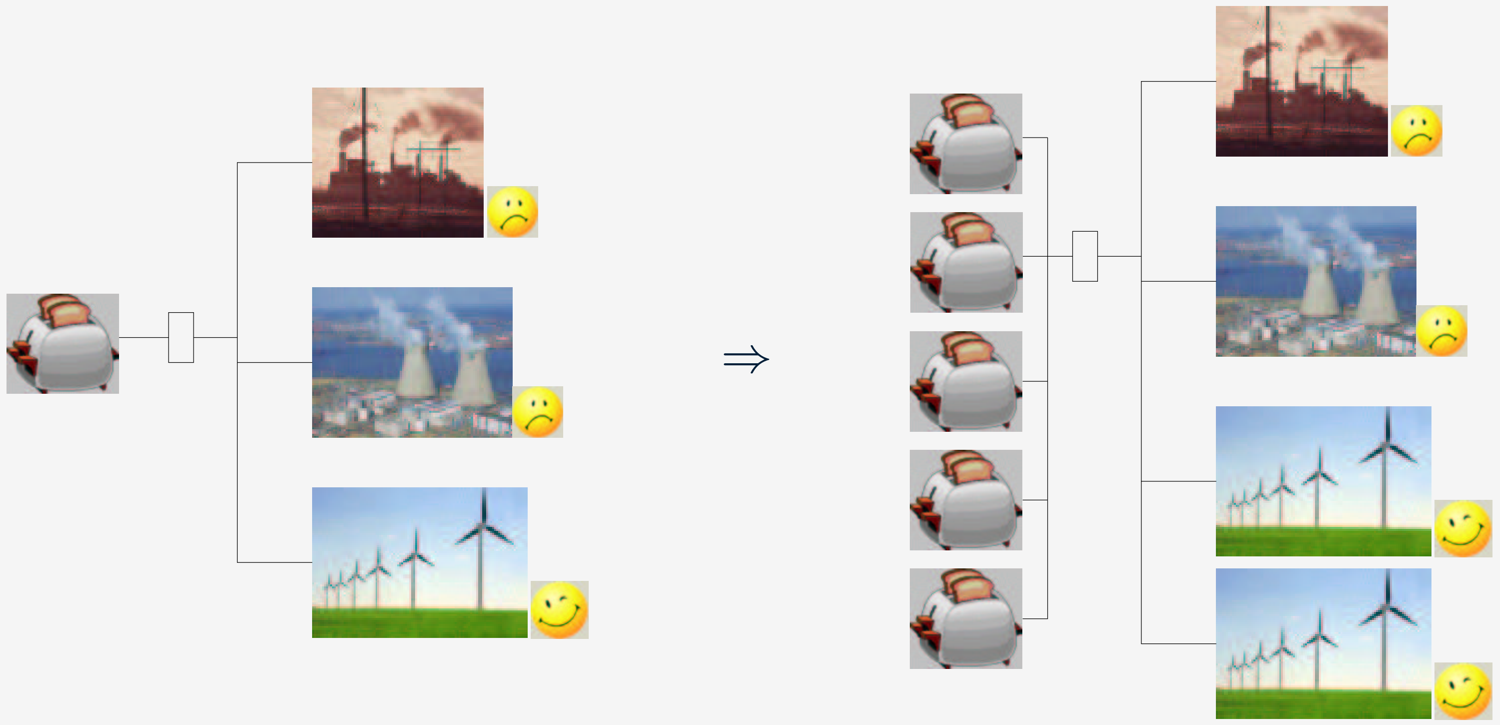
user: many appliances = many plugs

producer: when extra power is needed, switches in extra power plants

University of Waterloo

< > - +

grid computing



(2) scalable:

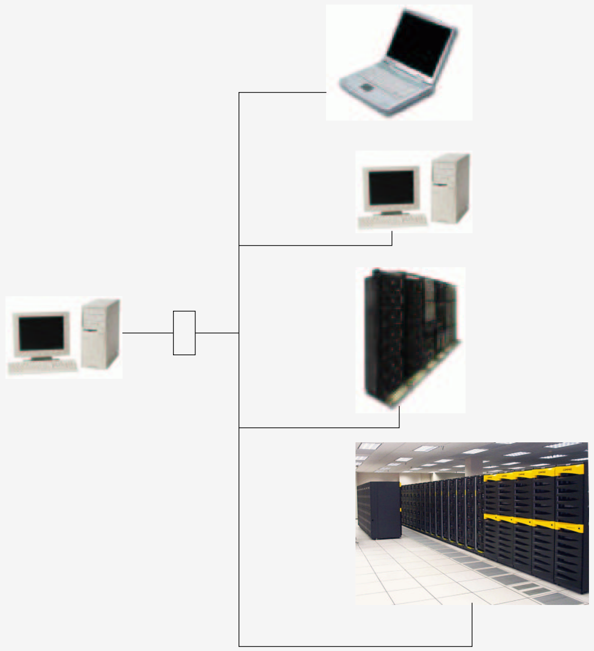
user: many appliances = many plugs

producer: when extra power is needed, switches in extra power plants

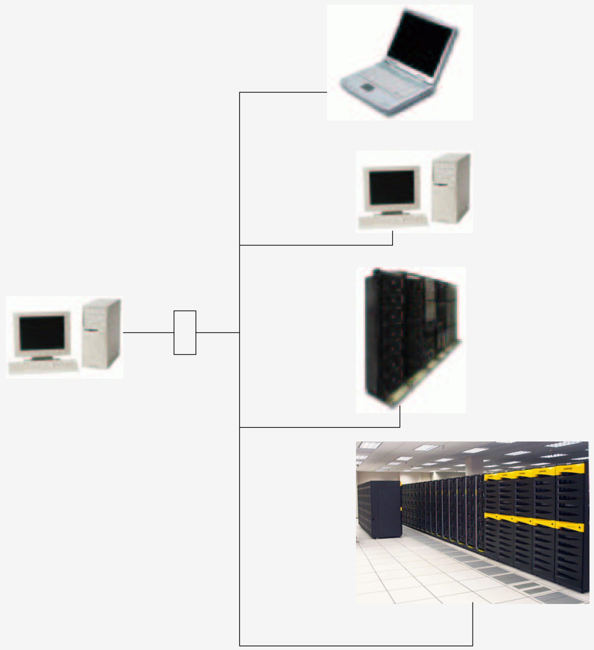
grid computing

- grid computing: analogous concept
  - user wants computing power
    - computing cycles
    - storage
    - network capacity
  - user doesn't care where the power is produced
    'computing power is interchangeable commodity'

(1) standard interface



grid computing

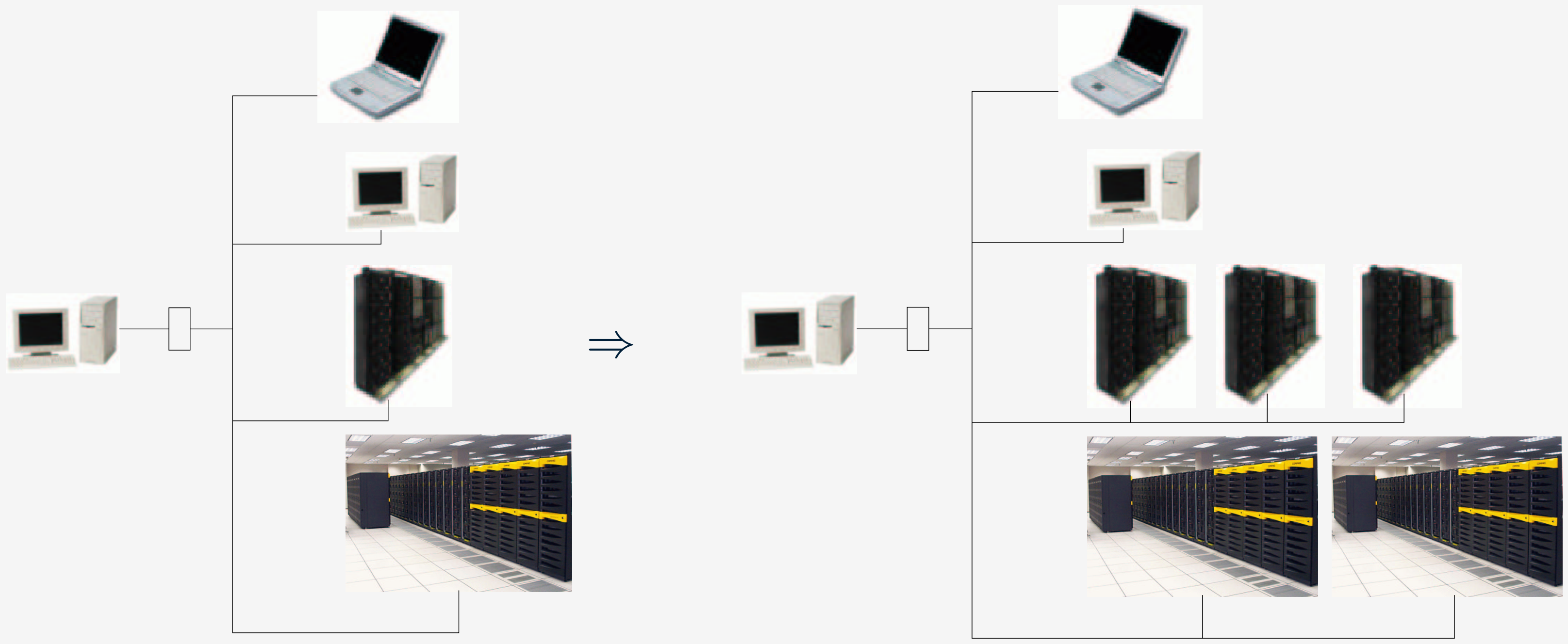


(2) scalable:

user: many applications at the same time

producer: when extra power is needed, switches in extra 'computing plants' (compute engines)

grid computing



(2) scalable:

user: many applications at the same time

producer: when extra power is needed, switches in extra 'computing plants' (compute engines)

grid computing

- the analogy is not perfect: computing power $\neq$ electrical power
- some additional requirements for computational grids:

(3) secure resource sharing

information is not anonymous (unlike electrical power)

(4) ‘transactions’, fault-tolerance

information is not replaceable (unlike electrical power)

(5) resource allocation, scheduling (queues)

large computers work with queueing systems

(6) install and compile application code and data on compute engines

variety of hardware, OS, OS and software versions, . . .

(7) communication between processors for parallel computing

(2) one approach... : TaskSpaces framework for grid computing

two major design choices for grid computing prototype framework:

(A) use the Java language

- concept of 'virtual machine':
  - behaves the same on all OS, hardware
  - 'executable byte-code' fully interchangeable
- has built-in security support
- object = data + code
- has built-in networking

$\Rightarrow$ provides natural and adequate answers to

(1) standard interface

(3) secure resource sharing

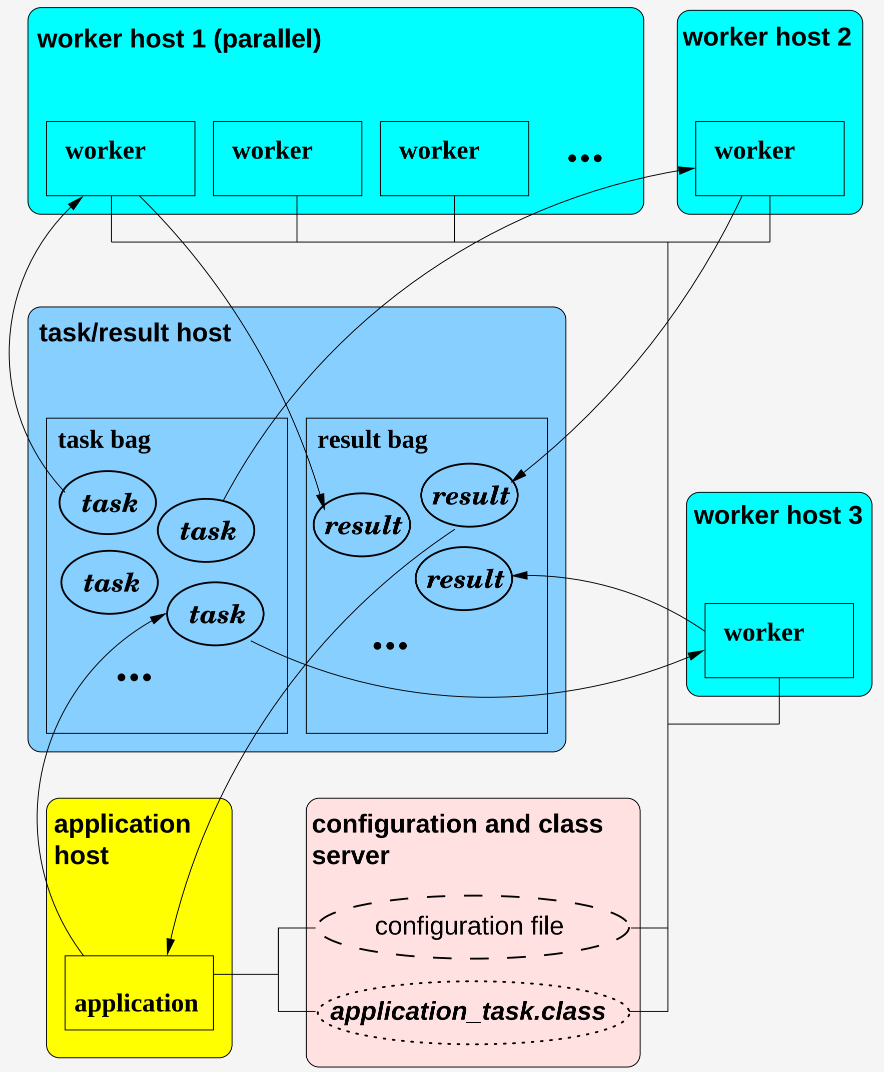
(6) install and compile application code on compute engines: not needed! $\longrightarrow$ download task objects = code + data

(7) communication between processors: send objects

< > - +

TaskSpaces: design choices

(B) use tuple space concept, bag-of-tasks paradigm



- space and time decoupling
- worker executable < 2 kb

⇒ provides natural and adequate answers to

(2) scalable:
user: concurrent applications in one or several bags
producer: switches in extra ‘worker farms’

(5) resource allocation, scheduling: task bag is simple decoupled ‘superqueue’

high throughput grid experiment

Blue Horizon, SDSC, San Diego, CA (4 workers/processor)	64	128	240
P4 Linux, CU Boulder, CO (2 workers/processor)	4	4	4
Itanium Linux, CU Boulder, CO (2 workers/processor)	4	4	4
forseti1, NCSA, Urbana, IL (1 worker/processor)	16	16	16
hermod, NCSA, Urbana, IL (1 worker/processor)	16	16	16
Total number of workers	104	168	280
Total execution time	105s	103s	101s

High throughput grid experiment (50 Jacobi iterations, 500x500 grid points per worker). Number of worker processes and total execution times are shown. Problem size is constant per worker process.

$\Longrightarrow$ TaskSpaces framework: parallel scientific computing with intertask communication is scalable on the internet!

University of Waterloo

< > - +

TaskSpaces grid computing framework

(1) standard interface: Java task object

(2) scalable: through bag-of-tasks
user: many applications at the same time
producer: switches in extra ‘computing plants’

(3) secure resource sharing: digital certificates, to be implemented

(4) ‘transactions’, fault-tolerance: rudimentary, needs more thinking

(5) resource allocation, scheduling: task bag = decoupled superqueue, rudimentary

(6) get application code on compute engines: download task object

(7) interprocessor communication for parallel computing: send objects

(8) user gets charged according to use: to be implemented

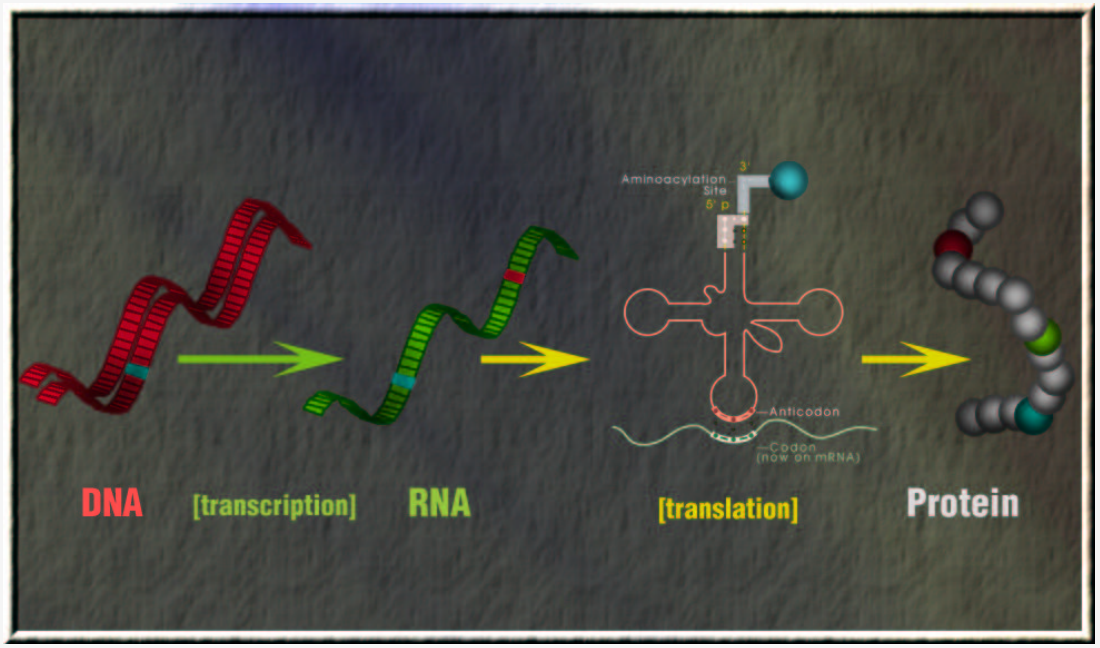
(9) Quality of Service

(10) resource discovery

University of Waterloo

< > - +

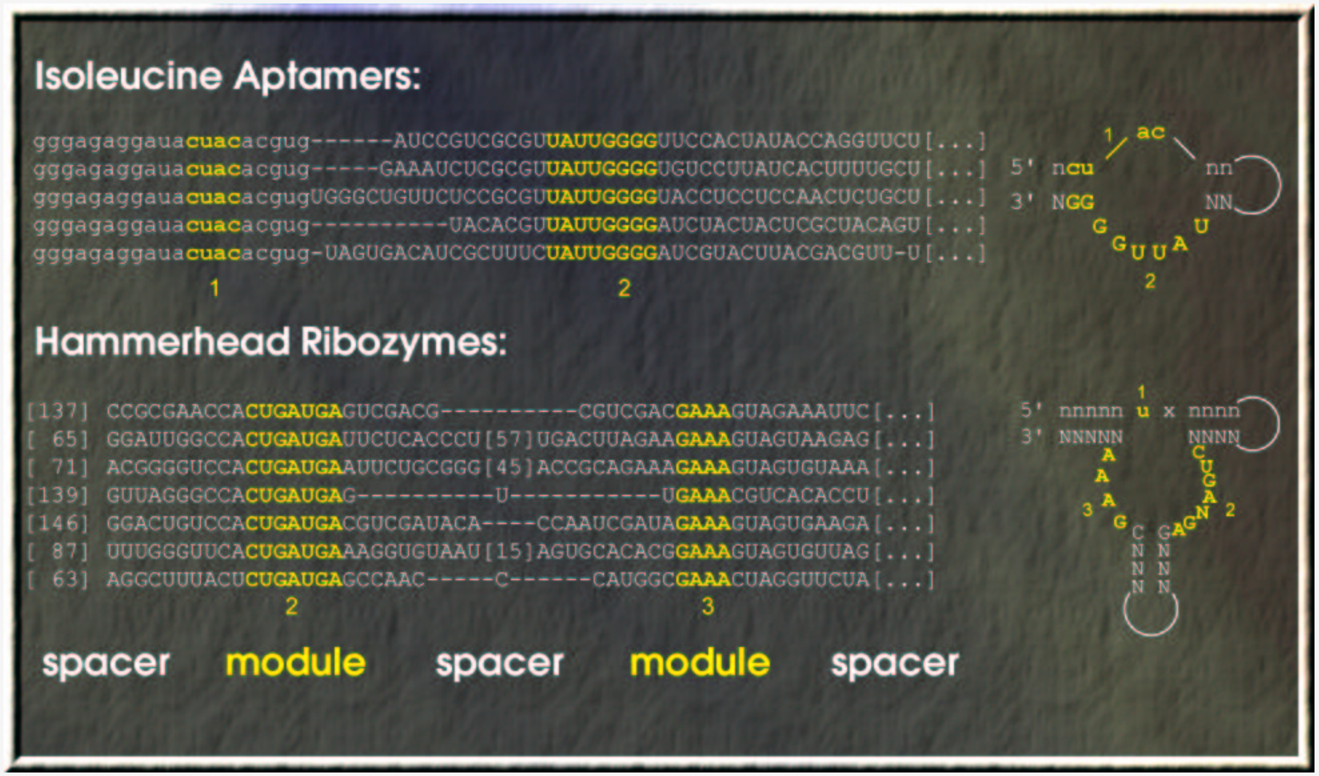
(3) Application: finding correctly folded active RNA motifs



- modern cells mostly use RNA molecules as *passive message*, but they also use RNA as an *active catalyst*
  - example: Hammerhead Ribozyme catalyzes cleavage of RNA
- RNA molecules that bind to specific molecules can experimentally be selected (SELEX)
  - example: Isoleucine Aptamer binds to Isoleucine aminoacid

chemically actif motifs

- chemical activity characterized by specific *motif*
- *motif* has several *modules* with specified paired bases
- the *modules* are separated by arbitrary *spacer*



Isoleucine Aptamers:

```
gggagaggauacuacacgug------AUCCGUCGCGUUAUUGGGGUUCCACUAUACCAGGUUCU[...]
gggagaggauacuacacgug-----GAAAUCUCGCGUUAUUGGGGUGUCCUUAUCACUUUUGCU[...]
gggagaggauacuacacgugUGGGCUGUUCUCCGCGUUAUUGGGGUACCUCCUCCAACUCUGCU[...]
gggagaggauacuacacgug---------UACACGUUAUUGGGGAUCUACUACUCGCUACAGU[...]
gggagaggauacuacacgug-UAGUGACAUCGCUUUCUAUUGGGGAUCGUACUUACGACGUU-U[...]
                1                    2
```

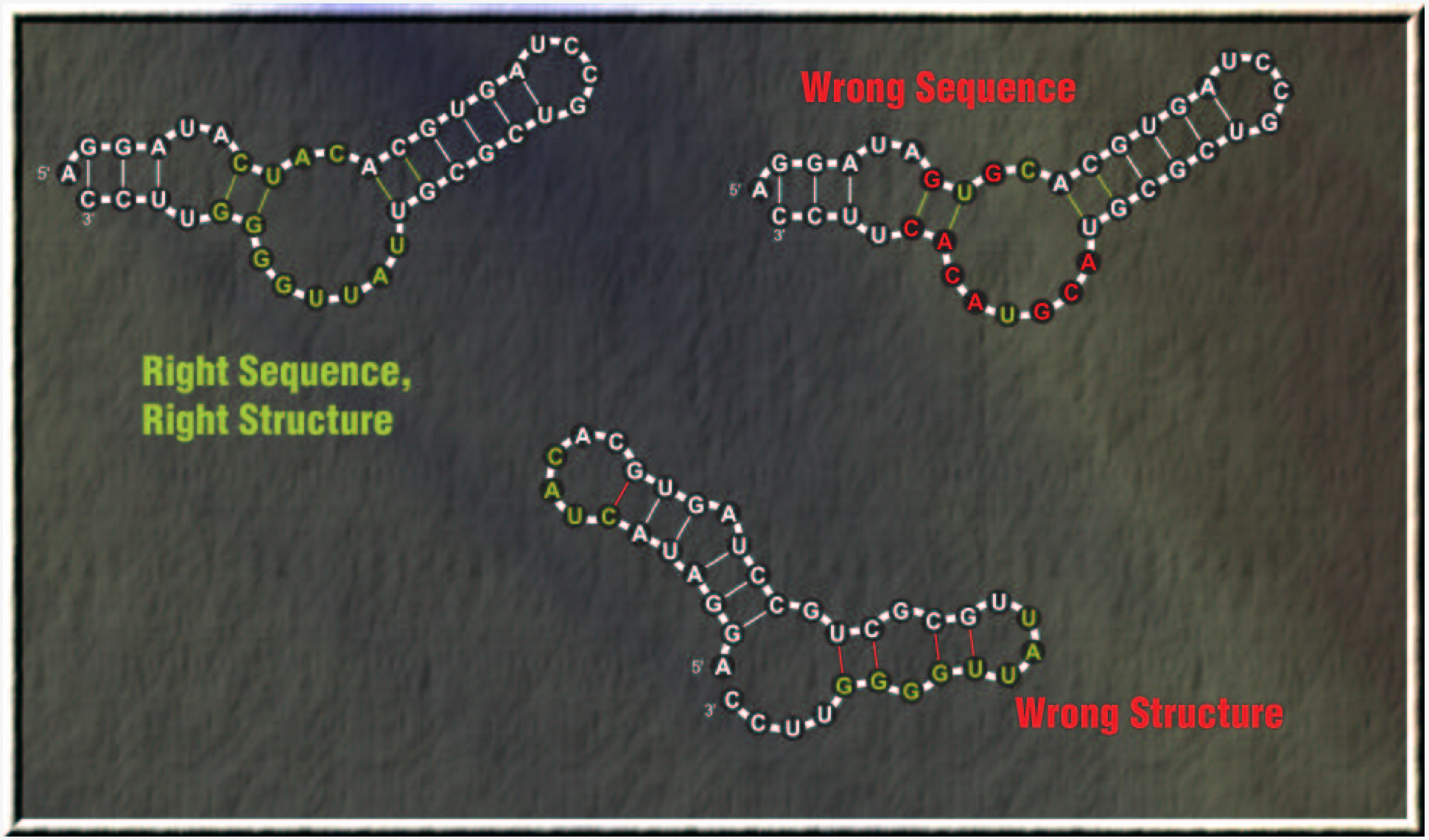
Hammerhead Ribozymes:

```
[137] CCGCGAACCACUGAUGAGUCGACG----------CGUCGACGAAAGUAGAAAUUC[...]
[ 65] GGAUUGGCCACUGAUGAUUCUCACCCU[57]UGACUUAGAAGAAAGUAGUAAGAG[...]
[ 71] ACGGGGUCCACUGAUGAAUUCUGCGGG[45]ACCGCAGAAAGAAAGUAGUGUAAA[...]
[139] GUUAGGGCCACUGAUGAG----------U-----------UGAAACGUCACACCU[...]
[146] GGACUGUCCACUGAUGACGUCGAUACA----CCAAUCGAUAGAAAGUAGUGAAGA[...]
[ 87] UUUGGGUUCACUGAUGAAAGGUGUAAU[15]AGUGCACACGGAAAGUAGUGUUAG[...]
[ 63] AGGCUUUACUCUGAUGAGCCAAC-----C------CAUGGCGAAACUAGGUUCUA[...]
                  2                           3
```

spacer module spacer module spacer

sequence and structure

- right sequence and right structure are both important



(Isoleucine aptamer)

finding correctly folded active RNA motifs

QUESTION:

given a random RNA molecule of a certain length (e.g. 50 nucleotides), what is the probability that this RNA molecule

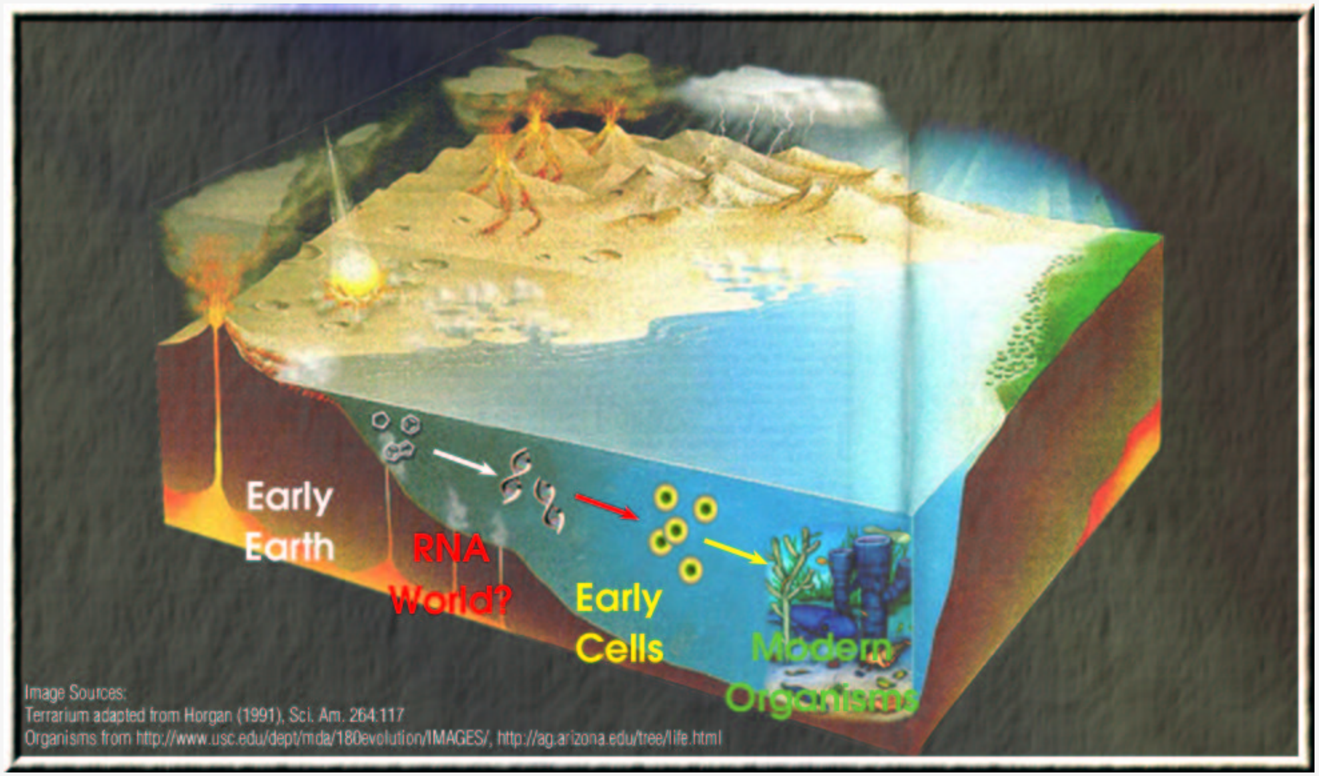
- contains the (Isoleucine aptamer) motif in the sequence

and

- folds in the correct (Isoleucine aptamer) structure

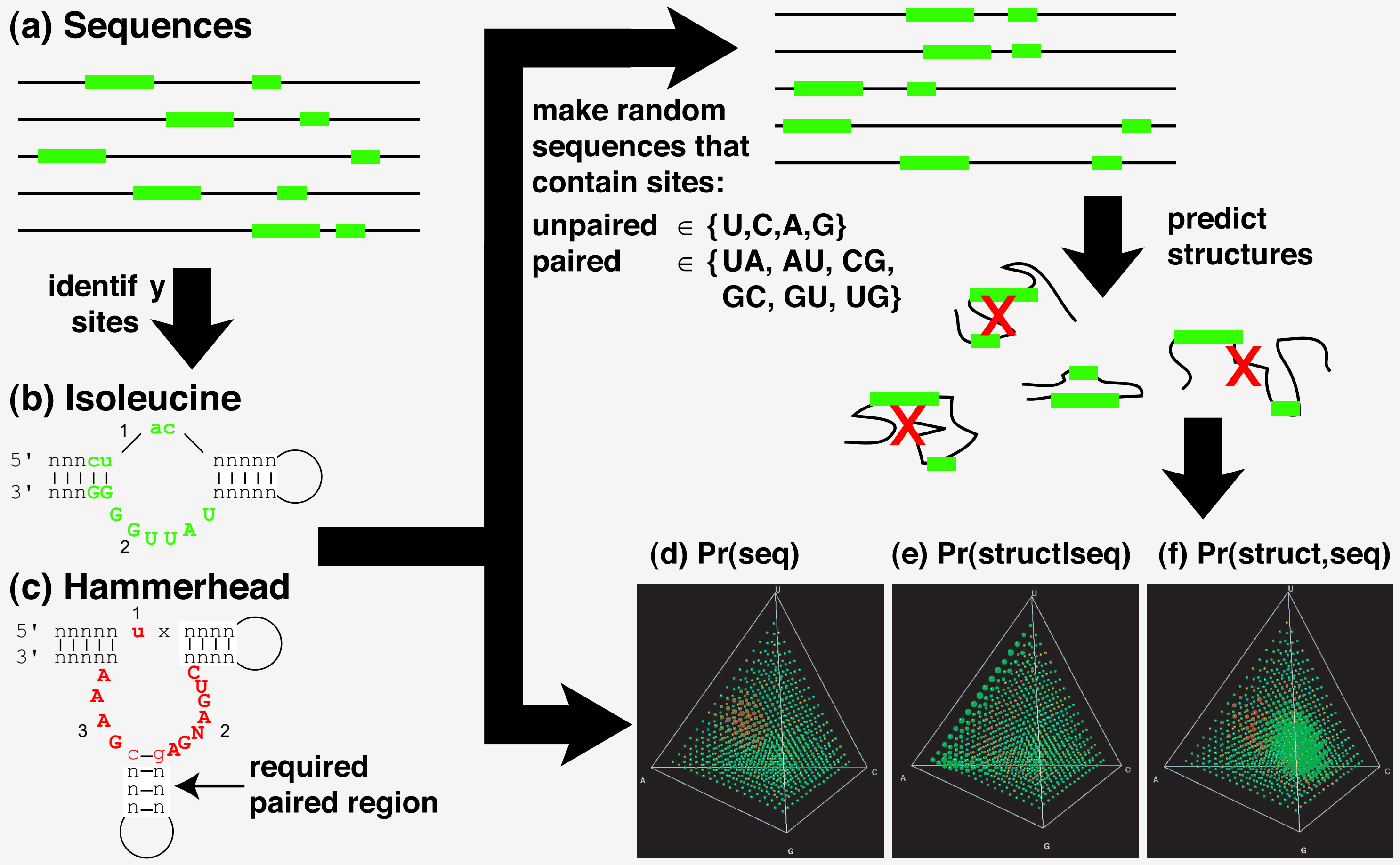
- applications:
  - **(1)** improve efficiency of SELEX experiments by biasing composition of random pool
  - **(2)** how likely is origin of an RNA world from a small number of random RNA molecules?

application: RNA world


Early Earth
RNA World?
Early Cells
Modern Organisms
Image Sources:
Terrarium adapted from Horgan (1991), Sci. Am. 264:117
Organisms from http://www.usc.edu/dept/mda/180evolution/IMAGES/, http://ag.arizona.edu/tree/life.html

if a small number of random RNA molecules has a reasonable probability of containing molecules with various chemical functions, then a primitive metabolism may have originated from random RNA molecules (some recent estimates propose that $\sim$ 1,000 molecules may be sufficient)

procedure for estimating probability

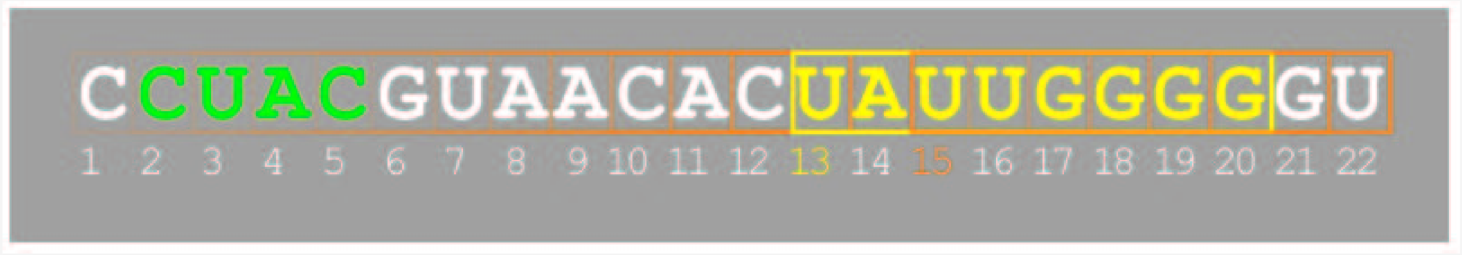


how to find this probability?

- finding $P(sequence\ and\ structure) \sim 10^{-11}$ by computational folding of fully random RNA molecules would take too long!! ($\sim 10^{12}$ molecules would need to be folded for each composition studied)
- instead: use $P(sequence\ and\ structure) =$

$$P(sequence) \cdot P(structure|sequence)$$

- Step 1: calculate $P(sequence)$ by combinatorial formula



$P(sequence) \sim 10^{-8}$ can be approximated well by combinatorial formula that is easily computable $\Rightarrow$ computer folding calculation of $P(structure|sequence) \sim 10^{-3}$ requires only $\sim 10^{4}$ sample size

University of Waterloo

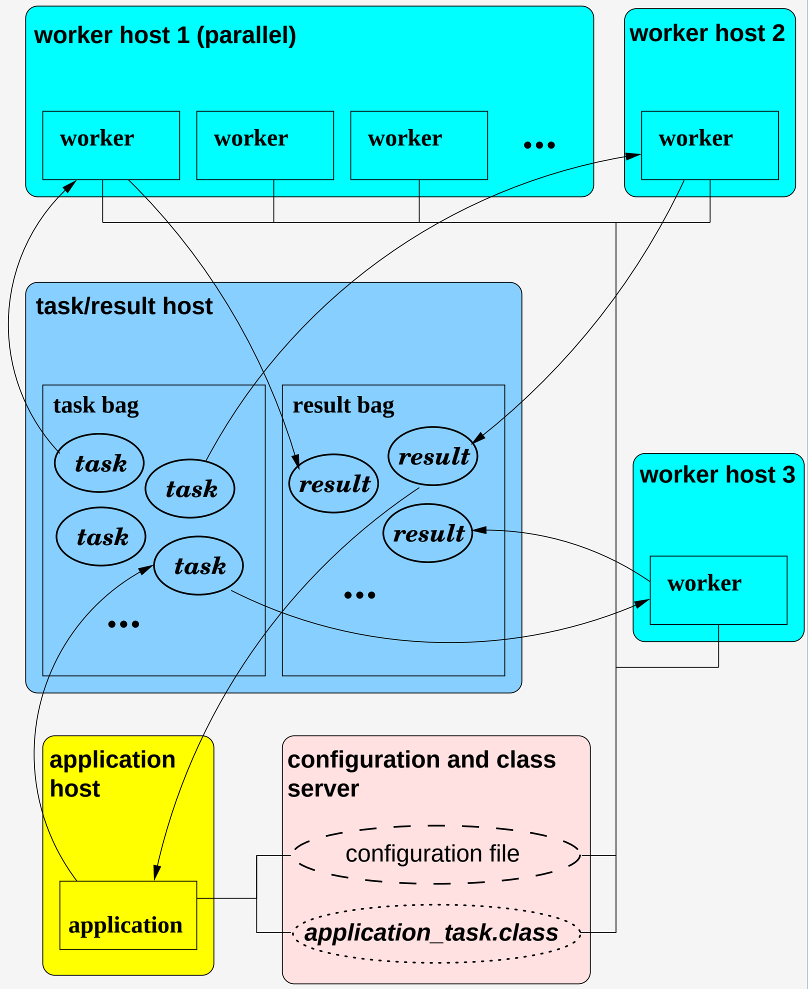
< > - +

use TaskSpaces

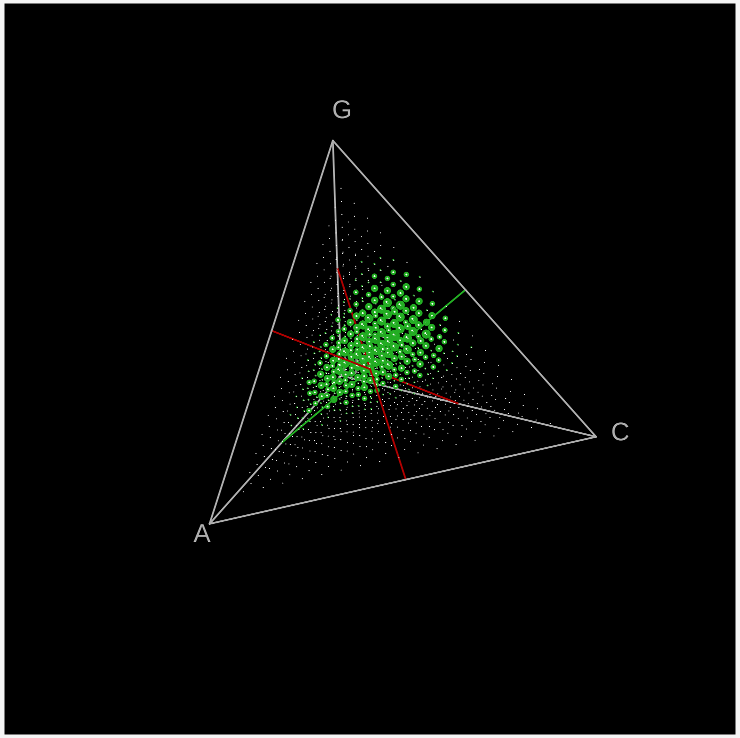
- example of one task = fold 10,000 randomized RNAs of length 50 that contain Isoleucine aptamer motif sequence, and count how many fold in the right structure
- use Vienna folding program
  - written in C
  - called from Java TaskSpaces framework on every processor
  - C needs to be compiled on every architecture
  - C executable can be downloaded with Java executable
- no communication between subtasks, so use TaskSpaces in taskfarming mode

TaskSpaces: example run

- 5% composition steps
  (=969 different compositions, tasks)
  100 nucleotides length
  10,000 random sequences per composition, . . .
  $\longrightarrow$ 77,520,000 foldings
- use Linux Cluster Platinum at NCSA (Illinois)
  $\sim$ 1000 processors
- use 260 processors, not all concurrently
  10,000 CPU hours



results



results for length 100

- Ile best at (15%, 25%, 35%, 25%) ACGU composition, factor 3.3 better than uniform composition
- HH best at (35%, 10%, 25%, 30%) ACGU composition, factor 13.6 better than uniform composition

- RNA world: $4.8\,10^9$ random 100-mers needed for Ile, $1.6\,10^{10}$ random 100-mers needed for HH = amount of RNA in 15,000 cells

$\Rightarrow$ too many for random RNA pool without preceding simpler self-reproducing system

(4) Conclusions and future work

- conclusions:
    - TaskSpaces prototype framework has been developed for parallel scientific computing on computational grids

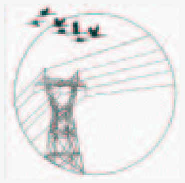
      ('A lightweight Java Taskspaces framework for scientific computing on computational grids', Proceedings of the ACM Symposium on Applied Computing, Track on Parallel and Distributed Systems and Networking, 1024–1030, 2003.)

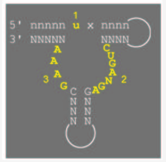
    - TaskSpaces has been applied to parallel bioinformatics problem: finding correctly folded active RNA motifs

      ('Abundance of correctly folded RNA motifs in sequence space, calculated on computational grids', Nucleic Acids Research, in press, 2005.)

Conclusions and future work

- work to be done:
  - continue development of TaskSpaces prototype (modularity, security, fault-tolerance, . . .)
  - prepare clean implementation, distribute
  - web interface for job submission, log file access, grid operator
  - try on SHARCNET
  - further bioinformatics applications





University of Waterloo

< > - +

A Lightweight, Scalable Grid Computing Framework for Parallel Bioinformatics Applications

Hans De Sterck[1], Rob Markel[2], and Rob Knight[3]

[1]Department of Applied Mathematics, University of Waterloo (hdesterck@uwaterloo.ca)

[2]Scientific Computing Division, National Center for Atmospheric Research, Boulder, Colorado, USA

[3]Department of Chemistry and Biochemistry, University of Colorado at Boulder, USA

First SHARCNET/HPCVL Joint Symposium on High Performance Computing and Applications, Ryerson University, Toronto, Tuesday, October 11, 2005

University of Waterloo

< > - +