Parallel Programming
(Spring 2015, Prof.dr. H. Wijshoff)

- Four parts:
  - Introduction to Parallel Programming and Parallel Architectures (partly based on slides from Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar accompanying ``Introduction to Parallel Computing'', Addison Wesley, 2003.)
  - Parallel Algorithms (mix slides and black/white board)
    - Parallel Algorithm Design
    - Parallel Numerical Computing
    - Parallel Graph Computing
    - Parallel Sorting
  - Existing Programming Paradigms (mix slides and black/white board)
  - New Programming Paradigms (black/white board)

Lab/Homework/Assignments

During the course of the semester, a choice of programming assignments and/or (theoretical) algorithmic problems will be offered.

Choice for one or the other will be left to the student.

These assignments make for 40% of the total load.

Open “book” final exam will make up for remainder 60%.

A Long History

- The advent of parallel computing dates back to the fifties of the last century
  - IBM introduced the 704 (full parallel floating point arithmetic) in 1954, through a project in which Gene Gene Amdahl was one of the principal architects.
  - In April 1958, S. Gill (Ferranti) discussed parallel programming and the need for branching and waiting.
  - Also in 1958, IBM researchers John Cocke and Daniel Slotnick discussed the use of parallelism in numerical calculations for the first time.
- In 1969, US company Honeywelll introduced its first Multics system, a symmetric multiprocessor system capable of running up to eight processors in parallel.
- The ILLIAC IV (1971) was one of the first attempts to build a massively parallel computer. One of a series of research machines (the ILLIACs from the University of Illinois), the ILLIAC IV design featured fairly high parallelism with up to 256 processors.

Milestones

1972: First Supercomputer: CRAY 1

1 MFLOP = 1 000 000 operaties/sec.

1989: CRAY YMP

1 GFLOP = 1 000 000 000 operaties/sec.

1996: ASCI red (Intel based parallel processor)

1 TFLOP = 1 000 000 000 000 oper./sec.

2008: IBM Roadrunner

1 PFLOP = 1 000 000 000 000 000 oper./sec.

At this moment (nov 2014):

NUDT (China): 54.9 PFLOP (0.054 EXAFLOP)

54 900 000 000 000 000 oper./sec. achieved by
3 120 000 cores using up to 18 MW

RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
1	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
2	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945
6	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Cray Inc.	115,984	6,271.0	7,788.9	2,325
7	Texas Advanced Computing Center/Univ. of Texas United States	**Stampede** - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz, Infiniband FDR, Intel Xeon Phi SE10P Dell	462,462	5,168.1	8,520.1	4,510
8	Forschungszentrum Juelich (FZJ) Germany	**JUQUEEN** - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM	458,752	5,008.9	5,872.0	2,301

What does 1 PFLOP mean?

Multiplying 2 numbers with 15 decimals

- Paper and Pencil: 1 per 4 minutes
- Calculator: 10 per minute (based on 300 cpm (character per minute) (750 cpm world champion, Guinness))
- 1 GFLOP: 60 000 000 000 per minute

So 1 GFLOP is 8 faster than the whole world population with calculators

1 PFLOP is 1 000 000 faster yet!!!!!!!

WHY do we need to compute at these rates?

Exponential growth of computational complexity

(Easy) example: CHESS

- Assume an average of 10 possible moves per turn
- Average chess match: 80 turns

So 10^80 different possible outcomes

With 1 PFLOP: 10^65 sec =

4x10^57 years =

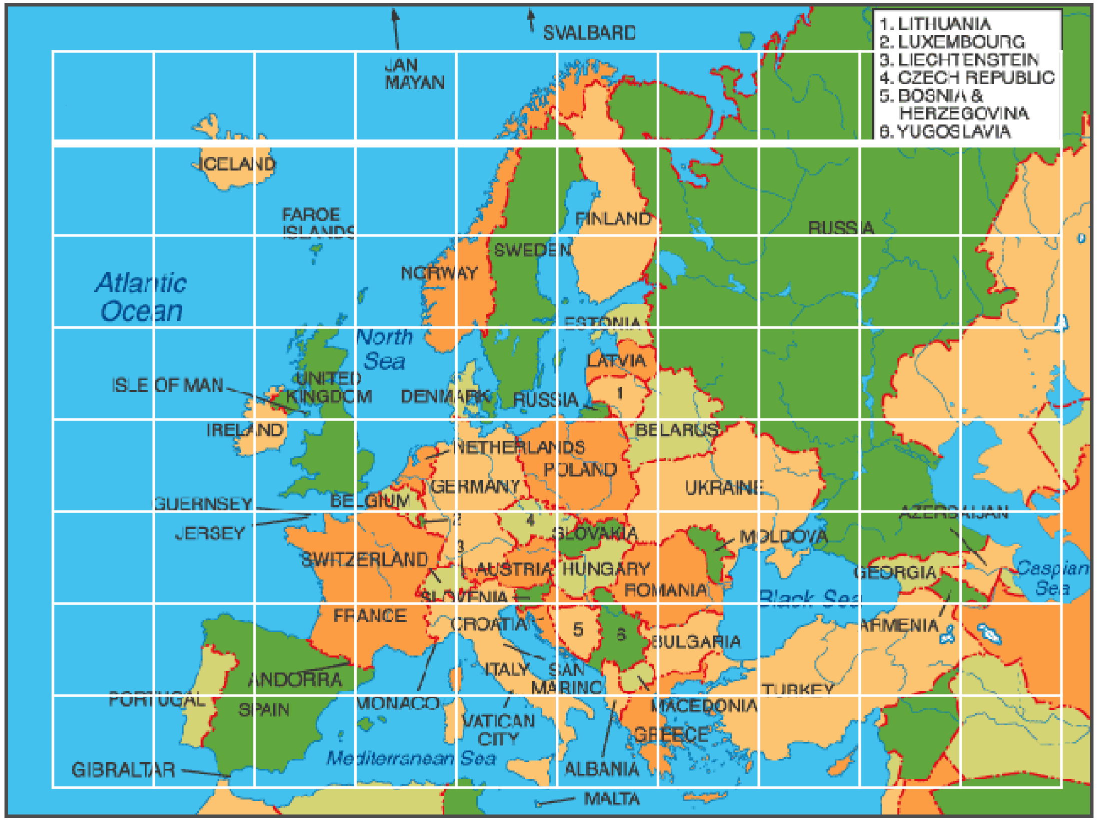
4x10^54 centuries =

10^48 x the existence of the universe

WHY (II)

Large Scale of computations

Example: Weather Forecasting



Computation (Simulation)

- For each grid point the interaction with its neighbor gridpoints are computed with respect to temperature, air pressure, moisture,
- Europe’s surface: 5 700 000 km^2
- Air heigth: 10 km
- With a 1m x 1m x 1m grid this results in:

  57 000 000 000 000 000 grid points

Computation (II)

- Several computations per grid point:
  Assume for each second and for 5 variables,
  then for a prediction of 12 hours:
  5 x 12 x 60 x 60 = 216 000 per grid point
- With a 1 PFLOP computer this takes:
  57 x 10^16 X 216 x 10^3 / 10^15=
  12 x 10^22 / 10^15 = 12 x 10^6 sec. =
  3333 hours = 138 days !!!!!!!!!

HPC Grand Challenges

"A Research and Development Strategy for High Performance Computing", Executive Office of the President, Office of Science and Technology Policy, November 20, 1987

- Prediction of weather, climate, and global change
- Challenges in materials sciences
- Semiconductor design
- Superconductivity
- Structural biology
- Design of pharmaceutical drugs
- Human genome
- Quantum chromodynamics
- Astronomy
- Challenges in Transportation
- Vehicle Signature
- Turbulence
- Vehicle dynamics
- Nuclear fusion
- Efficiency of combustion systems
- Enhanced oil and gas recovery
- Computational ocean sciences
- Speech
- Vision
- Undersea surveillance for anti-submarine warfare

Recently this list of applications was enlarged significantly

Next to applications in engineering and design, we have

- Scientific Applications: structural characterization of genes and proteins, new materials: understanding chemical pathways, bio-informatics and astrophysics, etc
- Commercial Applications: servers for large scale web servers (google, facebook, etc.), trading systems, etc.
- Applications in Computer Systems (the Internet itself): intrusion detection, cryptography, etc.
- Applications for social networks: online data mining,…
- Data Mining at large