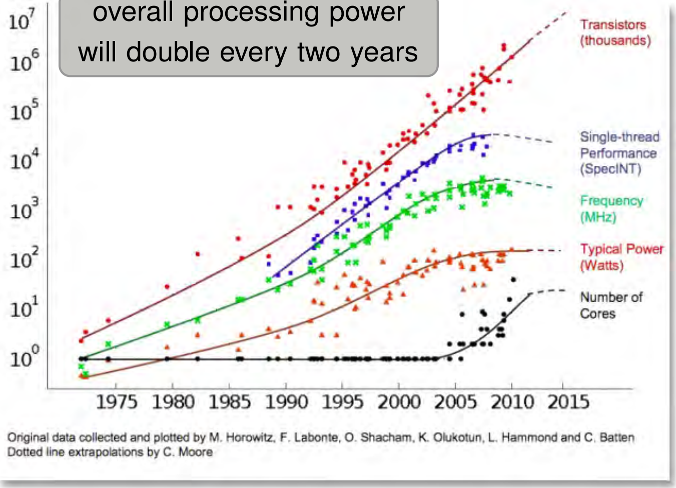
Monte-Carlo-Simulation unter Verwendung von GPU

(Prototype Development of vectorized *BASEMENT*)

Samuel Peter - Davide Vanzo

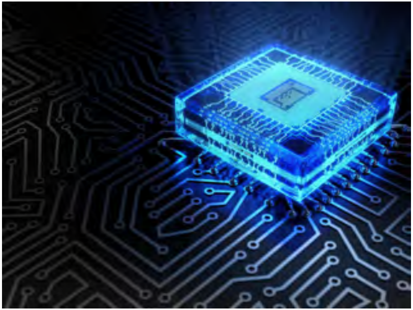
Moores Law

overall processing power will double every two years



source: zdnet.com

Analogy: Processor as Kitchen

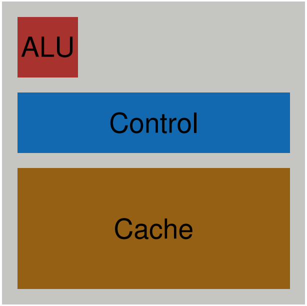


source: hdwlp.com

source: blog.ice.edu

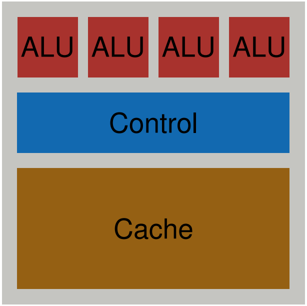
Analogy: Processor as Kitchen

Processor	Kitchen
Code	Recipe
Cache	Working surface
Arithmetic Logic Unit	Stove
Control	Cook

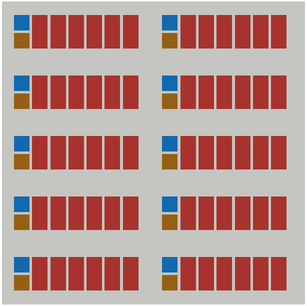


Analogy: Processor as Kitchen

multithreaded CPU



general purpose GPU



Key Development Challenges

order → Recipe Kitchen → delivery

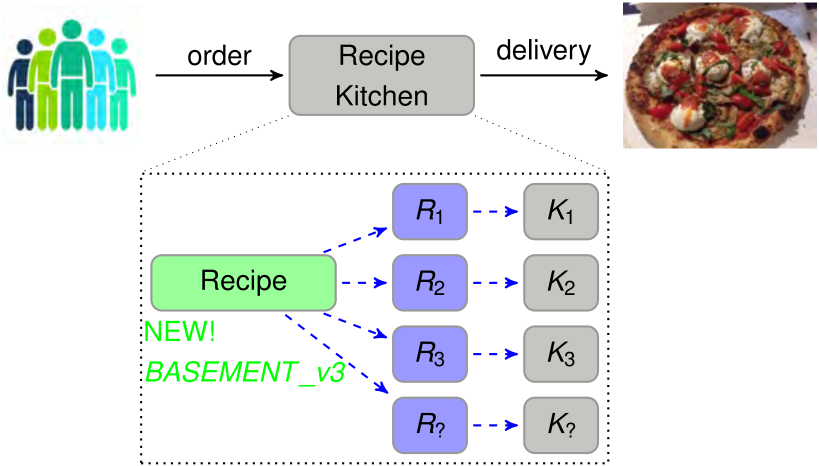
Clients:

- consistency
- quality
- speed

Producer:

- longevity of recipe
- independent of kitchen

Key Development Challenges



OP2 Framework

- Developed by mathematicians and software engineers at Oxford University and Imperial College, mainly funded by Rolls-Royce.
- *Open-source* framework for the execution of *unstructured grid* applications.
- Generating appropriate *back-end code* for different target platforms by *source-source translation*.

→ *BASEMENT_v3*:

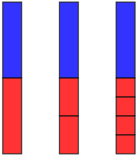
One recipe with highlighting *ingredients*, *actions*, *order* etc.

Data Parallelism

Amdahl's law (1967)

Speedup limit for fixed workload

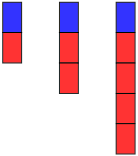
$$S(N) \leq \frac{1}{(1-p) + \frac{p}{N}}$$



Gustafson's law (1988)

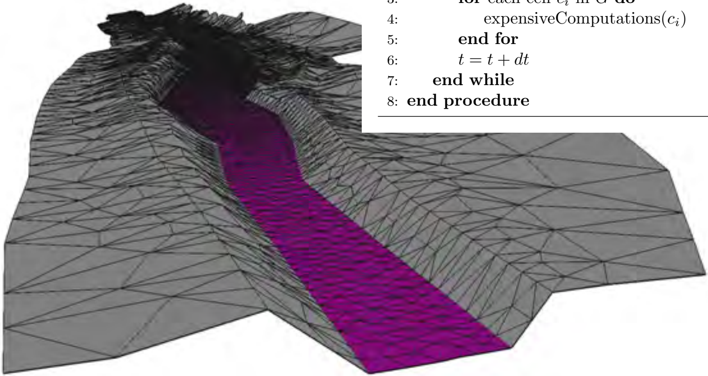
Speedup limit for fixed time

$$S(N) \leq 1 + p(N-1)$$



fraction of parallel code p, number of parallel threads N

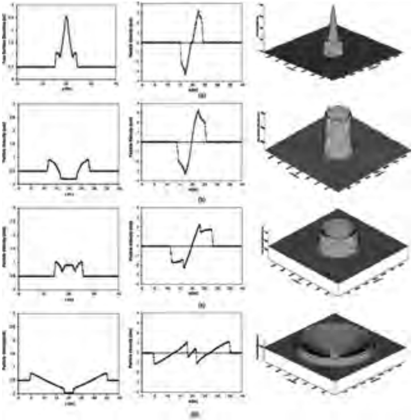
Data Parallelism: Solving SWE



```
procedure FloodWaveCalculation(grid G)
    while t ≤ t_max do
        for each cell c_i in G do
            expensiveComputations(c_i)
        end for
        t = t + dt
    end while
end procedure
```

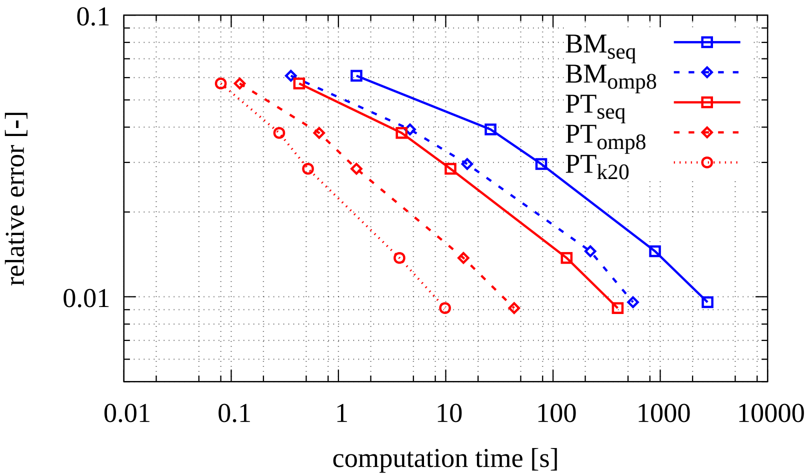
Benchmark: circular dam break

- Toro's circular dam break test with reference solution
- frictionless, *CFL* = 0.8, t_{out} = 3.5 s
- 5 meshes: 10k, 50k, 100k, 500k, 1000k cells
- 5 architectures:
  - *BASEMENT_v2.7*: sequential, 8 CPUs
  - *Prototype*: sequential, 8 CPUs, NVIDIA Kepler20 GPU



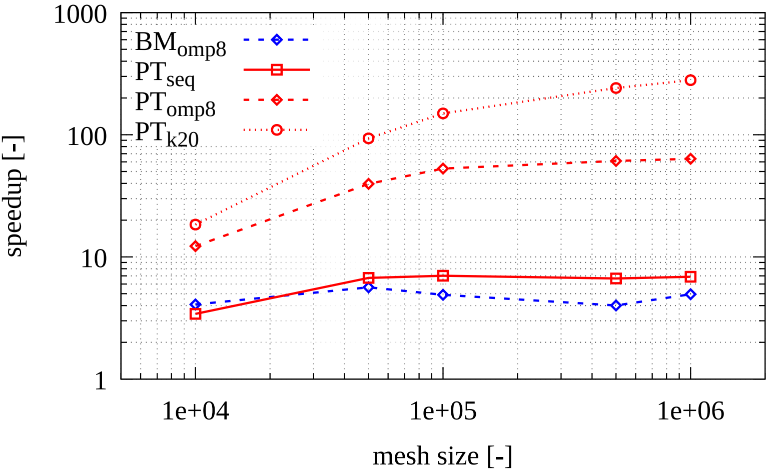
source: Skoula, Borthwick, & Moutzouris (2006)

Benchmark: circular dam break



source: ICOLD Benchmark, Graz (2013)

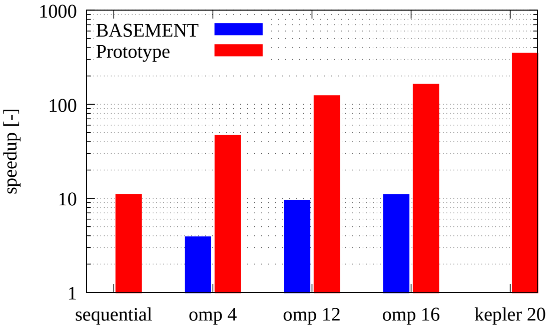
Benchmark: circular dam break



source: ICOLD Benchmark, Graz (2013)

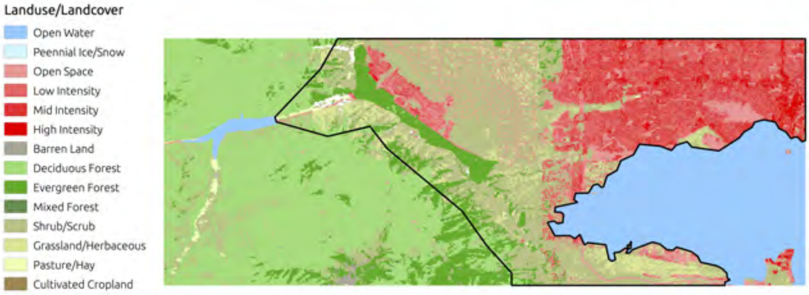
Benchmark: sediment dune & uniform flow

- Grass bedload formula, $CFL = 0.8$, $t_{out} = 3\,h$, mesh with 32’000 cells
- 9 architectures:
  - *BASEMENT_v2.6*: sequential / 4, 12, 16 CPUs
  - *Prototype*: sequential / 4, 12, 16 CPUs / NVIDIA Kepler20 GPU



Case Study: Monte Carlo Simulation

Data	ICOLD Benchmark, Graz 2013
Breachmodel	*BASEbreach*, probabilistic hydrograph
Floodmodel	*BASEMENT*, vectorized prototype, GPU NVIDIA K20



Case Study: Monte Carlo Simulation

Computational grid:

triangular mesh
200'000 cells

Solving shallow water equations:

accurate HLLC solver
explicit time integration (CFL=0.8)
real runtime: 12h
computation time: 1min

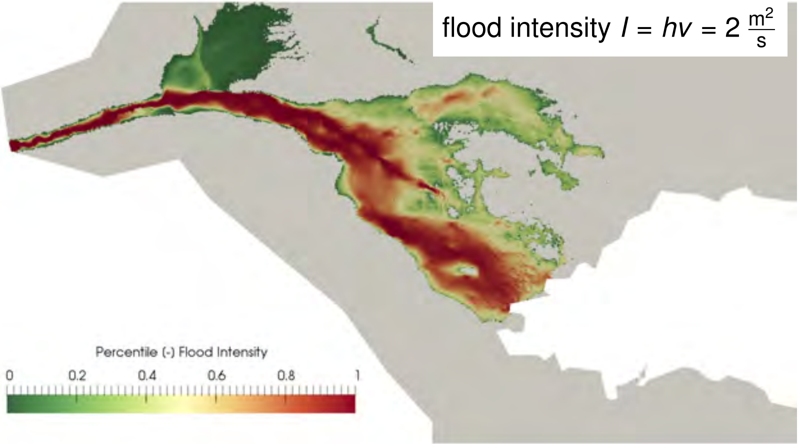
Monte-Carlo simulation:

stochastic upper boundary condition
24h of computations → 1500 simulations

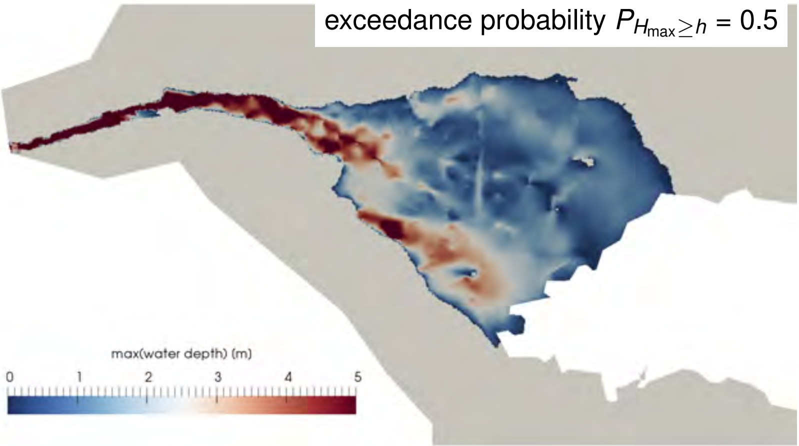
movie

Case Study: Monte Carlo Simulation

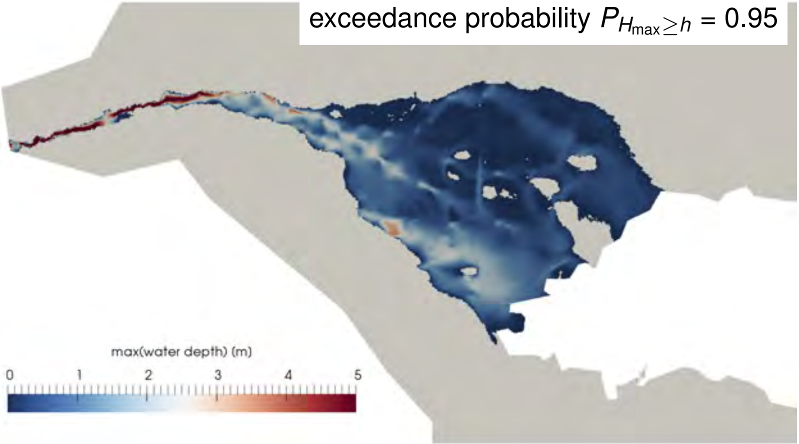
flood intensity $I = hv = 2\ \frac{m^2}{s}$



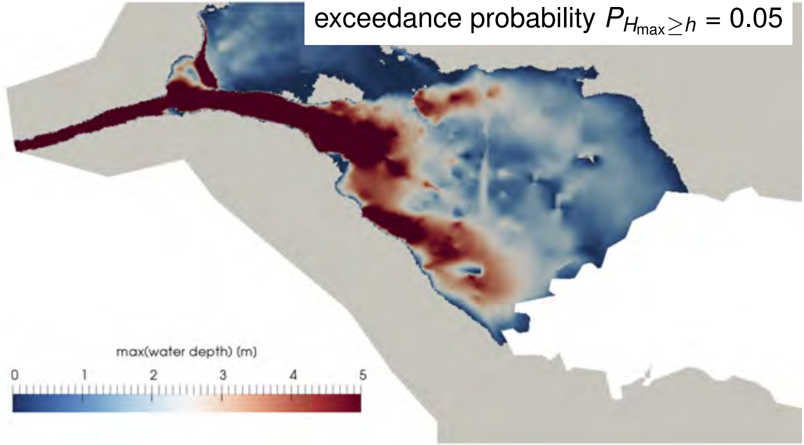
Case Study: Monte Carlo Simulation



Case Study: Monte Carlo Simulation



Case Study: Monte Carlo Simulation



Conclusions

- Vectorized *BASEMENT_v3* succesfully reached prototype status.
- One code, multiple architectures (sequential, multithreaded CPU, GPU, etc.) thanks to OP2 framework.
- Compared to current *BASEMENT* version, increased performance due to consequent data vectorization and new processor technologies.
- Achieved speedup implies a *game changer* for many applications of *BASEMENT*