

HIGH PERFORMANCE COMPUTING: TOWARDS BETTER PERFORMANCE PREDICTIONS AND EXPERIMENTS

Tom Cornebize

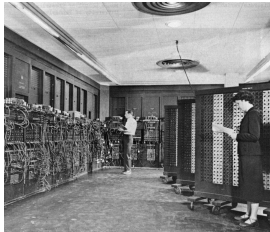
2 June 2021, PhD defense



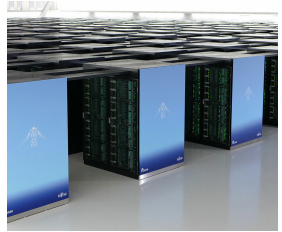
NO SCIENCE WITHOUT COMPUTING



Arithmometer (1851)



ENIAC (1945)

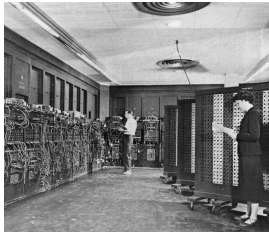


Fugaku (2021)

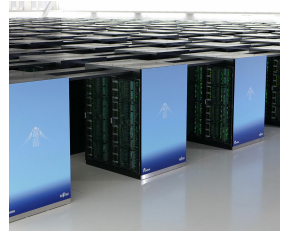
NO SCIENCE WITHOUT COMPUTING



Arithmometer (1851)



ENIAC (1945)

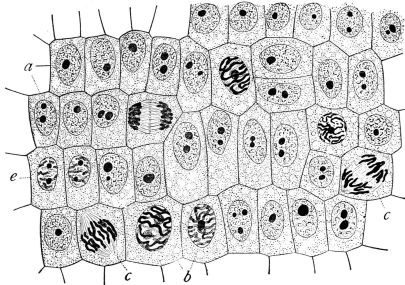


Fugaku (2021)

Last decades:

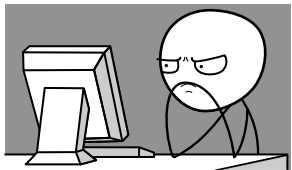
- Exponential **performance** improvements (e.g. sequencing an entire human genome costed \$100,000,000 in 2001, \$1000 now)
- At the price of **complexity** (both software and hardware)

EXPERIMENTAL STUDY OF COMPUTER PERFORMANCE



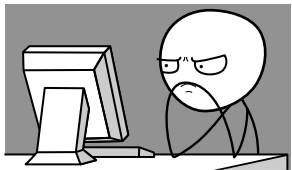
Complexity \Rightarrow Variability and Opacity
 \Rightarrow No perfect model
 \Rightarrow Need for experiments

Typical Performance Evaluation Questions (Given my application and a supercomputer)



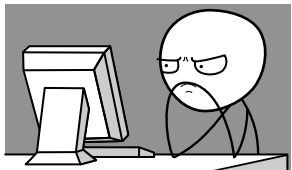
- **Before** running
 - How many nodes?
 - For how long?
 - Which parameters?

Typical Performance Evaluation Questions (Given my application and a supercomputer)



- **Before** running
 - How many nodes?
 - For how long?
 - Which parameters?
- **After** running
 - Performance as “expected”?
 - Problem in the app or the platform?

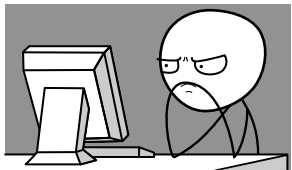
Typical Performance Evaluation Questions (Given my application and a supercomputer)



- **Before** running
 - How many nodes?
 - For how long?
 - Which parameters?
- **After** running
 - Performance as “expected”?
 - Problem in the app or the platform?

So many large-scale runs, solely to tune performance?!?

Typical Performance Evaluation Questions (Given my application and a supercomputer)



- **Before** running
 - How many nodes?
 - For how long?
 - Which parameters?
- **After** running
 - Performance as “expected”?
 - Problem in the app or the platform?


So many large-scale runs, solely to tune performance?!?

Holy Grail: Predictive Simulation on a “Laptop”

Capture the **whole application** and **platform complexity**

SIM(EM)ULATION: THE SMPI APPROACH




Full reimplementation of MPI on top of 

- C/C++/F77/F90 codes run **unmodified out of the box**
- Simply replace mpicc/mpirun by smpicc/smpirun



SIM(EM)ULATION: THE SMPI APPROACH



Full reimplementation of MPI on top of 

- C/C++/F77/F90 codes run **unmodified out of the box**
- Simply replace mpicc/mpirun by smpicc/smpirun



Emulation: how?

- Computations run for real on a laptop
- Communications are faked, good fluid network models
- **Performance model** for the target platform

Goal: **predict** the performance of a parallel application

Goal: **predict** the performance of a parallel application

Thesis contribution

- Case study: High Performance Linpack (HPL)
- Skip the expensive computations (mostly **dgemm**) and replace them by a performance model
- Extensive (in)validation, comparing simulations with reality