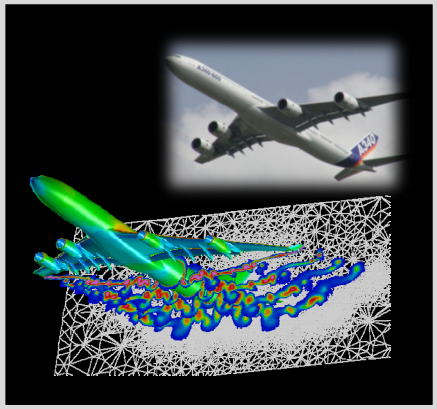


High Performance Computing

ADVANCED SCIENTIFIC COMPUTING

Prof. Dr. – Ing. Morris Riedel
Adjunct Associated Professor
School of Engineering and Natural Sciences, University of Iceland, Reykjavik, Iceland
Research Group Leader, Juelich Supercomputing Centre, Forschungszentrum Juelich, Germany

@Morris Riedel @MorrisRiedel @MorrisRiedel

SHORT LECTURE 13

Systems Biology & Bioinformatics

November 21, 2019
Webinar

UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

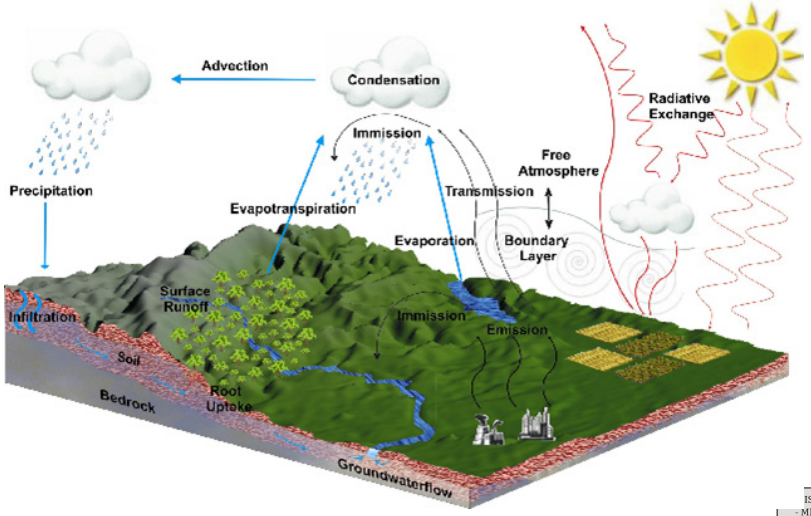
JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

DEEP Projects

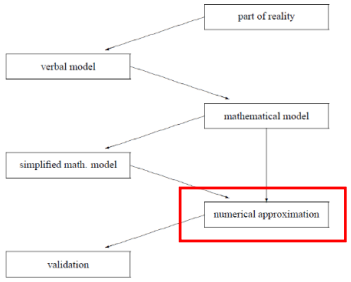
HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

HAICU HELMHOLTZ ARTIFICIAL INTELLIGENCE COOPERATION UNIT

Review of Short Lecture 12 – Terrestrial Systems & Climate



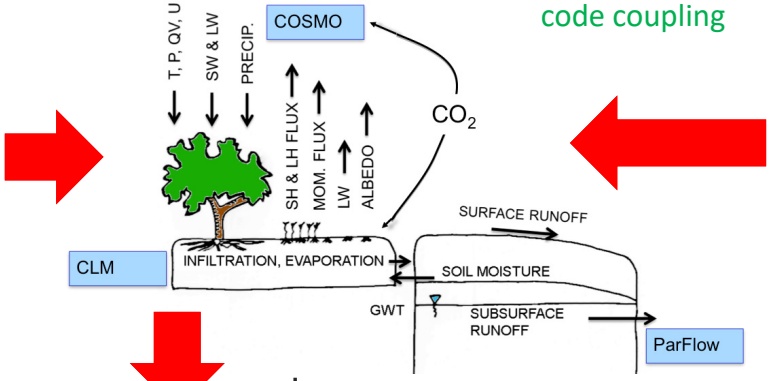
Physical system changes are accompanied by major state changes of land surfaces & ecosystems

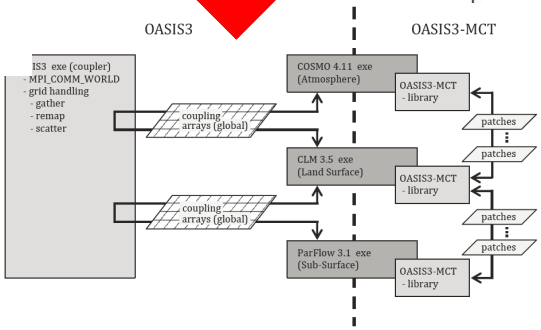


Solving Partial Differential Equations (PDEs) is often too complicated & computationally expensive or impossible to analytically compute driving the need for numerical approximation

modified from [1] SimLab Terrestrial Systems [2] F. Gasper et al.

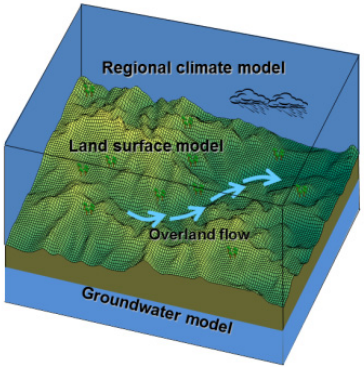
Towards reality: code coupling



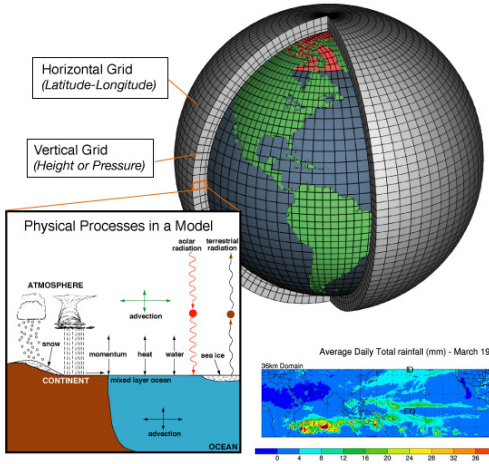


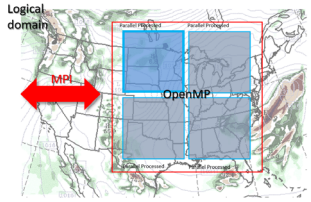
modified from [3] Wikipedia on 'Numerical Weather Prediction'

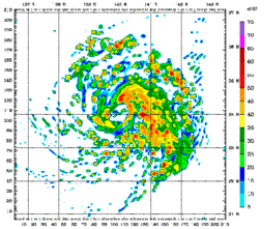
modified from [4] WRF – Code and Parallel Computing



Numerical Weather Prediction Example



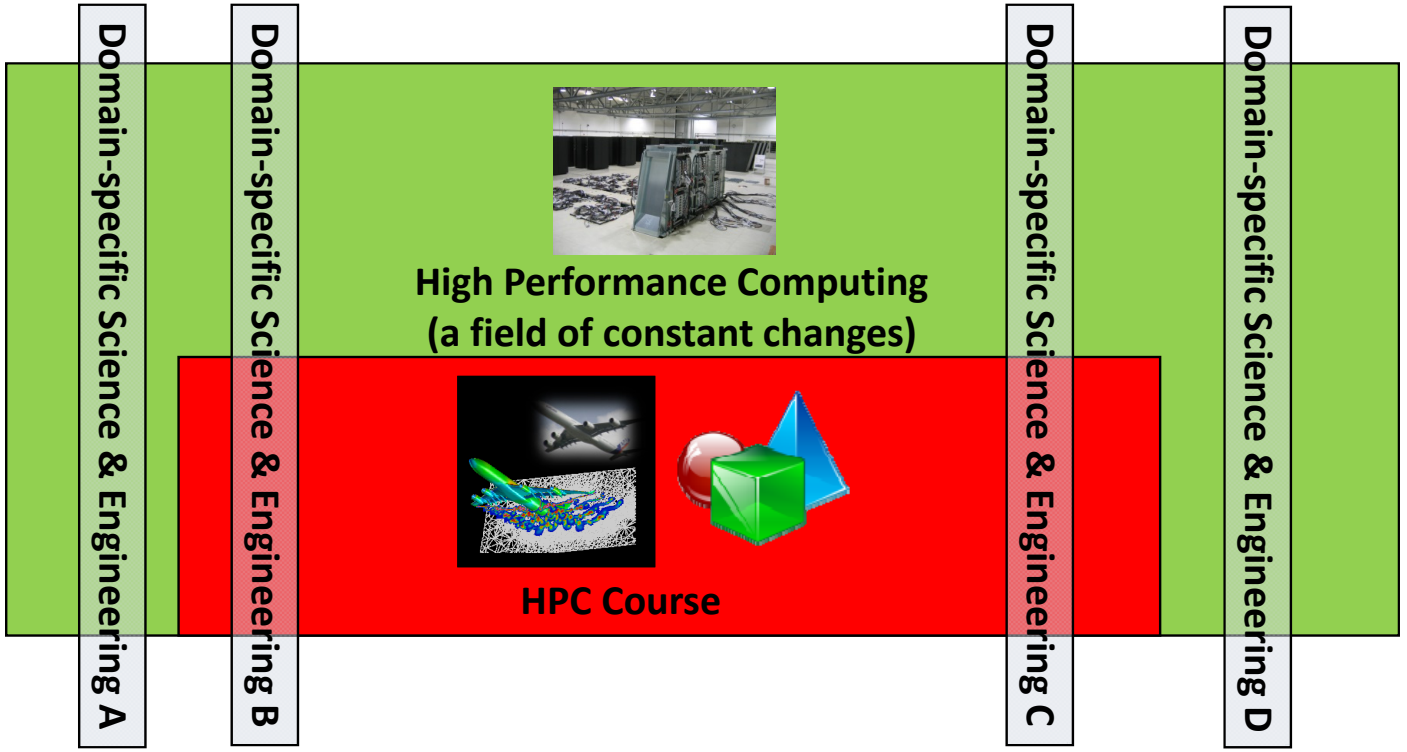




[5] Wikipedia on 'WRF'

HPC-A[dvanced] Scientific Computing (cf. Prologue) – Second Part

- Consists of techniques for programming & using large-scale HPC Systems
  - Approach: Get a broad understanding what HPC is and what can be done
  - Goal: Train general HPC techniques and systems and selected details of domain-specific applications



Outline of the Course

1. High Performance Computing
2. Parallel Programming with MPI
3. Parallelization Fundamentals
4. Advanced MPI Techniques
5. Parallel Algorithms & Data Structures
6. Parallel Programming with OpenMP
7. Graphical Processing Units (GPUs)
8. Parallel & Scalable Machine & Deep Learning
9. Debugging & Profiling & Performance Toolsets
10. Hybrid Programming & Patterns
11. Scientific Visualization & Scalable Infrastructures
12. Terrestrial Systems & Climate
13. Systems Biology & Bioinformatics
14. Molecular Systems & Libraries
15. Computational Fluid Dynamics & Finite Elements
16. Epilogue

+ additional practical lectures & Webinars for our hands-on assignments in context

- Practical Topics
- Theoretical / Conceptual Topics

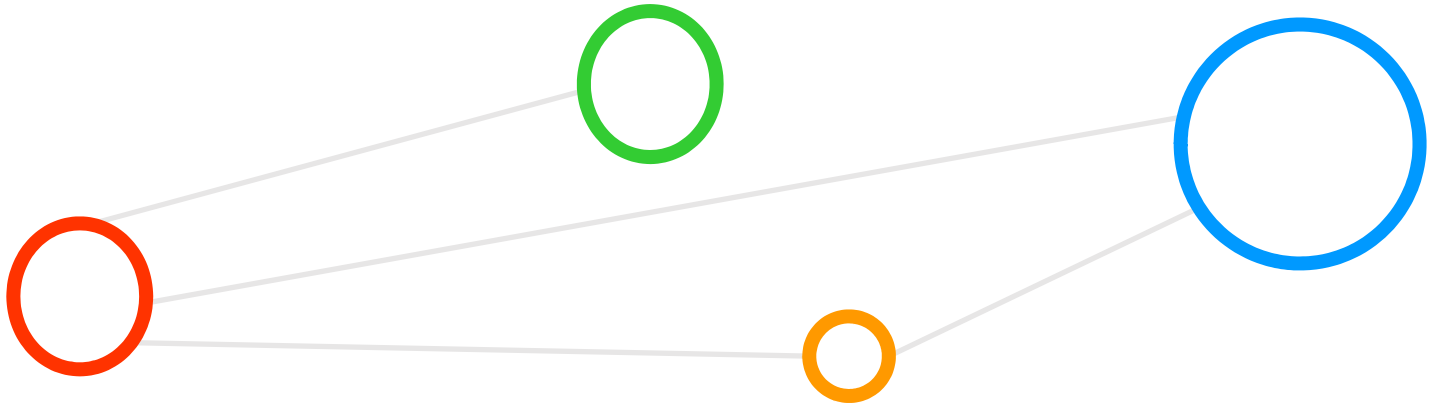
Outline

- Systems Biology
  - Motivation to Simulate Complex Systems & Basic Terminology
  - Scientific Case Protein Folding & Role of Specific Shapes
  - Role of Monte Carlo Methods & Repeated Random Sampling
  - SMMP Parallel Package Example to Simulate Proteins
  - Neuroscience Parallel Applications & NEST Parallel Code
- Bioinformatics
  - Motivation for Analysis of DNA Sequences & Terminology
  - Scientific Case Gene Sequencing & Role of Databases
  - Web Portals & BLAST Parallel Application Example
  - Drug Development using Scalable Computing Infrastructures
  - Human Brain Project Example for Challenges in Neuroscience

- **Promises from previous lecture(s):**
- ***Practical Lecture 0.2:*** **Lecture 12 & Lecture 13 provides more insights about selected applications in Terrestrial Sytems & some applications in Neuroscience**
- ***Lecture 5:*** **Lecture 13 will provide more details on using different & scalable parallel algorithms for systems biology & bioinformatics applications**
- ***Lecture 10:*** **Lecture 13 will provide more details on various systems biology & bioinformatics application codes that use parallel computing**
- ***Lecture 12:*** **Lecture 13 provides more details on general & so-called ensemble methods to estimate uncertainties that are often used in HPC**

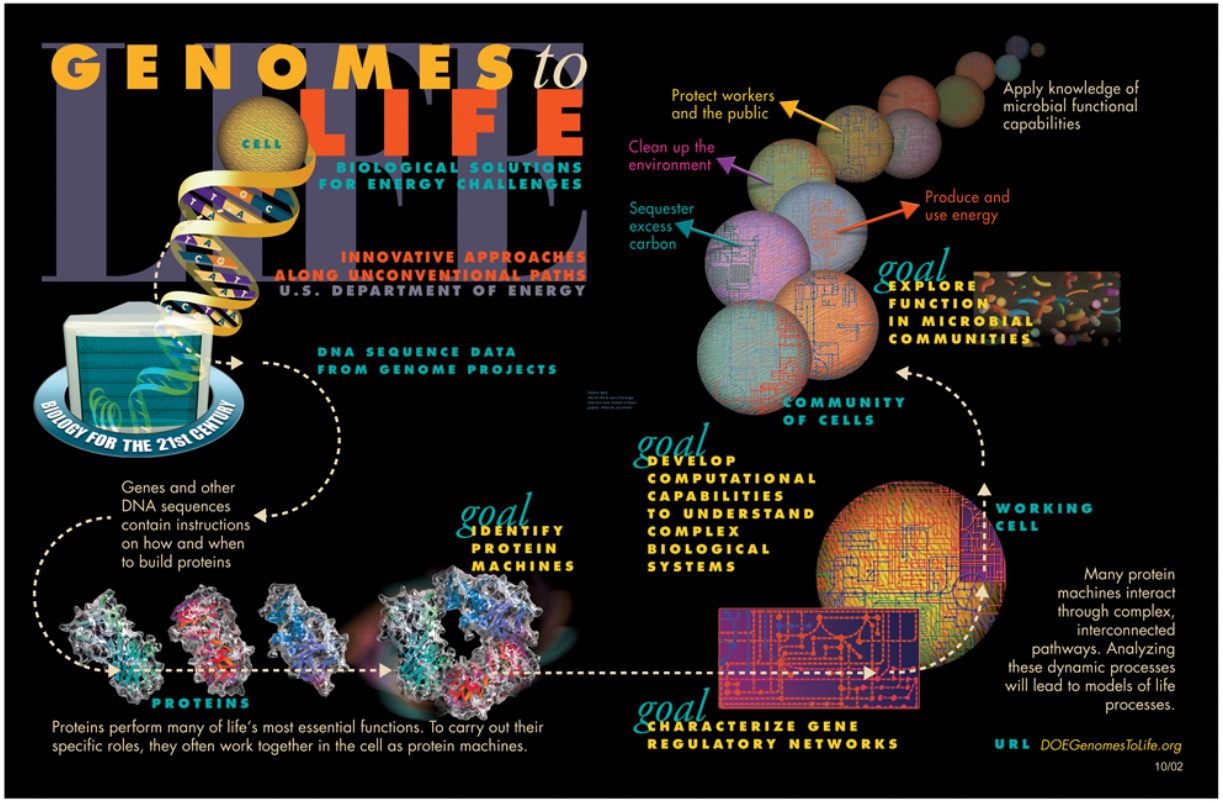
- **Note that this lecture is only a short lecture that usually needs a full course**
- **The goal is to understand selected HPC application fields & provide a few pointers to other advanced related university courses/topics/tutorials**

Systems Biology



Systems Biology – Motivation

- Focus
  - Study, model, and simulate a ‘complex biological system‘
  - E.g. understand and simulate small ‘Protein machines‘
  - E.g. understand and simulate models of life processes in cells (‘protein interactions pathways & joint work‘)



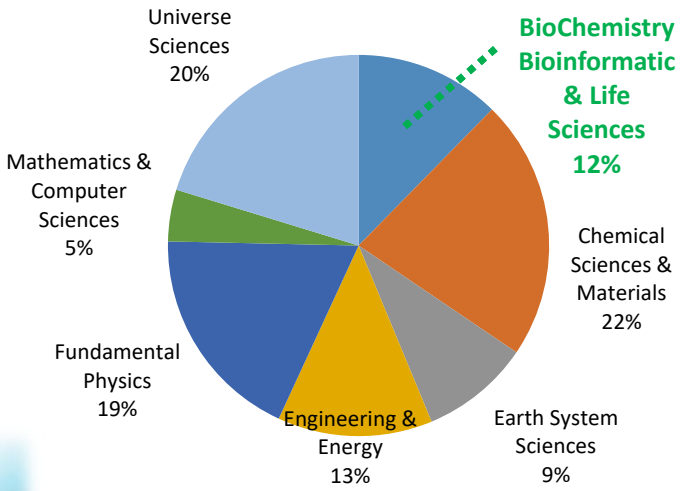
- **Systems biology is the computational and mathematical modeling of complex biological systems**
- **Modelling complex interactions within complex biological systems require powerful HPC systems**

modified from [6] Wikipedia on Systems Biology

Systems Biology – Complex Domain-Specific Terminology

- Selected terms (overlaps with many other domain-specific terminology)
  - Atoms, molecules, peptides, cells, tissue, energy fields, proteins, etc.
- Selected fields
  - Life sciences (broad), biochemistry, bioinformatics, biophysics, etc.
- Selected tasks
  - Analysis, sampling, monte carlo
- Importantance in HPC
  - Focus on the 'task at hand' in parallelization:
    Not get carried away by terminology

- **Modelling a complex biological systems requires an understanding of the complex terminology**
- **Working closely together with domain scientists ensures correct understanding & modelling ideas**

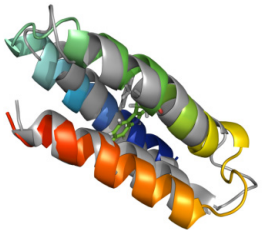


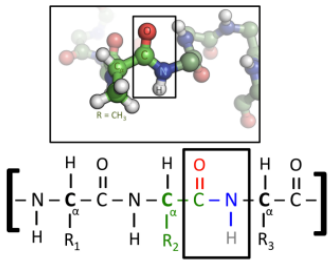
[7] PRACE Usage per community

Systems Biology – Domain-Specific Terminology – Proteins

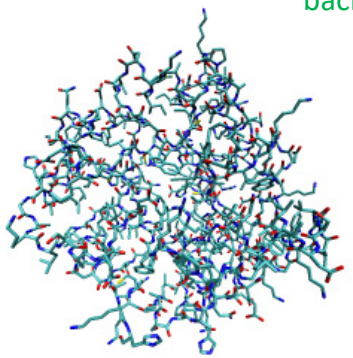
- Proteins
  - Large biological molecules that consist of one or more long chains of amino acid residues
  - Responsible for transporting molecules
  - Catalyzing and regulating biochemical reactions in the cell (or as antibodies for fighting infections)
  - E.g. 3D-structure of the 'protein triose-phosphate isomerase'

Function of a protein is closely related to its 3D shape

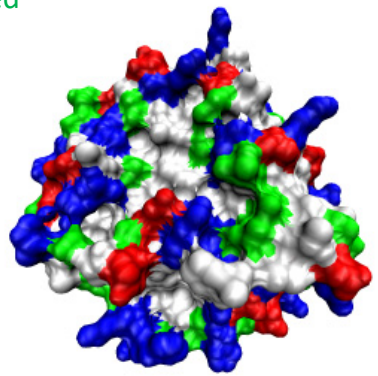




(all atom representation colored by atom type)



(simplified representation illustrating backbone conformation, colored by secondary structure)



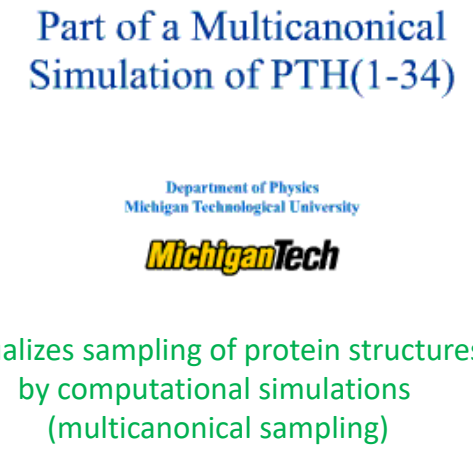
(Solvent-accessible surface representation colored by residue type – acidic residues red; basic residues blue, polar residues green, non-polar residues white)

(once linked in the protein chain, an individual amino acid is called a residue)

[8] Wikipedia on Protein

Systems Biology – Scientific Case Protein Folding (1)

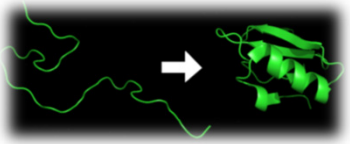
- Understanding ‘physics of the proteins‘
  - Only functional if they assume specific shapes
  - Important to explore how these structures emerge from a protein's chemical composition (sequence of amino acids as specified in the genome)
- Scientific impact
  - Lead to the rational design of novel drugs
  - Understanding of diseases that are caused by misfolding of proteins (e.g. protein mis-fold and aggregate conditions → neurological diseases)
  - Research on solutions for neurological diseases (e.g. Alzheimers)
- Computational simulations
  - Explore ‘sequence-structure relation’ – very complex problem
  - Extremely difficult for realistic protein models with high dimensionality and competing interactions



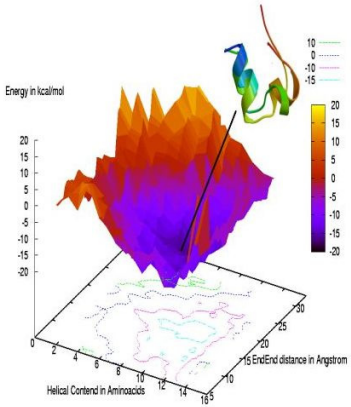
Visualizes sampling of protein structures by computational simulations (multicanonical sampling)

Systems Biology – Scientific Case Protein Folding (2)

- Need for ‘computational biophysical chemistry‘
  - Understand for a given protein the relation between its sequence of amino acids and the set of thermally accessible conformations
  - Comprehend the mechanism by which the protein folds into its native structure
- Complex systems (Protein folding is computationally hard)
  - All-atom models lead rough energy landscape
  - Probing the mechanism of folding in small proteins
  - Explore conditions under which proteins mis-fold and aggregate
  - Analysis of the energy landscape of molecules
  - Time intensive calculation of energy function → Potential for parallelization



[9] Protein Folding



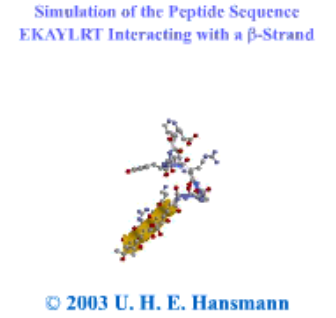
[10] A. Schug et al.

- **Understanding functions of proteins is a biological case of complex systems requiring HPC systems**

Systems Biology – SMMP Package Example (1)

- Approach
  - Time intensive calculation of energy function: parallelized calculation of the energy
  - Targets complexity through the development of numerical methods and techniques
  - Energy of a protein can be calculated by exploiting different force fields
  - Use various force fields to calculate the energy of a protein and protein-protein interactions
- Selected Features
  - Open source software package (some parts use MPI)
  - Contains various modern Monte Carlo algorithms
  - Contains more than 60 functions and subroutines

[11] Eisenmenger & Hansmann et al.



Structure of peptide (strand, alpha-helix, etc.) EKAYLRT depends on its environment

- **Simple Molecular Mechanics for Proteins (SMMP) is a parallel software tool for protein simulations**

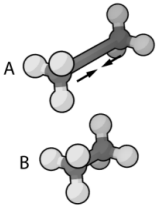
Systems Biology – SMMP Package Example (2)

- Example: Parallelization of ECEPP/3 force field *[12] J. Meinke et al.*
  - i and j are indices of atoms; r_{ij} is the distance between atom i and j,
  - L is the index of a dihedral angle in the protein chain.

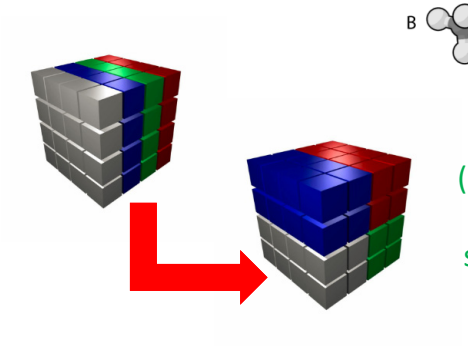
$$E_{\mathrm{ECEPP/3}} = \sum_{(i,j)} \frac{332 q_i q_j}{\epsilon r_{ij}} + \sum_{(i,j)} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \right) + \sum_{(i,j)} \left(\frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}} \right) + \sum_{l} U_l (1 \pm \cos(n_l \xi_l)) \, ,$$

Energy function equation that needs to be calculated

(used to describe the potential energy of a system of particles (e.g. molecules and atoms)



[13] Wikipedia on 'force fields'

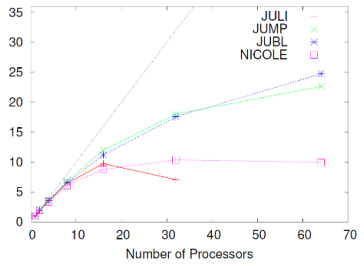


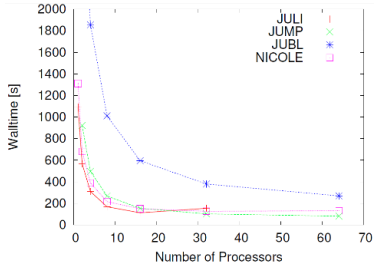
(default arrangement for 4x4x4 communicator)

(cubic arrangement of communicator scale much better)

(Monte Carlo simulation of Top7 protein – 50 sweeps)

- Selected parallelization approaches
  - Optimize MPI communicators for the energy calculation with a cubic arrangement
  - Use of sweeps of a Monte Carlo simulation (e.g. of the designed protein Top7, a 92 residue protein with 1477 atoms)





(maximum speedup 25x on JUBL – BlueGene/L System)

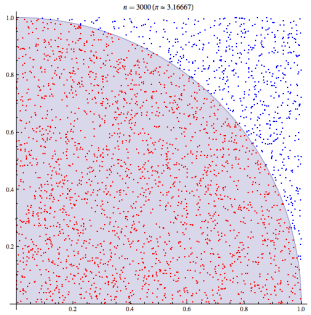
(significant reduction of used walltime)

Systems Biology – Monte Carlo Methods – Approach

- Method applicability
  - Useful when it is difficult or impossible to apply a deterministic algorithm
  - Define a domain of possible inputs
  - Generate inputs randomly from a probability distribution over the domain
  - Perform a deterministic computation on the inputs and aggregate the results

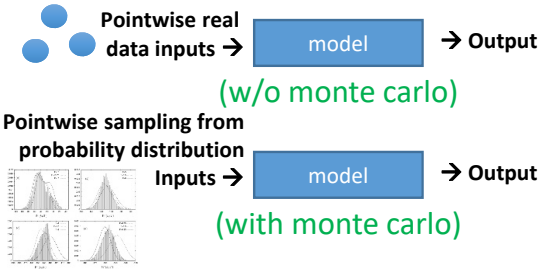
(origin related to techniques in playing & recording results in gambling casinos)

(also known as MC ensemble method)



(Example of approximating the value of pi – After placing many (30000) random sampling points, the estimate for pi is within 0.07% of the value – happens with an approximate probability of 20%)

- **Monte Carlo (MC) methods rely on repeated random sampling to obtain numerical results**
- **MC run simulations many times over to obtain the distribution of an unknown probabilistic entity**



(count the number of points inside the circle and the total number of points)

(the ratio of the two counts is an estimate of the two areas ~ pi/4)

(estimate of the two areas gives ~pi/4, so multiply with 4 to estimate pi)

[14] Wikipedia on Monte Carlo method

Systems Biology – Monte Carlo Methods – MPI Example

```
#include <math.h>
#include <mpi.h>
#include <gsl/sdl_rng.h>
#include "gsl-spring.h"

int main( int argc; char *argv[]) {
  int i, k, N;
  double u, ksum, Nsum, gsl_rng *r;
  MPI_Init(&argc,&argv);
  MPI_Comm_size(MPI_COMM_WORLD, &N);
  MPI_Comm_rank(MPI_COMM_WORLD, &k);

  r = gsl_rng_alloc(gsl_rng_spring20);

  for (i=0; i<10000; i++) {
    u = gsl_rng_uniform(r);
    ksum += exp(-u*u);
  }

  MPI_Reduce(&ksum, &Nsum, 1, MPI_DOUBLE,
          MPI_SUM, 0, MPI_COMM_WORLD);
  printf("MC estimate is %f\n", (Nsum/10000/N));
  MPI_Finalize();
  return 0;
}
```

modified from [15] MSMC

'simplified demo code'

- Include library for a parallel random number generator (PRNG) – Important part of MC!
- Initializes random number generator (with a specific type)
- Use 10000 variates on each processor to create 'local sum'
- Gsl_rng_uniform(r) function returns a double precision floating point number uniformly distributed in the range [0,1) (idea of probability distribution)
- Use MPI_Reduce to create 'global sum' and printout estimate

(MPI code prints out estimate of integral)

$$I = \int_0^1 \exp(-u^2) du$$

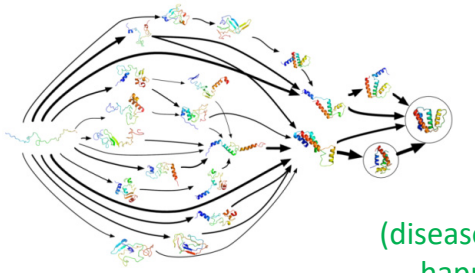
Systems Biology – Monte Carlo Methods – Example (1)

- Scientific case: Understanding protein folding & pathways
  - Proteins perform functions within living organisms (e.g. respond to stimuli)
  - Proteins differ in their sequence of amino acids, results in different foldings
  - Correct and unique 3D structure is essential to the functions of proteins

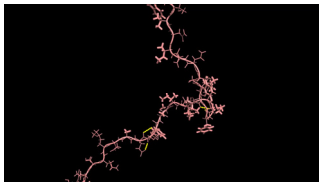
(Research 'folds' to find out what the protein does and how it does it)

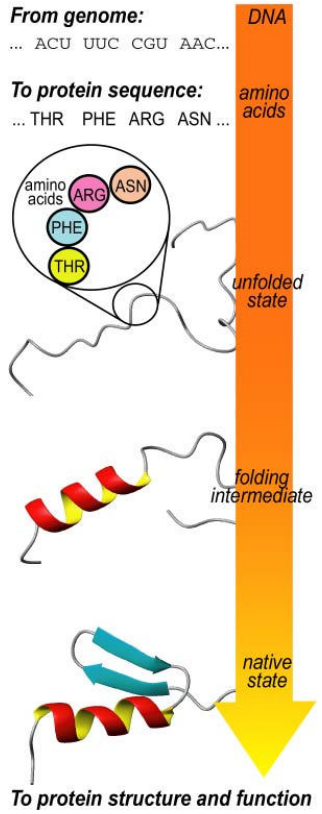
- Process of protein folding as a parallel computing application
  - Using Monte Carlo simulations
  - Simulations that use stochastic methods
  - to generate new configurations of a system
  - Initial conditions of particles, then Monte Carlo 'moves' that changes configurationof particles (if acceptance criteria is fulfilled)

(out of the astonomical number of possible ways to fold, a protein can pick one fold in microseconds)



(diseases may happen)





[17] S. Mohanty et al, JSC Simlab Biology *[16] Folding@Home*

- **The Monte Carlo 'method' is used in many parallel computing applications, e.g. also in finance**

Systems Biology – Parallel Neuroscience Application Example – Revisited

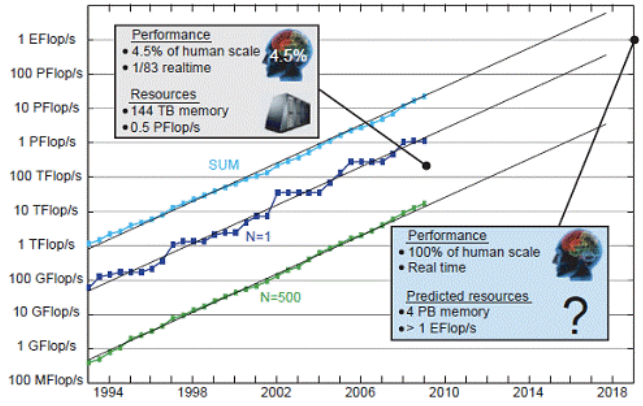
- Scientific case: understanding the function of the human brain
- Neuron/NEST code:
  - Parallel application codes to simulate biologically realistic neural networks (neurons + synapses)

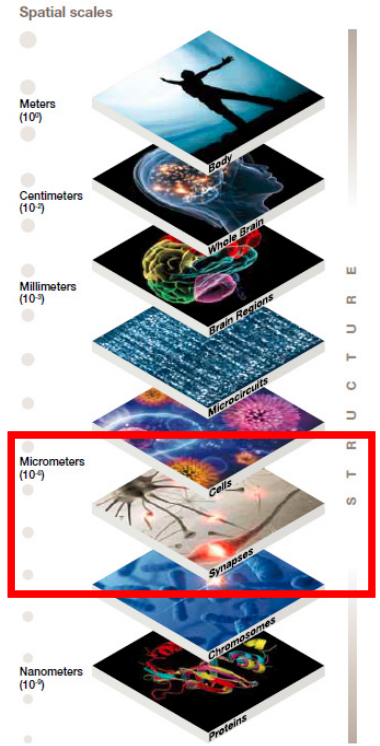
(the biggest supercomputers today just reach ~4.5% of human scale)

- Simulate models of the brain at different levels
  - Different 'granularity': Molecular, cellular, network level
  - Simulated brain will reach up to ~900 TB (Big Data!)

Simulated ~2 billion neurons

- 1 second biological time
- 40 minutes compute time (on K supercomputer)





[18] HBP Project

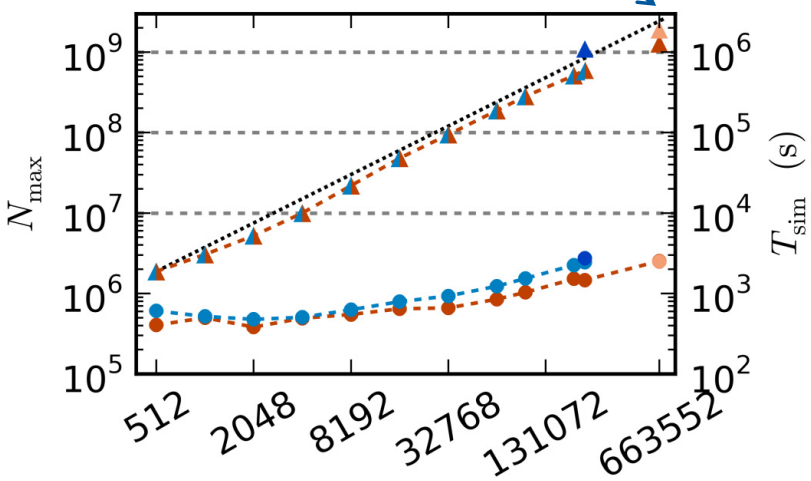
Systems Biology – Parallel Neuroscience Application Example – Revisited

- Simulations of spiking – parallel neural network models
  - Use parallelization (e.g. MPI cf. Lecture 3 and hybrid programming)

Human Brain Project

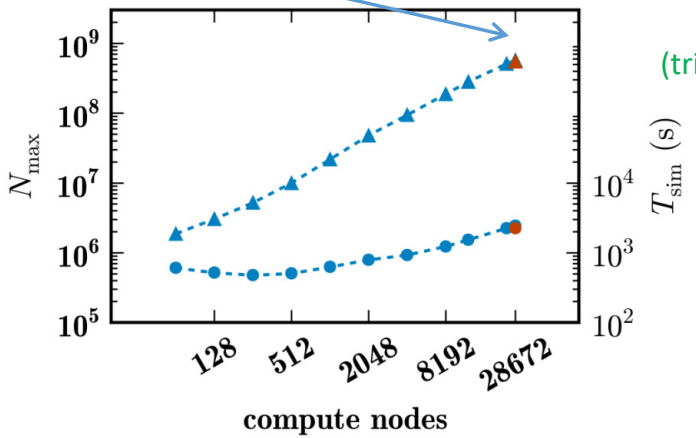
nest::

Largest spiking neural network simulation to date:
$1.86 \cdot 10^9$ neurons, $11.1 \cdot 10^{12}$ synapses



(11.250 synapses/neuron – light blue JUQUEEN, red: K computer)

550 Mio. Neurons, $5.5 \cdot 10^{12}$ synapses on 458752 cores

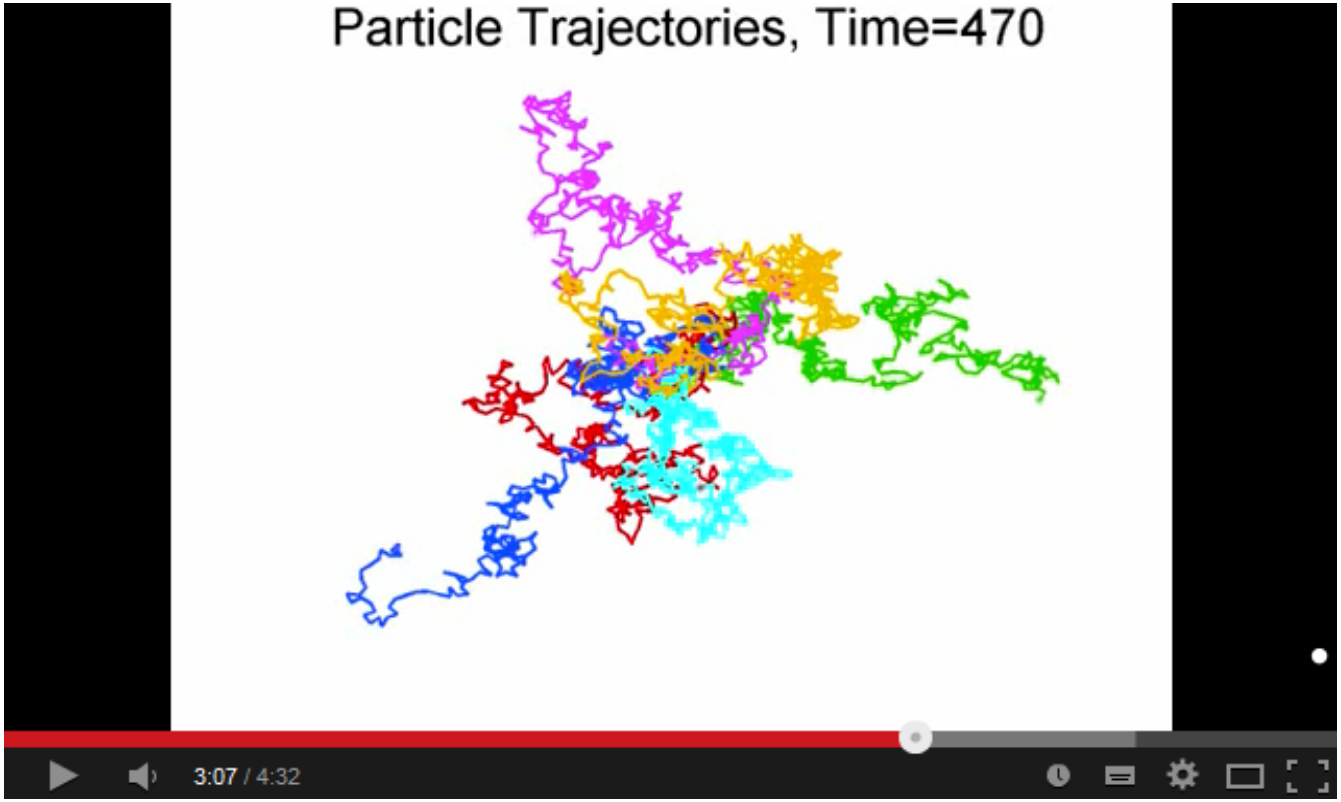


(blue: 8 threads, red: 16 threads per node)

(triangles: maximum network size, dots: simulation time)

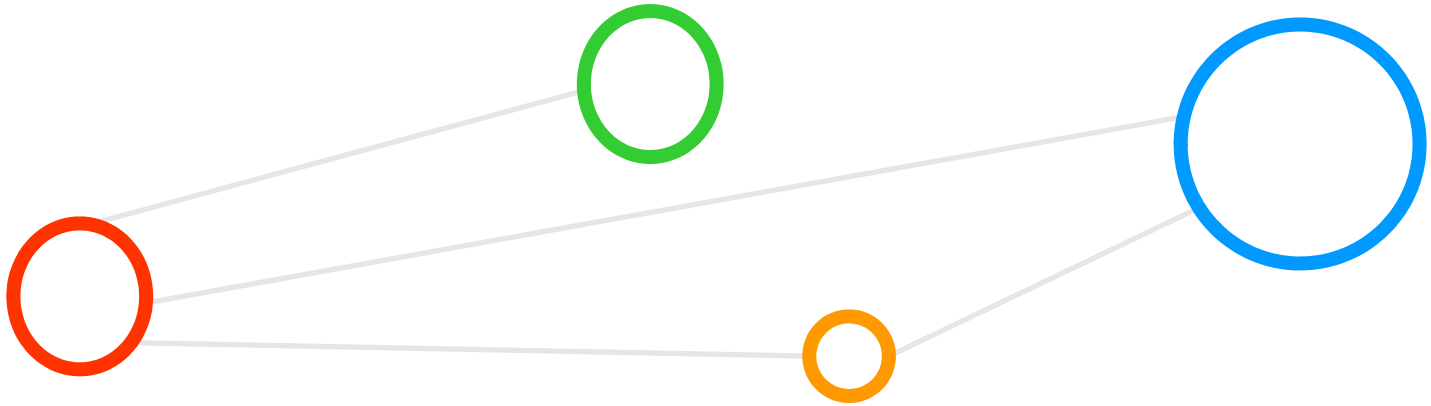
[18] HBP Project

[Video] Systems Biology – Important Monte Carlo Methods



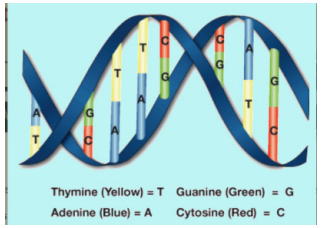
[19] YouTube Video, Monte Carlo Simulation

Bioinformatics



Bioinformatics – Motivation & BLAST Application Example

- Focus
  - Study and process biological datasets (cf. Systems Biology focus was ‘model complex systems‘)
  - Develop user-friendly tools & technologies
  - E.g. analysis of ‘DNA sequences‘
- Relationship to Systems Biology
  - Closely related with massive overlaps
  - Tools & techniques developed & exchanged
  - Datasets used in both fields from real biological systems
- BLAST Application Example
  - Scientific case: Infer functional and evolutionary relationships between gene sequences
  - General approach: Finds regions of local similarity between gene sequences
  - Two step method:
    1. Compare gene sequences to sequence databases (DBs)
    2. Calculate the statistical significance of matches



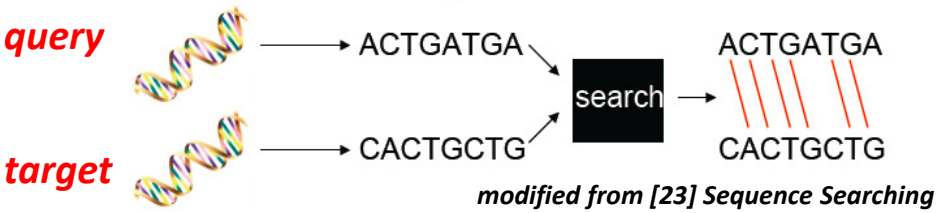
- **Bioinformatics develops methods & software tools for understanding biological datasets**
- **Interdisciplinary field combining computer science, statistics, mathematics and engineering**

- **The Basic Local Alignment Search Tool (BLAST) is one of the most known tools in comp. biology**
- **BLAST two step method: (1) Compares gene sequences to DBs (2) Calculates match significance**

modified from [22] Wikipedia on Bioinformatics ***[21] DNA sequences***

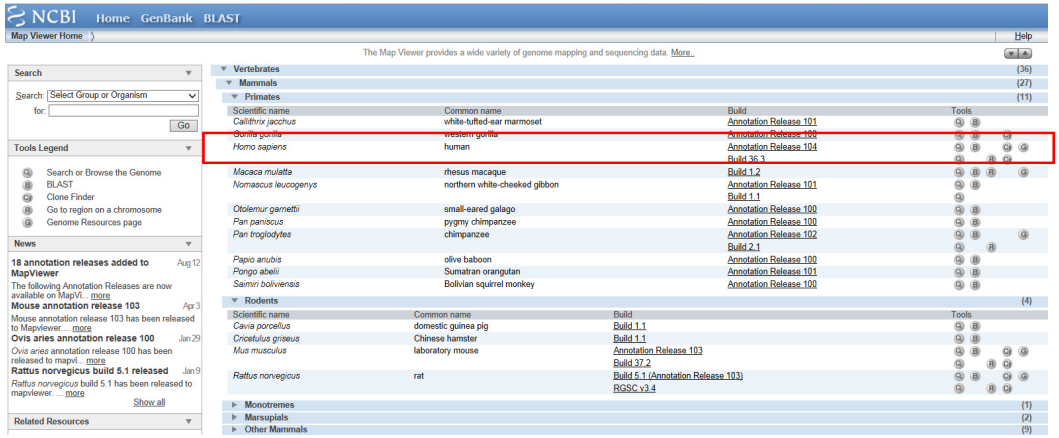
Blast Application in Computational Biology – Comparing Sequences

- Comparing sequences
  - Heuristic method → approximate solutions, exact is too compute-intensive
  - Finds similar sequences not by comparing either sequence in its entirety
  - Locates short matches between the two sequences to find similar ones
- Algorithm (many optimizations and variations exist)
  - Input A: Query sequence to search for
  - Input B: Sequence(s) to search against (target sequence)
    (Use of sequence DB here enables comparison with n sequences)
  - Output: Found sub-sequences in target/DB similar to subsequences in query



modified from [23] Sequence Searching

[24] NCBI GenBank Genomic BLAST databases

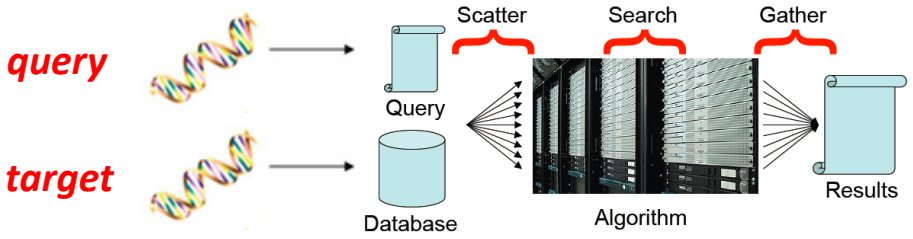


Blast Application in Computational Biology – Parallelization Approach

- Parallelization
  1. Apply a good parallelization method (e.g. domain decomposition or general parallelization approach)
  2. Write 'manually' good MPI (or OpenMP) code for (technical) communication between processors (e.g. across 1024 cores)
     - In this case the MPI implementation exists already: mpiBLAST
     - (practice: several different parallel implementations of BLAST exist)
- mpiBLAST as 'parallel computing application'
  - Open Source, parallel MPI implementation of BLAST

[25] MPI Blast

Master-worker scheme (cf. Lecture 3)



- **As the searches are independent from each other we can use parallel libraries to split the whole 'big search query' into smaller independent 'search tasks' computed in parallel**

- **As in many cases within parallel application development, commonly needed libraries or implementations of algorithms are already existing and should be re-used where possible**

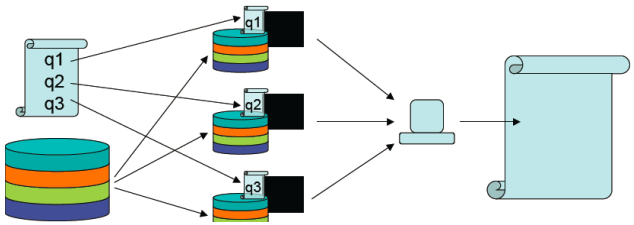
```
int main( int argc, char* argv[] ) {
  …
  MPI_Init(&argc, &argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
  MPI_Comm_size(MPI_COMM_WORLD, &node_count);
  …
  MPI_Barrier(MPI_COMM_WORLD);
  …
  MPI_Finalize();
  …
}
```

mpiblast.cpp

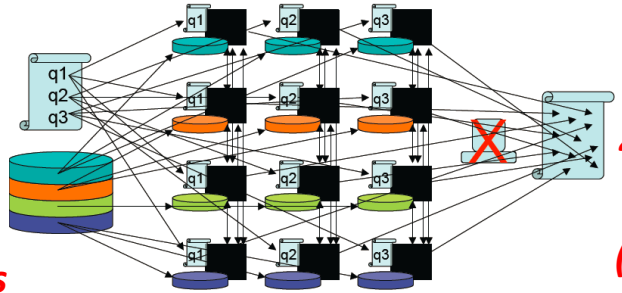
Blast Application in Computational Biology – Parallel I/O Approach

- I/O Approach
  - Integrate well technical code with 'problem-domain code' (e.g. target database data & data distribution)

'Distribute DB & Query fragments'



Knowing…
… system architecture
… parallel file system
… available memory
… MPI instead of files
… using communicators

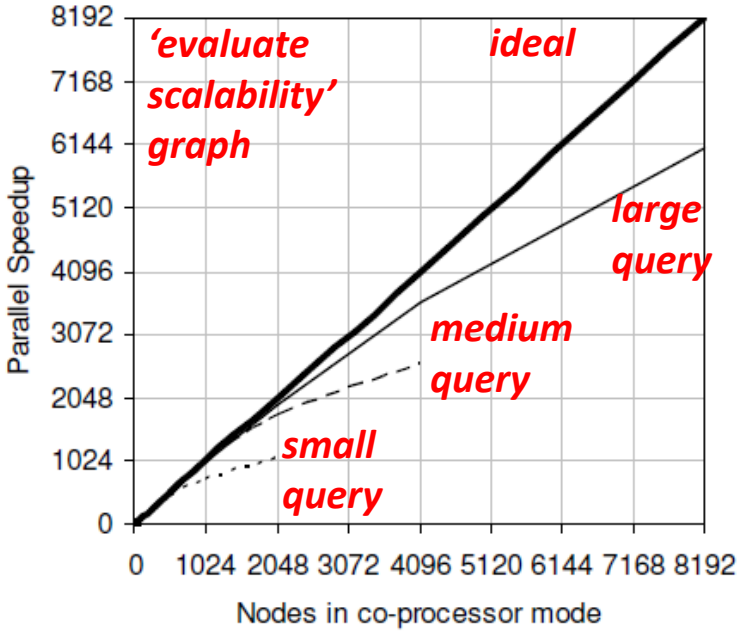


'not well integrated' → scalability limits (master bottleneck, IO, load imbalancing,…)

'optimizations' (e.g. think about I/O & network)

'well integrated' → truly scalable (parallel IO, multiple masters)

- **Achieving a speedup of 'processing some problem space data sets' is one of the major motivations of creating parallel computing applications that users can really experience**



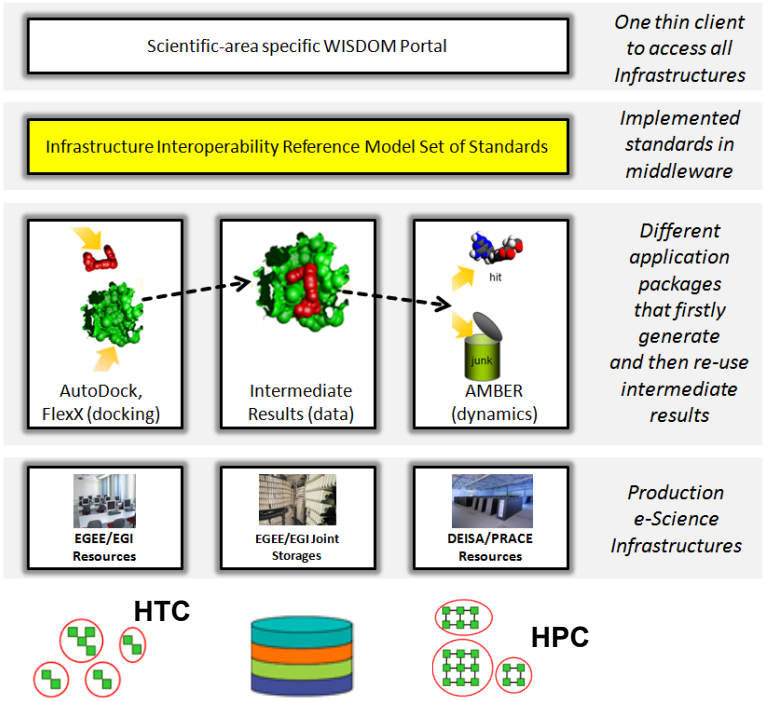
- **Optimizations in various forms and using a wide variety of different approaches can lead to significant improvements in speed-up of the parallel computing application**

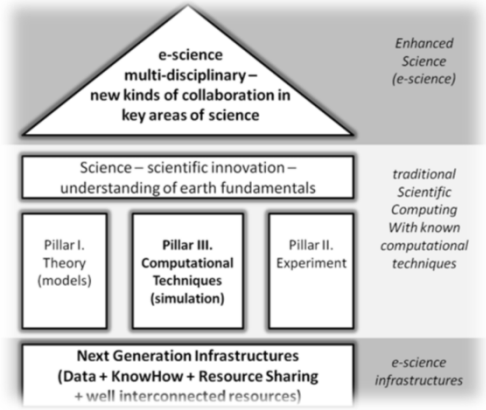
modified from [26] O. Thorsen et al., Parallel Genome Sequence Search

[23] Sequence Searching

Bioinformatics – Drug Development using Scalable Computing Infrastructures

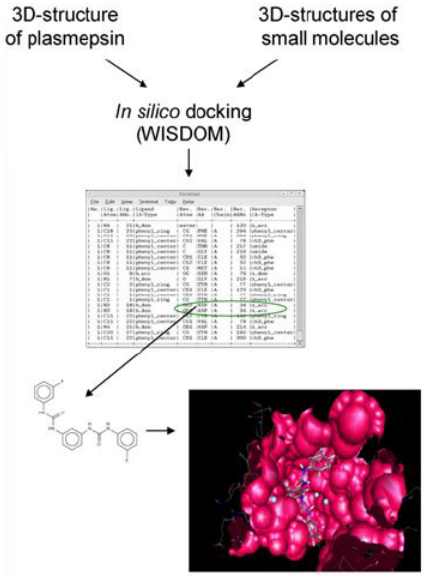
- Use of Interoperability of two different computing paradigms
  - E.g. use of HTC in EGI & HPC in PRACE infrastructures (cf. Short Lecture 11)





[27] M. Riedel et al., 'Research Advances by using interoperable e-Science Infrastructures', 2009

(Web-based portal make complex infrastructure access easy for biological domain scientists)

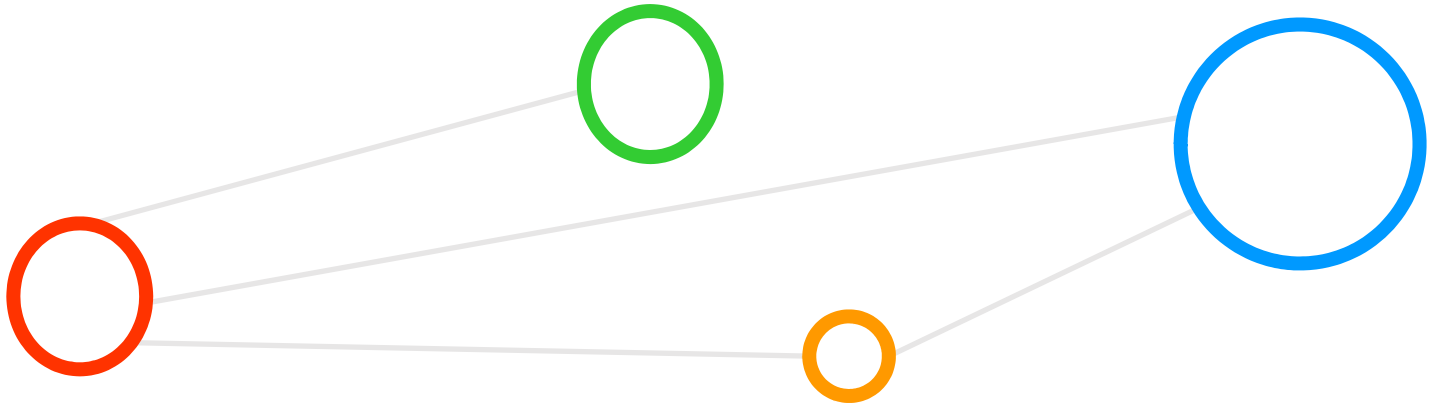


➢ **Lecture 14 will provide more pieces of information about molecular systems & libraries as well as selected parallel application methods**

[Video] Systems Biology & Bioinformatics in Neuroscience

[20] YouTube Video, Human Brain Project

Lecture Bibliography



Lecture Bibliography (1)

- [1] Terrestrial Systems Simulation Lab, Online: http://www.hpsc-terrsys.de/simlab
- [2] F. Gasper et al.: Scaling of the fully coupled Terrestrial Systems Modeling Platform on BlueGene/Q, Online: http://www.geosci-model-dev.net/7/2531/2014/gmd-7-2531-2014.pdf
- [3] Wikipedia on 'Numerical Weather Prediction', Online: http://en.wikipedia.org/wiki/Numerical_weather_prediction
- [4] John Michalakes et al., 'WRF Software: Code and Parallel Computing', Online: http://www2.mmm.ucar.edu/wrf/users/tutorial/201301/gill_software.pdf
- [5] Wikipedia on 'WRF', Online: http://en.wikipedia.org/wiki/Weather_Research_and_Forecasting_Model
- [6] Wikipedia on 'Systems Biology', Online: http://en.wikipedia.org/wiki/Systems_biology
- [7] PRACE, Online: http://www.prace-ri.eu/
- [8] Wikipedia on 'Protein', Online: http://en.wikipedia.org/wiki/Protein
- [9] Wikipedia on Protein Folding, Online: http://en.wikipedia.org/wiki/Protein_folding
- [10] A. Schug, W. Wenzel and U.H.E. Hansmann, J. Chem. Phys., 122 (2005) 194711
- [11] F. Eisenmenger, U.H.E. Hansmann, S. Hayryan and C.K. Hu, 'SMMP - A Modern Package for Protein Simulations, Comp.Phys. Comm. 138 (2001) 192'
- [12] Jan Meinke et al., 'Parallelization of ECEPP/3 in SMMP', Online: http://juser.fz-juelich.de/record/59934/files/FZJ-2014-02488.pdf
- [13] Wikipedia on 'Forcefields', Online: http://en.wikipedia.org/wiki/Force_field_(chemistry)
- [14] Wikipedia on Monte Carlo, Online: http://en.wikipedia.org/wiki/Monte_Carlo_method

Lecture Bibliography (2)

- [15] Getting started with MCMC, Online: http://darrenjw.wordpress.com/2010/12/14/getting-started-with-parallel-mcmc/
- [16] Folding@Home, Online: http://folding.stanford.edu/
- [17] JSC Simulation Lab Biology, Online: http://www.fz-juelich.de/ias/jsc/EN/AboutUs/Organisation/ComputationalScience/Simlabs/slbio/_node.html
- [18] Human Brain Project, Online: https://www.humanbrainproject.eu/de
- [19] YouTube Video, 'Episode 2: Monte Carlo Simulation', Online: http://www.youtube.com/watch?v=Xaymy3Blnq4
- [20] YouTube Video, ‚Human Brain Project', Online: http://www.youtube.com/watch?v=JqMpGrM5ECo
- [21] Graphic Representation of DNA Sequences, DNA Genealogy, Online: http://dna-footprints.com/89/abcs-dna-snps/
- [22] Wikipedia on 'Bioinformatics', Online: http://en.wikipedia.org/wiki/Bioinformatics
- [23] J. Archuleta et al., 'A Maintainable Software Architecture for Fast and Modular Bioinformatics Sequence Search, Sequence Searching, 2007
- [24] NCBI GenBank, Nucleotide Blast, Online: http://blast.ncbi.nlm.nih.gov/Blast.cgi
- [25] MPIBlast, Online: http://www.mpiblast.org/
- [26] Oystein Thorsenet al., "Parallel Genomic Sequence-Search on a Massively Parallel System", 4th international conference on Computing frontiers, Ischia, Italy, 59 - 68 (2007)
- [27] M. Riedel et al., 'Research Advances by using Interoperable e-Science Infrastructures', Journal of Cluster Computing, 12(4):357–372, 2009, Online: https://www.researchgate.net/publication/220405901_Research_advances_by_using_interoperable_e-science_infrastructures


services
measurement
within
fusion
new
funding
Policy-based
concepts
device
Analysis
forms
cross-disciplinary
resources
Centre
climate
Computer
expertise
Approaches
construction
Fusion
year
like
Computational
Test
E.g
Computing
disciplines
results
Enable
Cross-Disciplinary
methods
High
big
nodes
PB
community
Environment
Modelling
different
EU
Science
centers
Key
scientific
important
files
model
international
etc
DKRZ
Use
storage
basis
Infrastructure
simulations
ENES
technologies
SMAQ
engineering
References
formats
brain
increasing
via
EUDAT
physical
IPCC
Big
Project
Data
processing
Common
TB
MapReduce
create
computing
Methods
Using
European
databases
GB
Supercomputing
Work
images
codes
analysis
stored
Resources
analyze
Large
Summary
data-intensive
Service
Scientific
levels
often
DLCL
Level
Research
external
use
Matters
Concrete
performance
JSC
computational
research
e.g
science
HPC
Juelich
datasets
acquisition
system
compute
Landscape
management
System
Climate
modelling
AR5
Hardware
distributed
large-scale
access
hundreds
Nutshell
Services
cores
Structure
current
network
DLCLs
Understanding
structures
Technologies
Earth
Germany
Simulation
Processing
directory
project
General
Provide
NASA
Energy
systems
day
Health