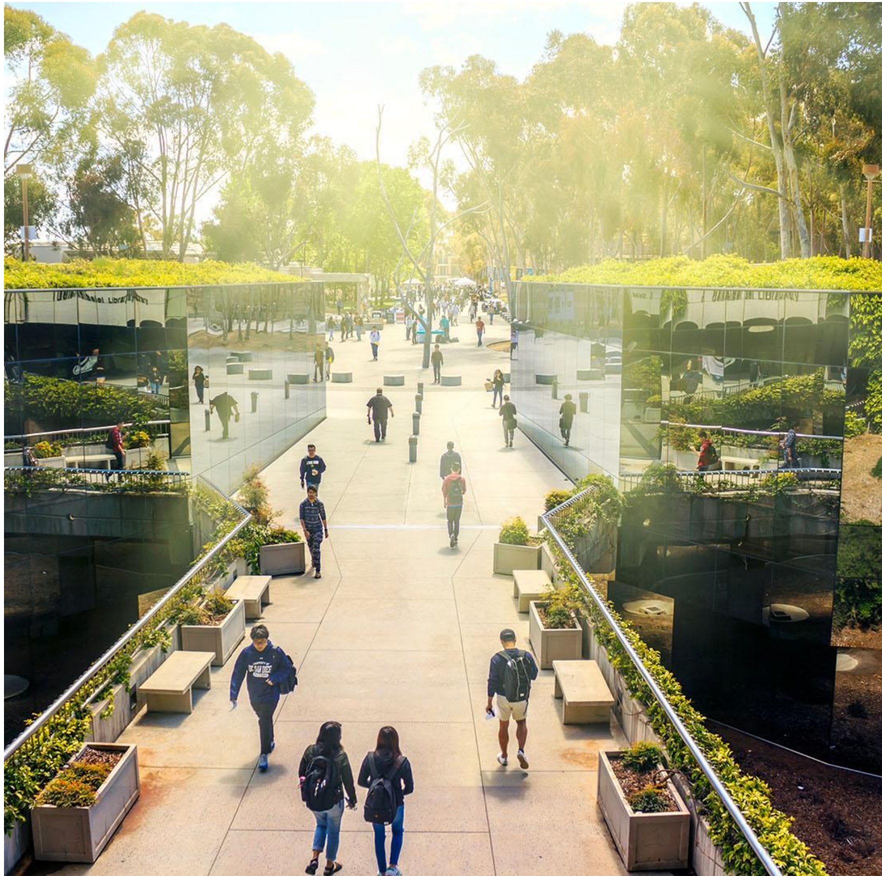
Supercomputing and Big Data - Industry Use Cases

Presenter: Miloš Ivanović mivanovic@kg.ac.rs

Institution: Faculty of Science, University of Kragujevac



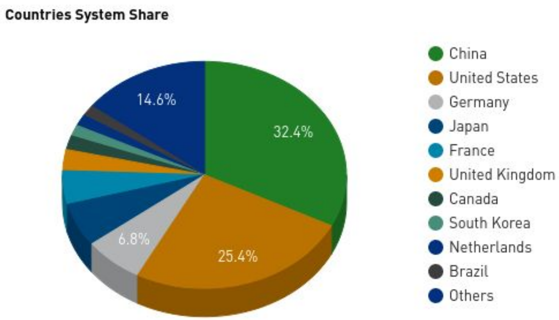
Contents

Supercomputing Dictionary

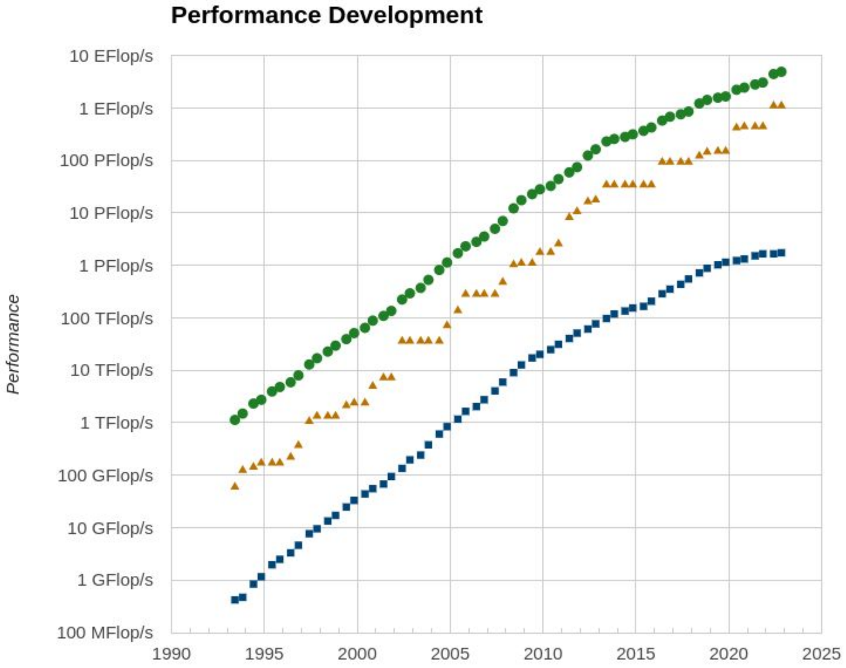
- **Supercomputing** is the biggest, fastest computing right this minute
- Likewise, a supercomputer is the **biggest, fastest computer right this minute**
- So, the definition of supercomputing is constantly changing
- **Rule of Thumb**: a supercomputer is 100 to 10,000 times as powerful as a PC
- **Jargon**: supercomputing is also called High Performance Computing (HPC)

Fastest supercomputers on Earth - https://top500.org/

- **Frontier - HPE Cray EX235a**
- AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11
- **Cores** - 8,730,112
- **Performance** - 1,102.00 PFlop/s
- **Power** - 21,100.00 kW



Performance over time

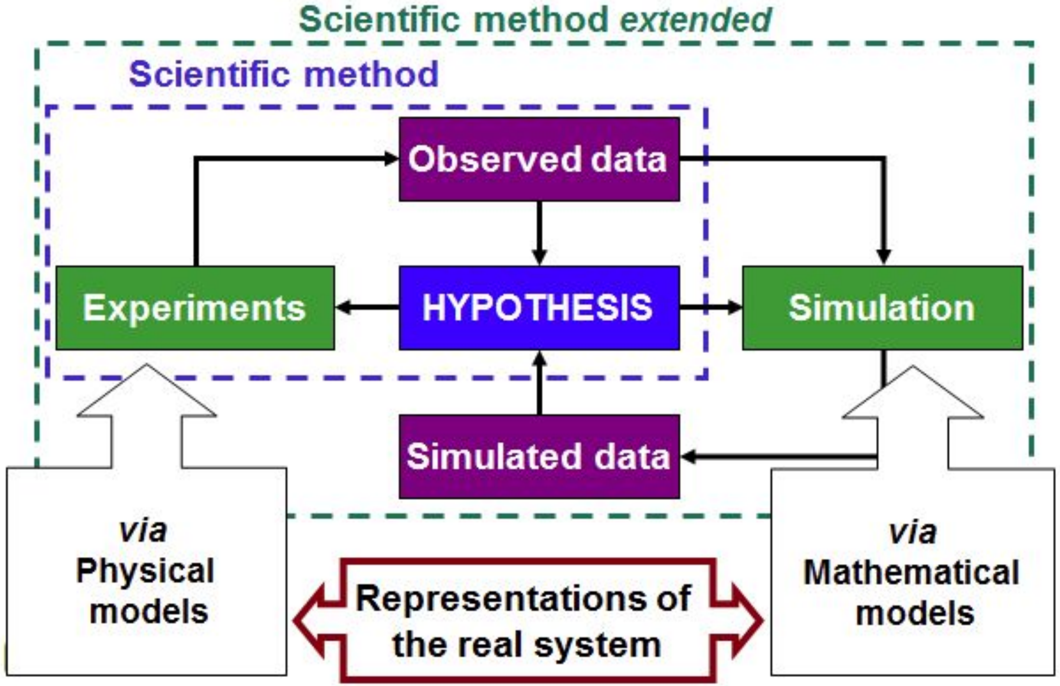


- **1997**: Deep Blue wins Kasparov
- **2011**: IBM Watson wins Jeopardy quiz
- **2016**: AlphaGo wins Go champion
- **2022:** GPT-3 model

Why do we need ever increasing performance?

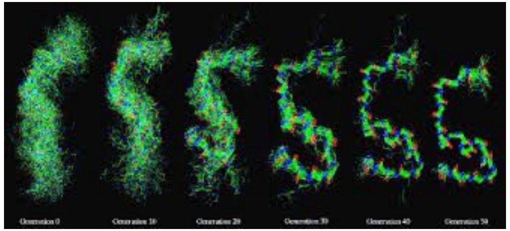
- **CASE 1 - Complete a time consuming application in less time**
  - Optimization of a new car design
  - Do more in less time
- **CASE 2 - Complete an operation under a tight deadline**
  - Weather forecast
  - High-throughput sensors
- **CASE 3 - Perform high number of operations per second**
  - High traffic website
- **CASE 4 - Problem doesn't fit the memory of a single PC**
  - The majority of scientific problems of interest

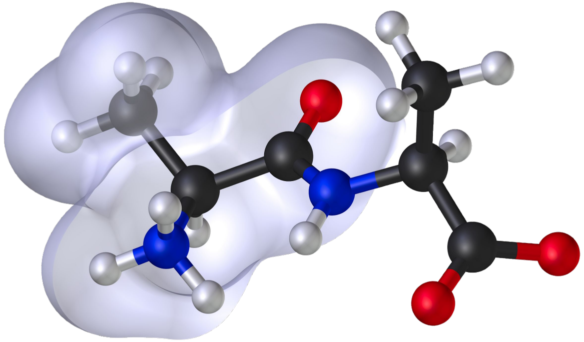
Scientific method



Too small

- Genome sequencing
- Molecular dynamics



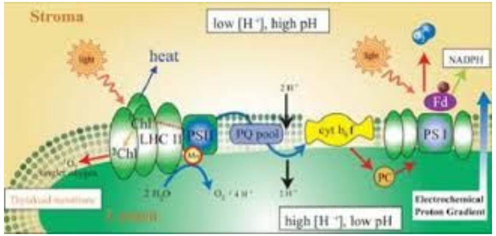


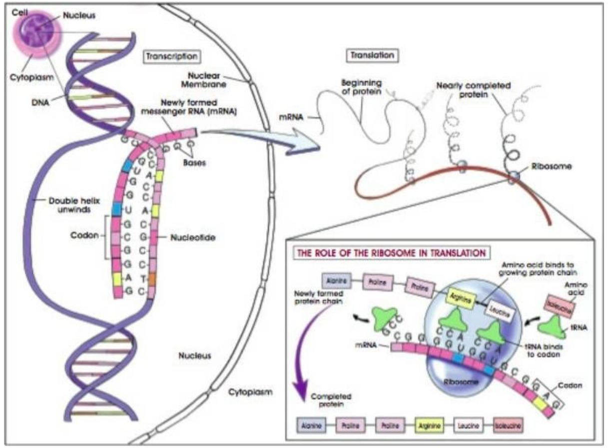
Too big

- Cosmology
- Galaxy formation
- Evolution of stars
- Tracing space probes

Too fast

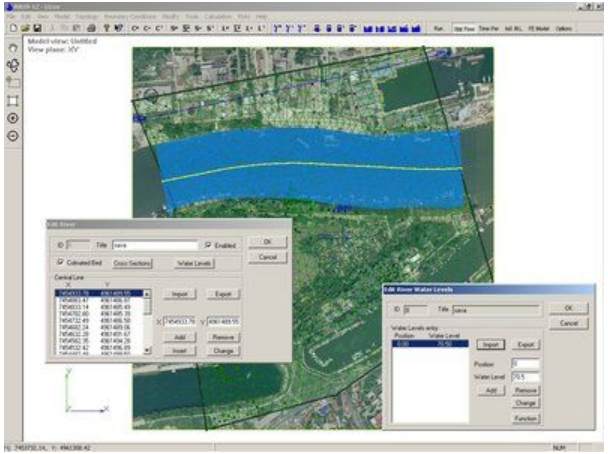
- Fotosynthesis
- Protein synthesis

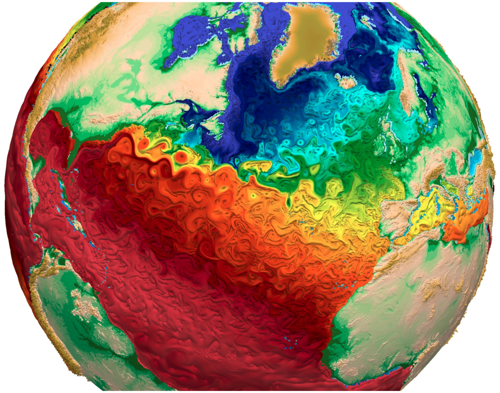


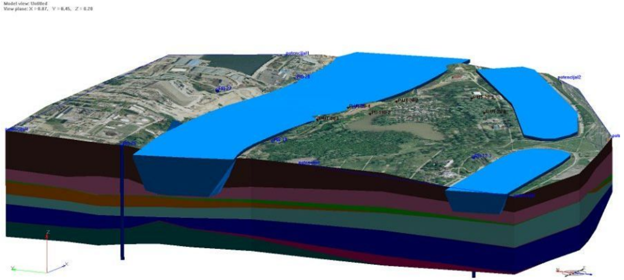


Too slow

- Geology
- Climate change

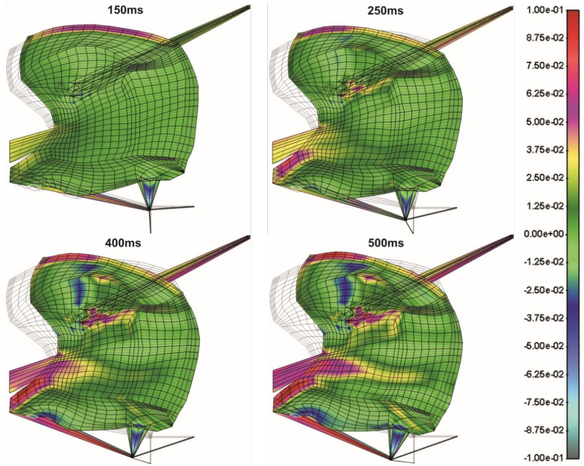


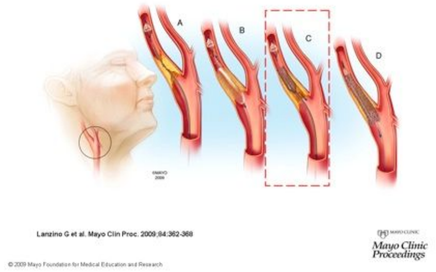


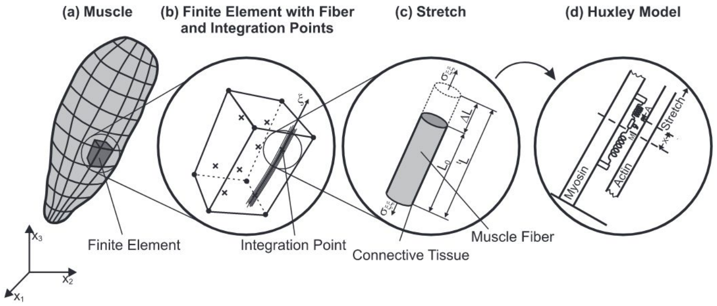


Too complex

- Blood flow
- Multiscale biomechanics
- Weather forecast







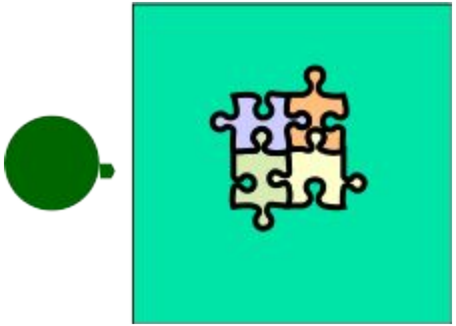
Supercomputers and Formula 1

- >85% performance is pure aerodynamics
- Limiting the number of processor ticks by propositions
- Simulation both when building the car and during the race

Parallel computing techniques

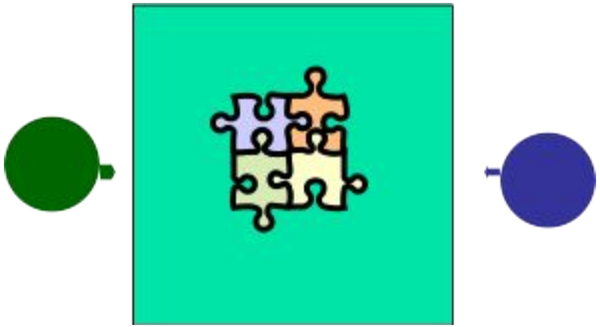
- **Shared memory**
  - Pipelining
  - Threads
  - Manycore architecture (GPUs)
- **Distributed memory**
  - HPC Clusters
  - Message Passing

Analogy of a puzzle



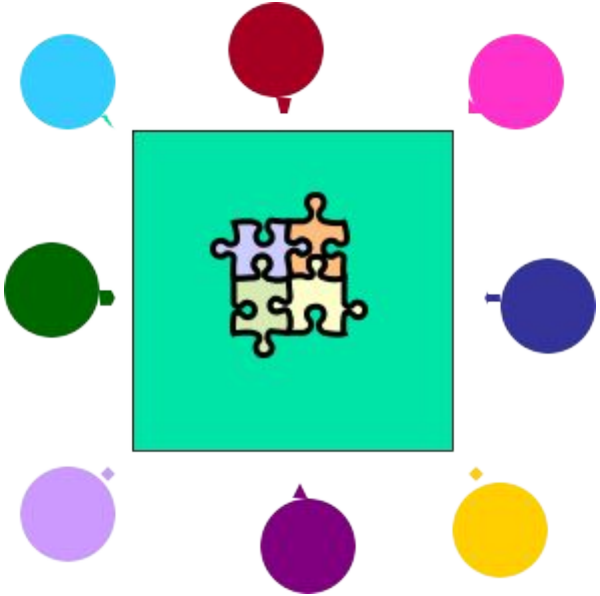
Let's say we need to put together a 1000 piece puzzle. A puzzle of this size certainly requires some time. For example, let it be one hour.

Two participants



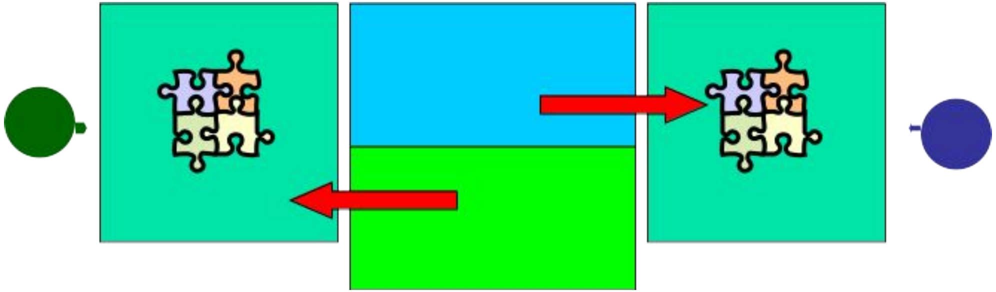
- If we add another participant, he can work on his part of the puzzle.
- However, occasionally both will reach for the same bit at the same time (competing for the same resource), which will cause a slowdown.
- Also, they will occasionally have to work together (communicate) around the border section.
- The acceleration will be almost 2-to-1: for example **35 minutes instead of 30 minutes**.

Better with more participants?



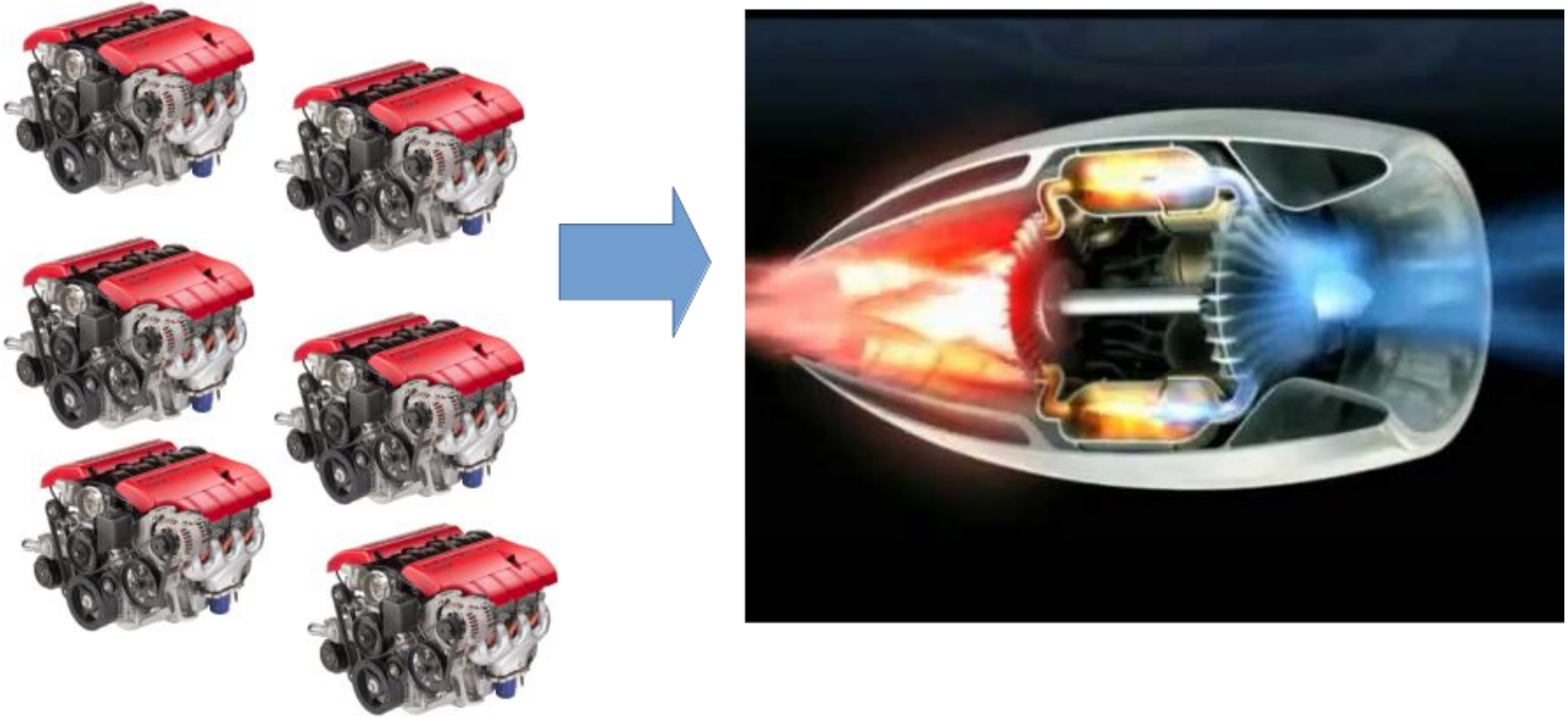
- If we add one more participant to the corners of the board, there will really be a lot of competition, a lot of communication across a large number of borders.
- Therefore, the acceleration will be far less than ideal; we'll be happy if we get 5-to-1
- **The conclusion is that adding new and new participants definitely has less and less effect**

Different way - distribute work

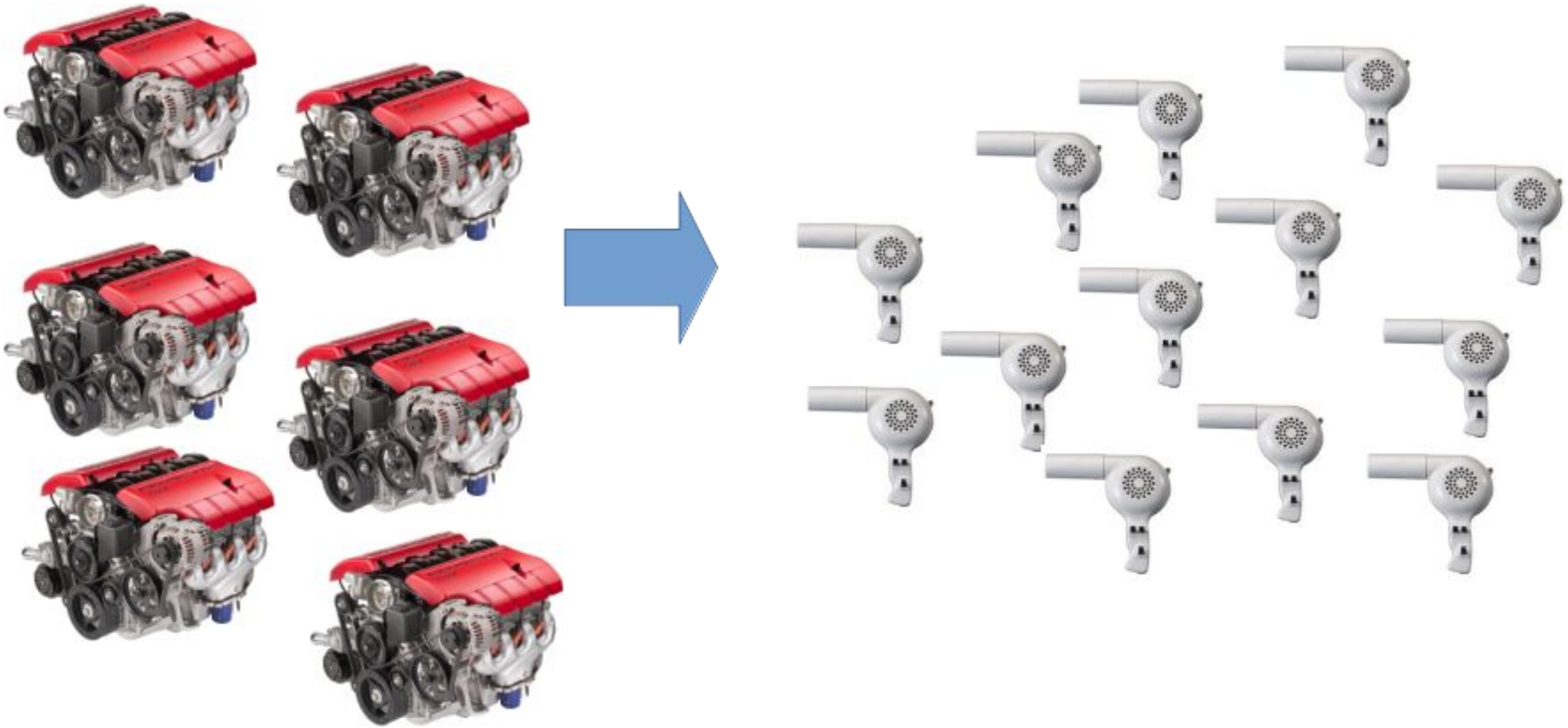


- Let's try something a little different. We will create two special boards.
- We will put half of the pieces on one board and half on the other board.
- Now the participants work completely independently, without any competition for resources
- **BUT, the cost of communication is FAR HIGHER (we need to assemble two parts into one), and we need to divide (decompose) the parts evenly enough, which can be non-trivial for some puzzles.**

What we want

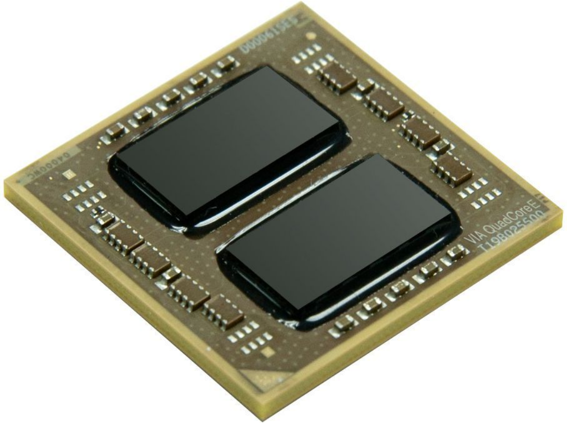


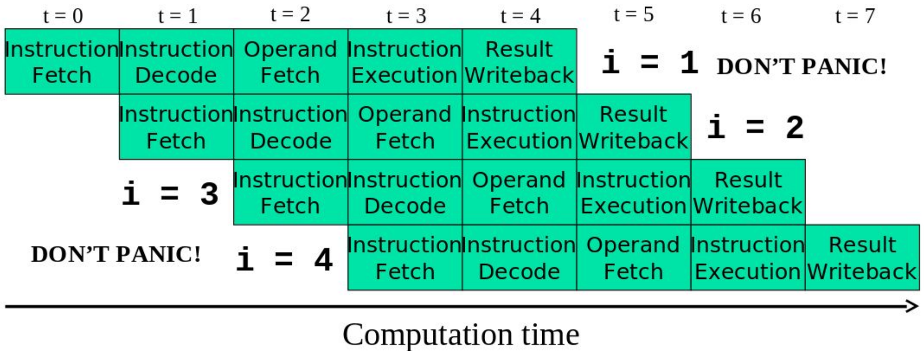
What we get with improper load balance



Shared memory parallelism

- Pipelining since 1992.
- Move away from single-core systems to multi-core processors
- Doesn't help much if developers aren't aware of them
- Serial programs doesn't benefit





Serial solution and thread parallel solution

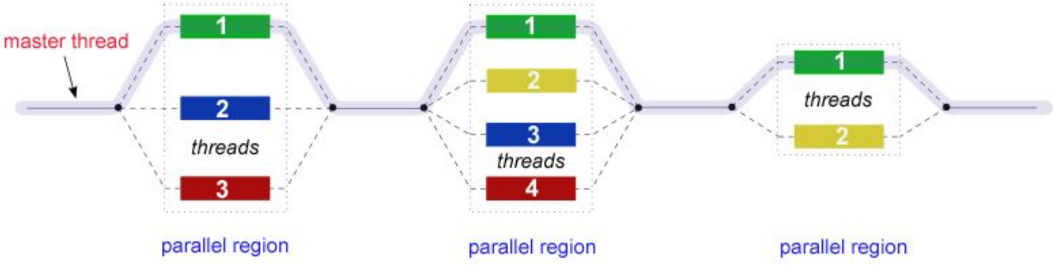
Serial

```
sum = 0;
for (i = 0; i < n; i++) {
    x = Compute_next_value(. . .);
    sum += x;
}
```

Parallel

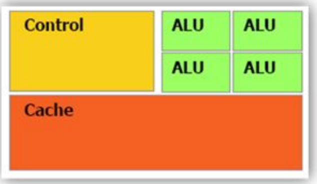
```
my_sum = 0;
my_first_i = . . . ;
my_last_i = . . . ;
for (my_i = my_first_i; my_i < my_last_i; my_i++) {
    my_x = Compute_next_value( . . .);
    my_sum += my_x;
}
```

Each core uses it's own private variables and executes this block of code independently of the other cores.



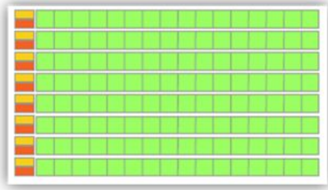
CPU trajectory vs. GPU trajectory

CPU



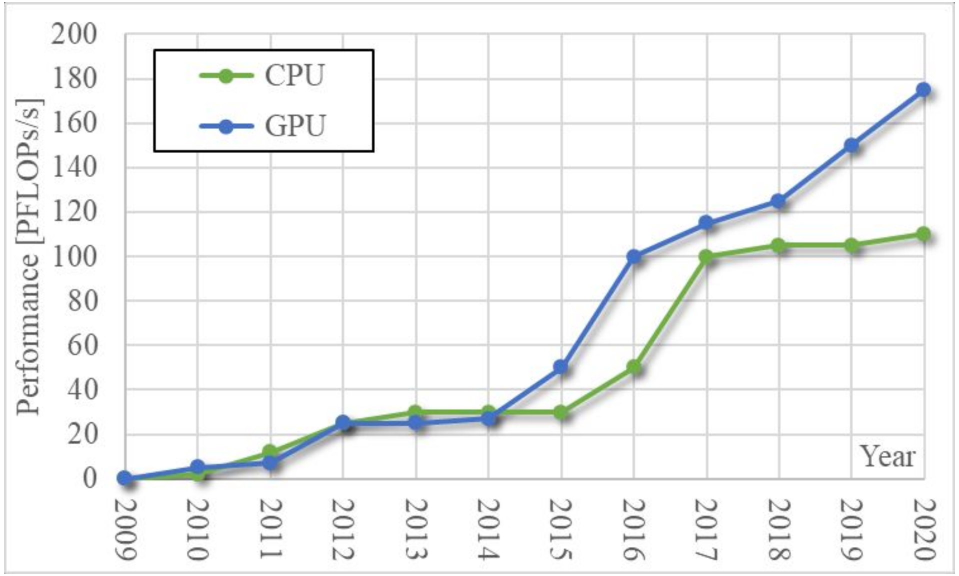
* Low compute density
* Complex control logic
* Large caches (L1$/L2$, etc.)
* Optimized for serial operations
  - Fewer execution units (ALUs)
  - Higher clock speeds
* Shallow pipelines (<30 stages)
* Low Latency Tolerance
* Newer CPUs have more parallelism

GPU



* High compute density
* High Computations per Memory Access
* Built for parallel operations
  - Many parallel execution units (ALUs)
  - Graphics is the best known case of parallelism
* Deep pipelines (hundreds of stages)
* High Throughput
* High Latency Tolerance
* Newer GPUs:
  - Better flow control logic (becoming more CPU-like)
  - Scatter/Gather Memory Access
  - Don't have one-way pipelines anymore

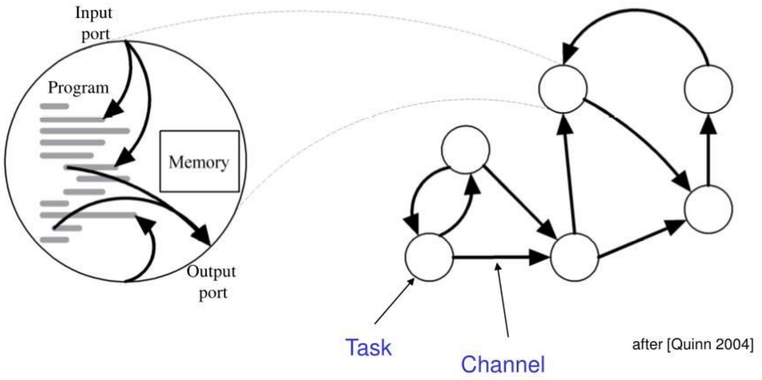
CPU vs. GPU performance



- The industry is moving form “instructions per second” to “instructions per watt”!
- CPU-GPU computing is about offloading compute intensive SIMD tasks onto GPU
- https://www.youtube.com/watch?v=-P28LKWTzrI
- Various libraries to ease the development
  - Numba, CuPy
  - cuBLAS, cuDNN

How to employ multiple nodes?

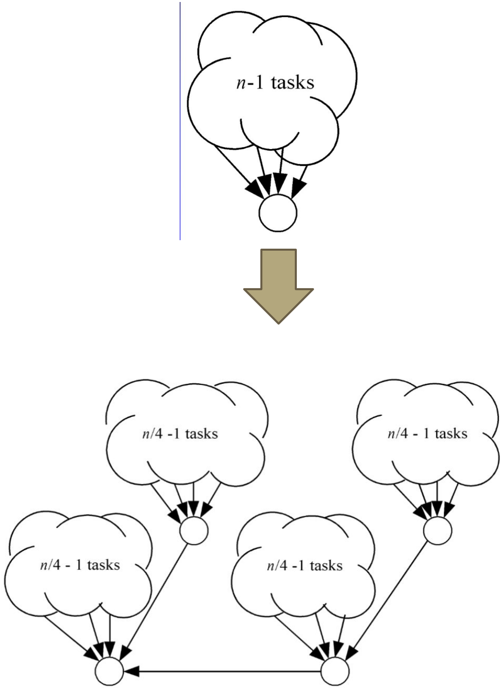
Task/channel model (2)

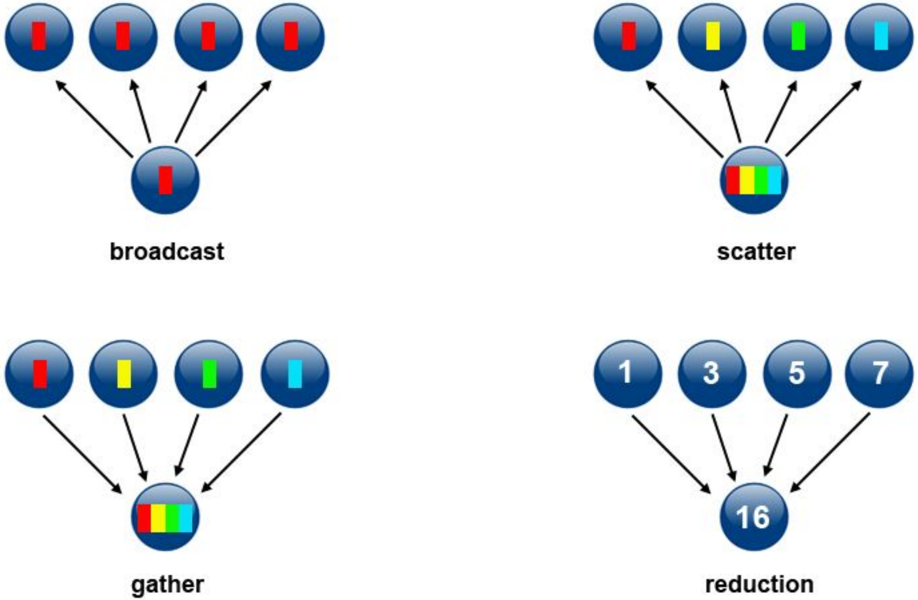


Directed graph of tasks (vertices) and channels (edges)

5

Collective Communications - Logarithmic complexity



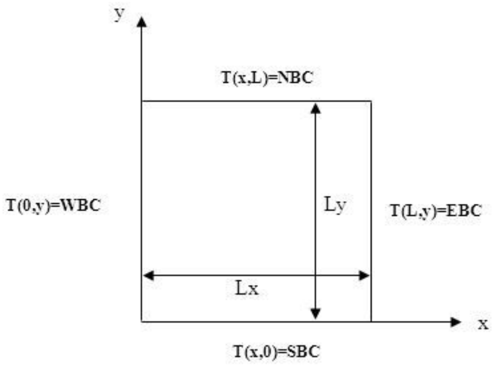


Heat conduction in 2D

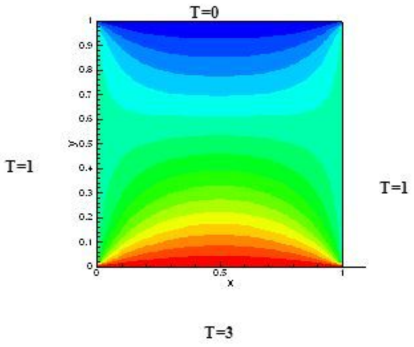
Governing Equation

$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$

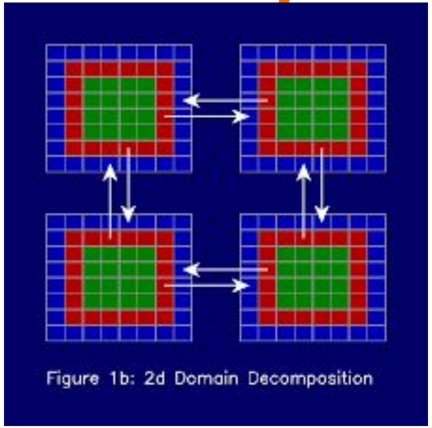
T : Temperature
t : Time
α : Thermal expansion coefficient

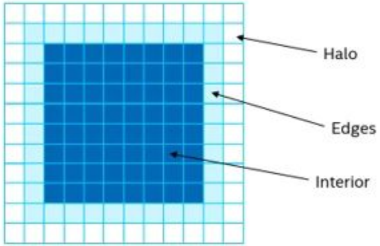


Ex)

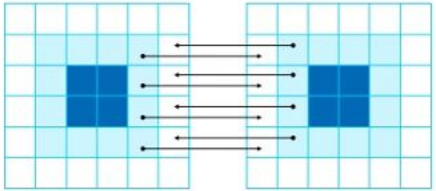


Heat conduction in 2D using MPI





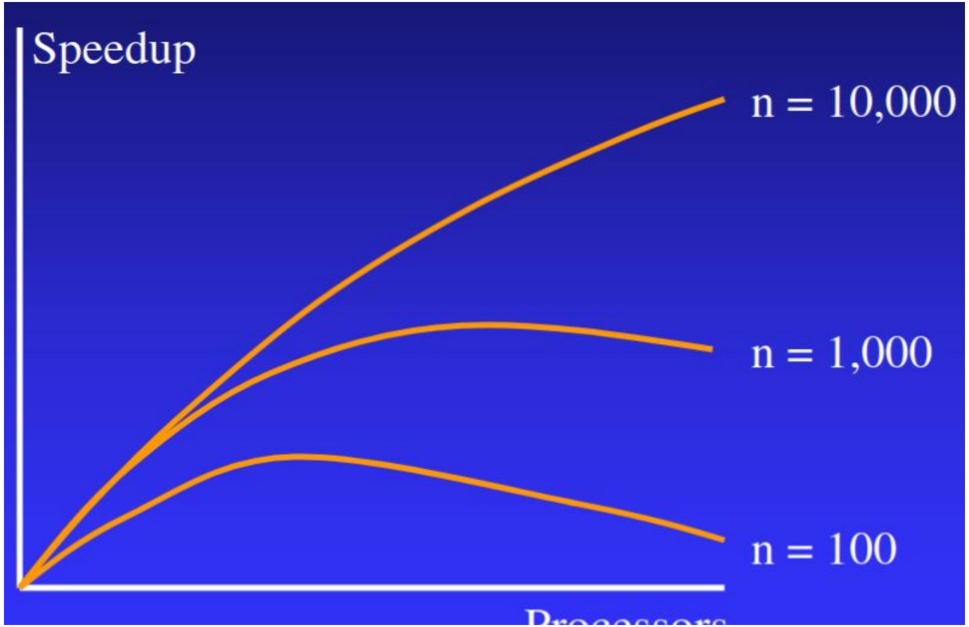
Exchange of edges between processes



Amdahl's law and Amdahl's effect

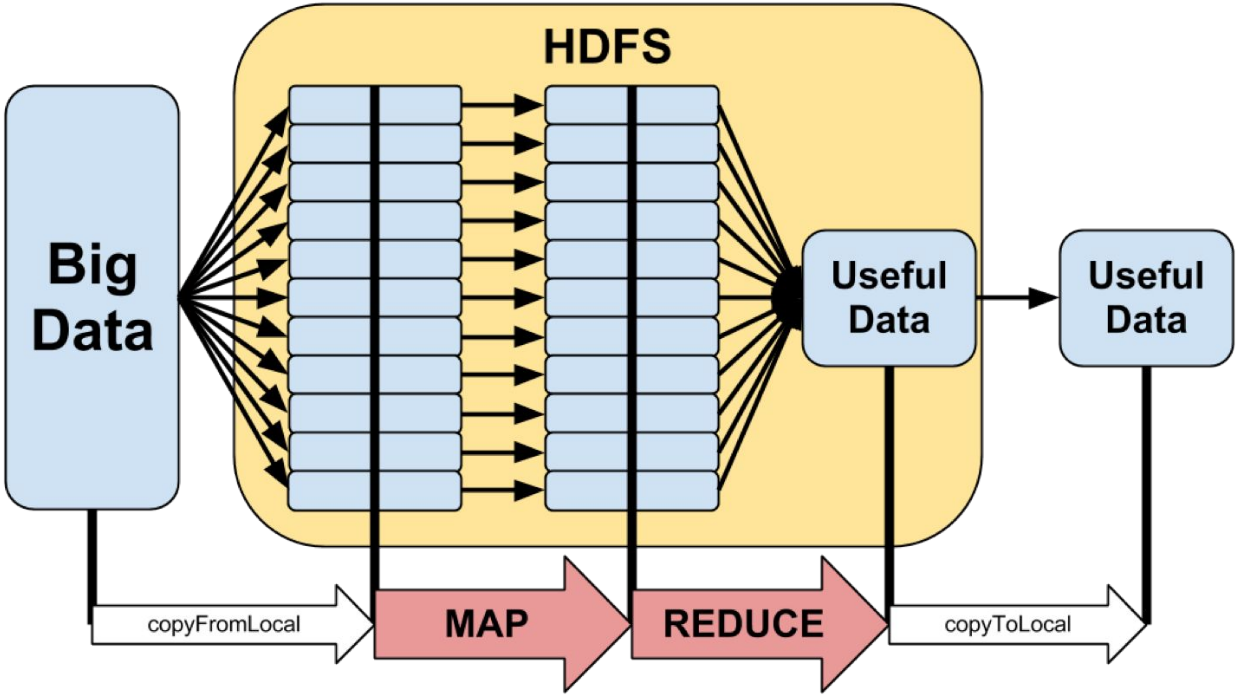
$$\psi(n,p) \le \frac{\sigma(n)+\varphi(n)}{\sigma(n)+\varphi(n)/p+\kappa(n,p)}$$

$$\psi \le \frac{1}{f+(1-f)/p}$$



What is Big Data?

Map/Reduce approach - Apache Spark



Our parallel computing solutions

Faculty of Science - CERAMO

Since the construction, the dam of

Višegrad HYDROPOWER PLANT

has recorded a water leakage

through the KARST TERRAIN,

which by 2010 has risen to

25 Olympic swimming pools per hour,

losing about $1 million per year.

Reducing losses and

preventing further erosion and collapse of the

dam

required urgent remediation.

Višegrad hydropower plant

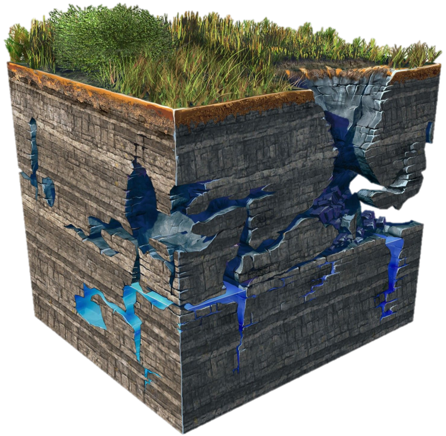
Dam building cost:	$300 million
Water losses:	16 m^3/s
Profit losses:	$1 million per year
Affected:	3 dams, several cities, hundreds of thousands people

The plan was to

fill the cracks under the dam

by stones and

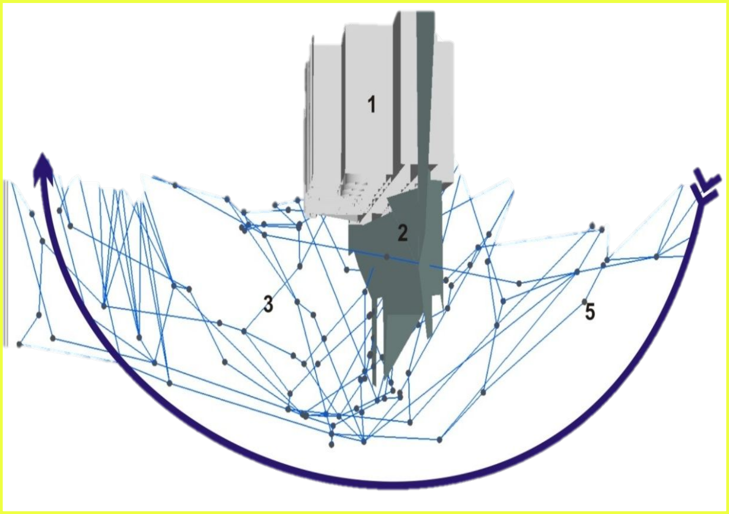
concrete.



But, NOBODY knew

WHERE

the cracks were.



To reveal the enigma,

we have developed

a computational model

that simulates hydraulic

and solute transportation processes

under the dam.

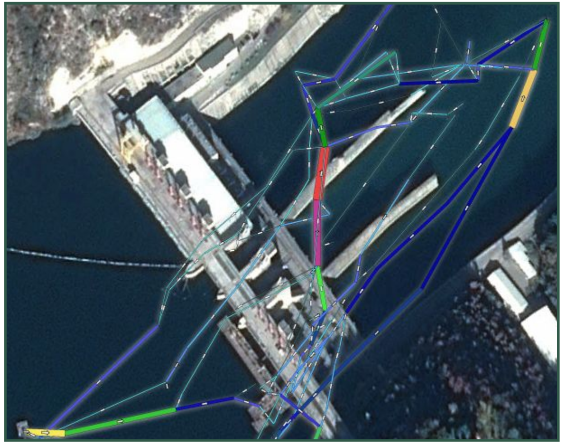
IF we obtained

such DIMENSIONS of the modeled cracks

that give results similar to the MEASUREMENTS,

we could claim that the model represents

a realistic picture of UNDERGROUND NET of fissures.



To solve the problem,
we have developed the software library

for EVOLUTIONARY BASED optimization.

However,

due to algorithm complexity
calculations would last for months,

which was NOT acceptable.

Decisions about remediation

had to be made
on daily basis!

In order to speed up the optimization process

we have developed

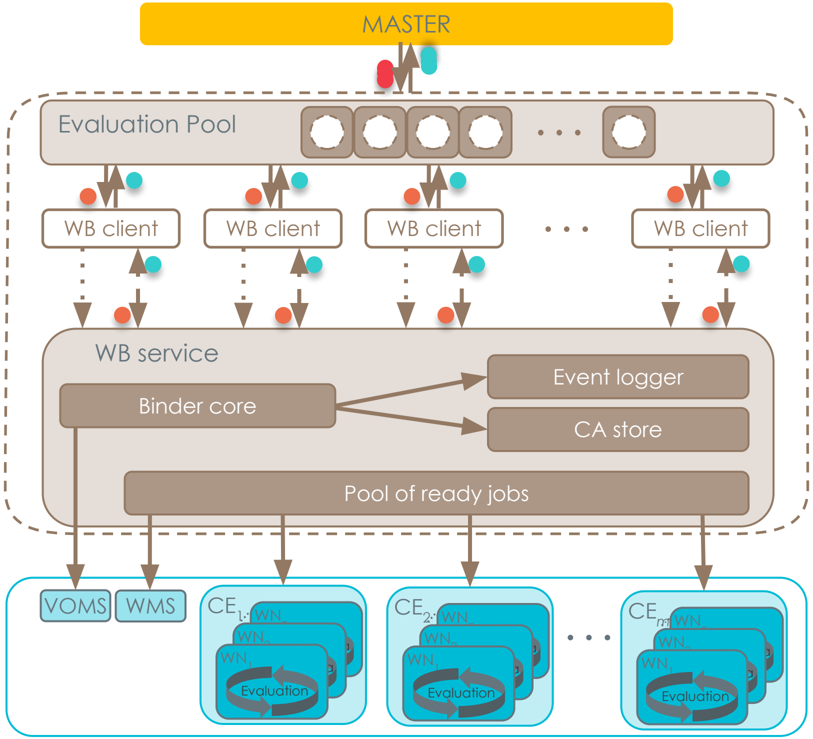
WoBinGO

a software framework for

genetic algorithm based optimizations

in

DISTRIBUTED computing environment.



Using WoBinGO,

estimation of karst configuration under the Višegrad dam

was performed in 10 hours only,

giving a POWERFUL TOOL for making daily decisions about remediation actions.

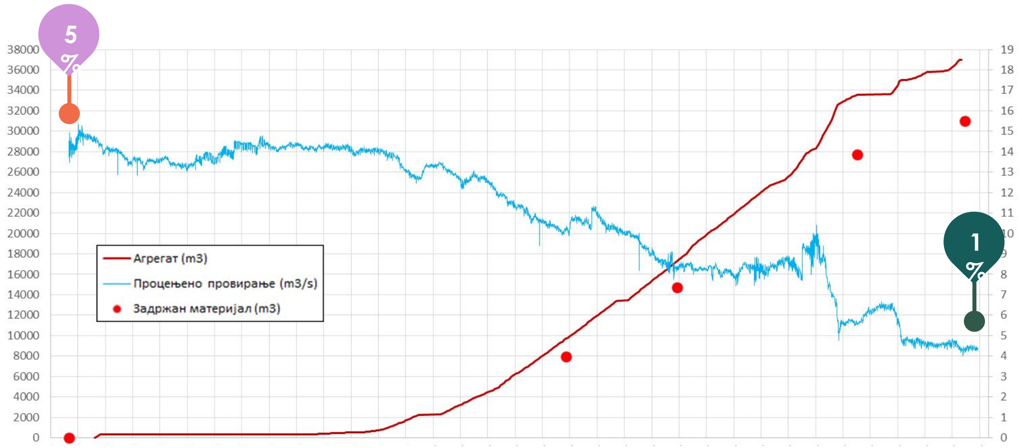
After a year, financed by World Bank,

the remediation was

completed successfully.

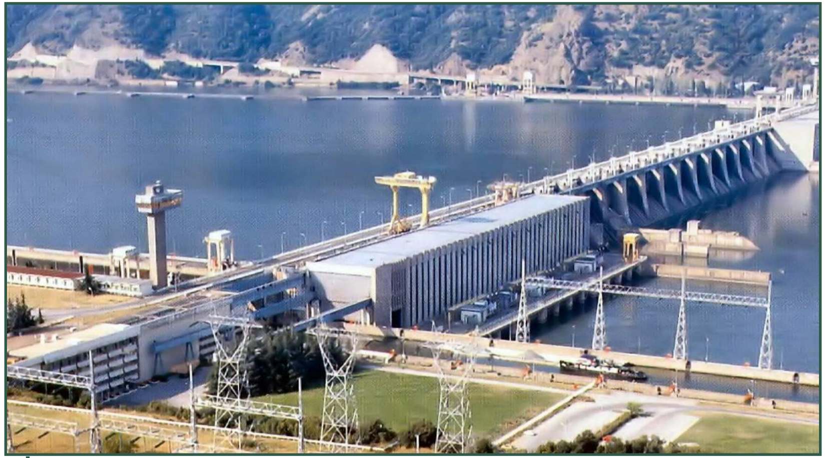
The leakage was

reduced about FIVE times.



Employing WoBinGO,

our spin-off company Vodéna



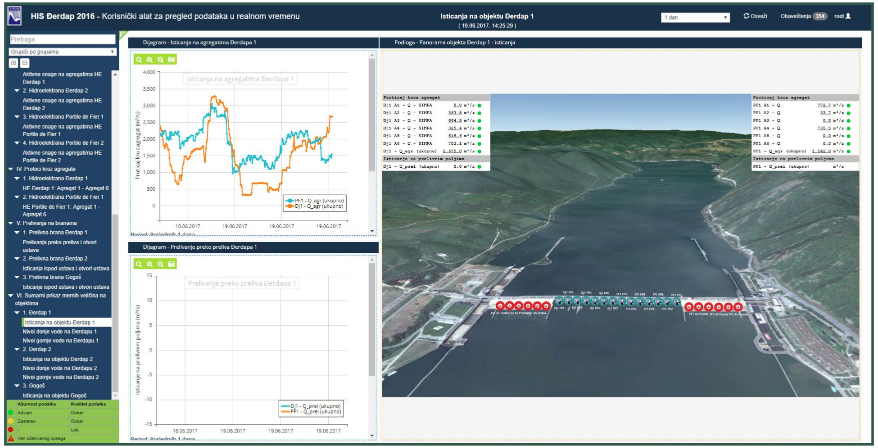
has developed

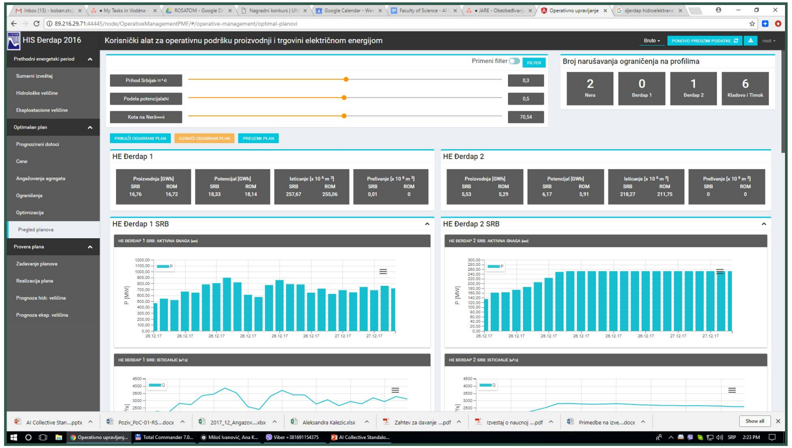
a power production OPTIMIZATION TOOL for the

Iron Gate,

the largest dam on the Danube river

and one of the largest hydro power plants in Europe.





Based on hydrological forecast

and the expected energy prices,

power PRODUCTION is optimized

under the given physical, ecological, and legal CONSTRAINTS.

Sales Forecasting

MİGROS TİCARET A.Ş.

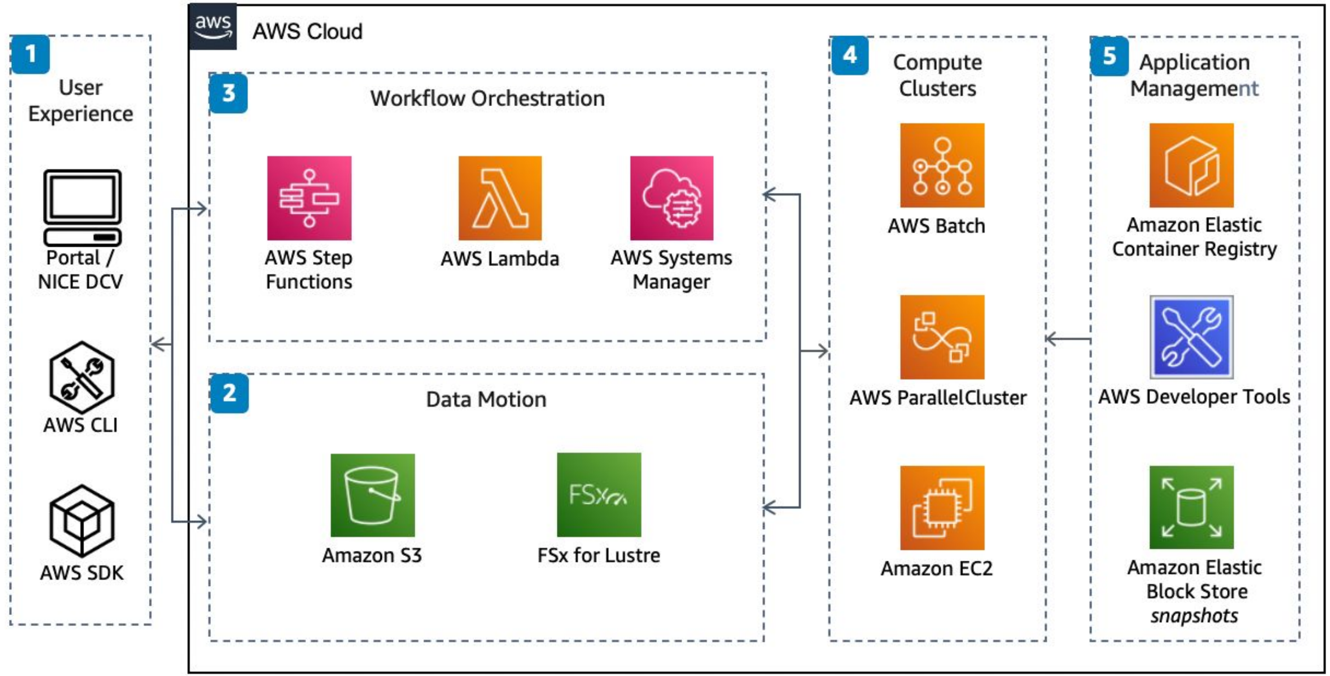
43,000 sales prediction models are simultaneously running, fully automated, monitored, retrained and adjusted

1000x

Increased demand forecasting granularity

MIGROS retail company

HPC in the Cloud



Faculty of Science
CERAMO

mivanovic@kg.ac.rs