UiT

THE ARCTIC
UNIVERSITY
OF NORWAY

Introduction

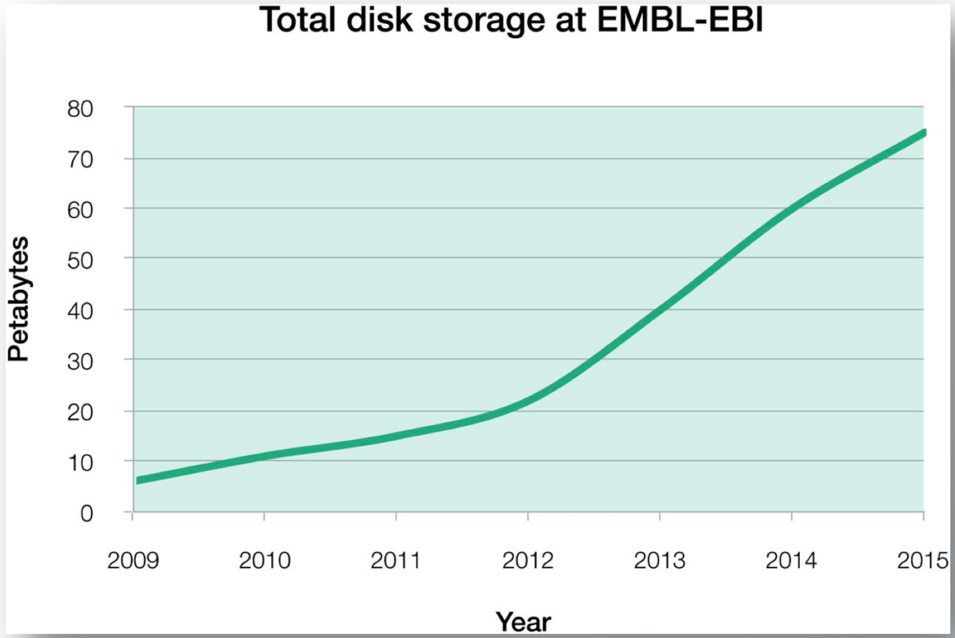
inf-2202 Concurrent and Data-intensive Programming

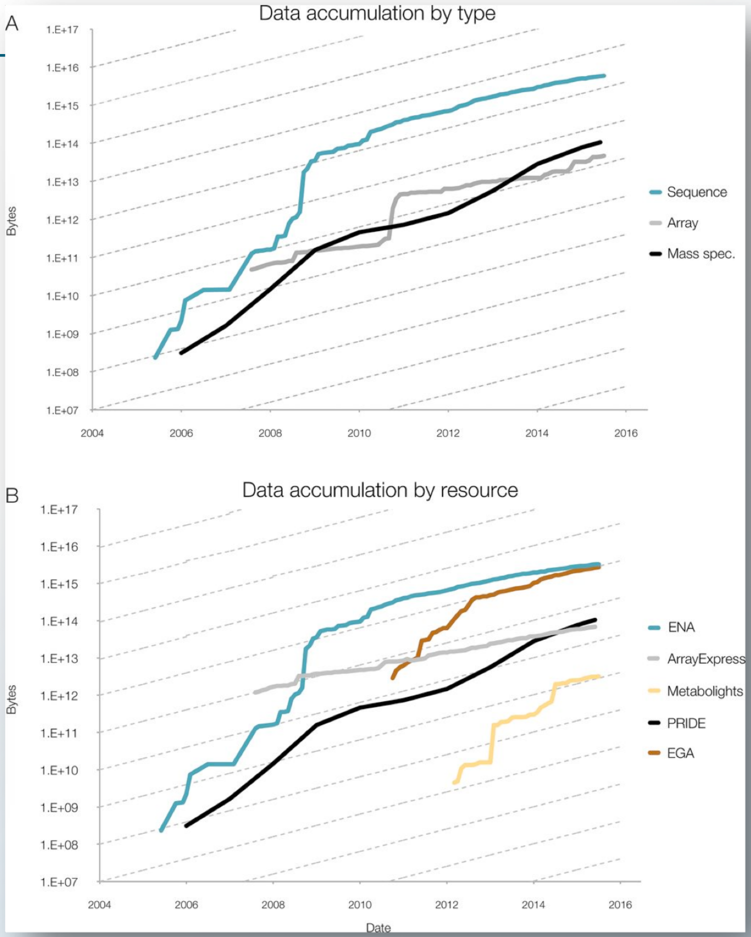
Fall 2017

Lars Ailo Bongo (larsab@cs.uit.no)

Outline

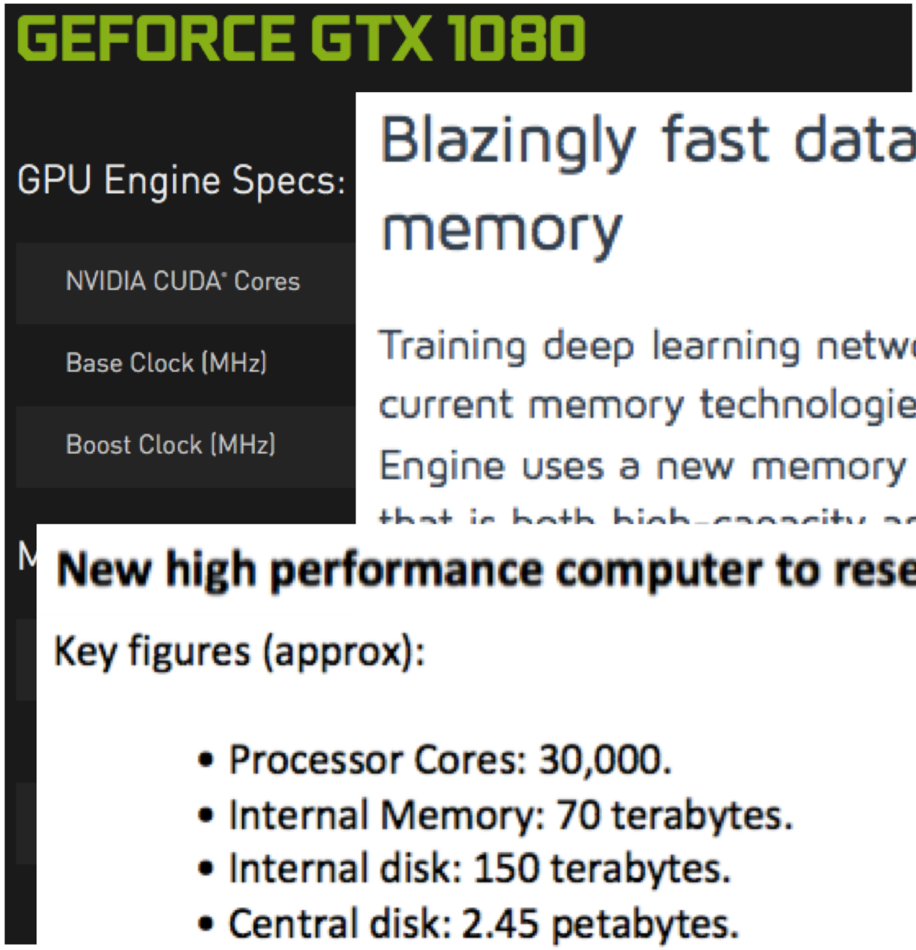
- Why?
- What?
- How?





Data growth at EMBL-EBI

Source: Charles E. Cook et al. *Nucl. Acids Res.* 2016;44:D20-D26



GEFORCE GTX 1080

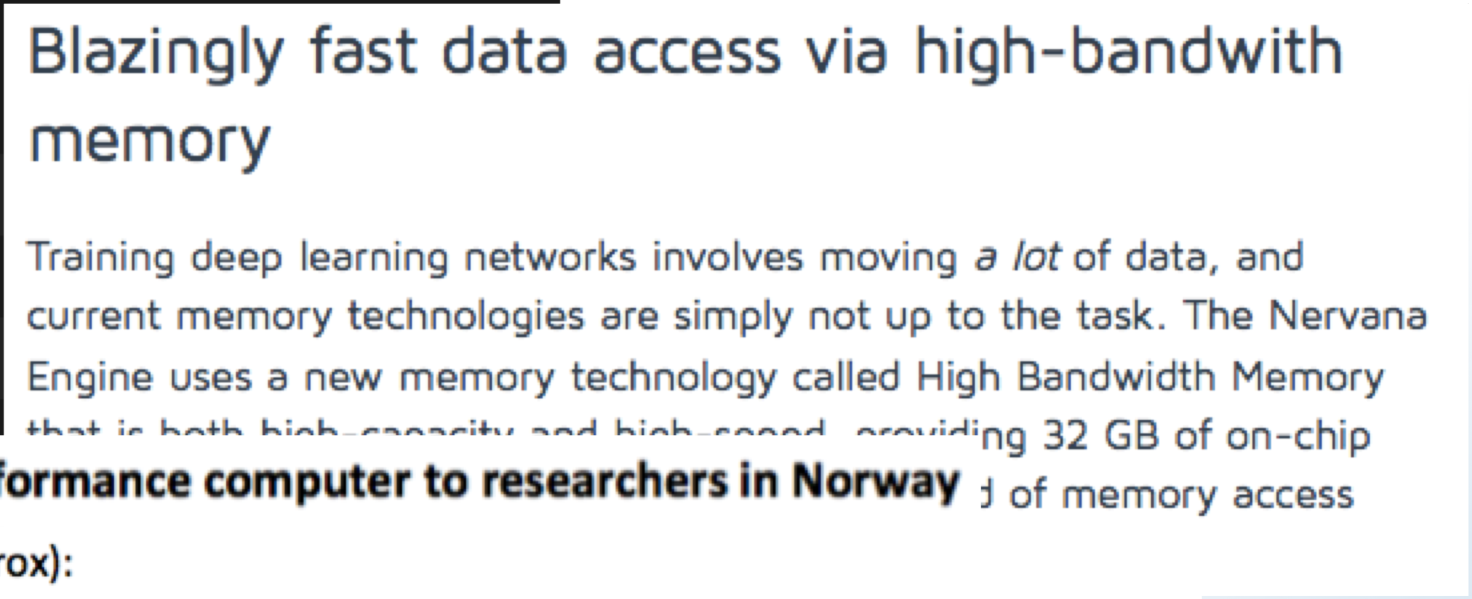
GPU Engine Specs:

NVIDIA CUDA® Cores

Base Clock (MHz)

Boost Clock (MHz)

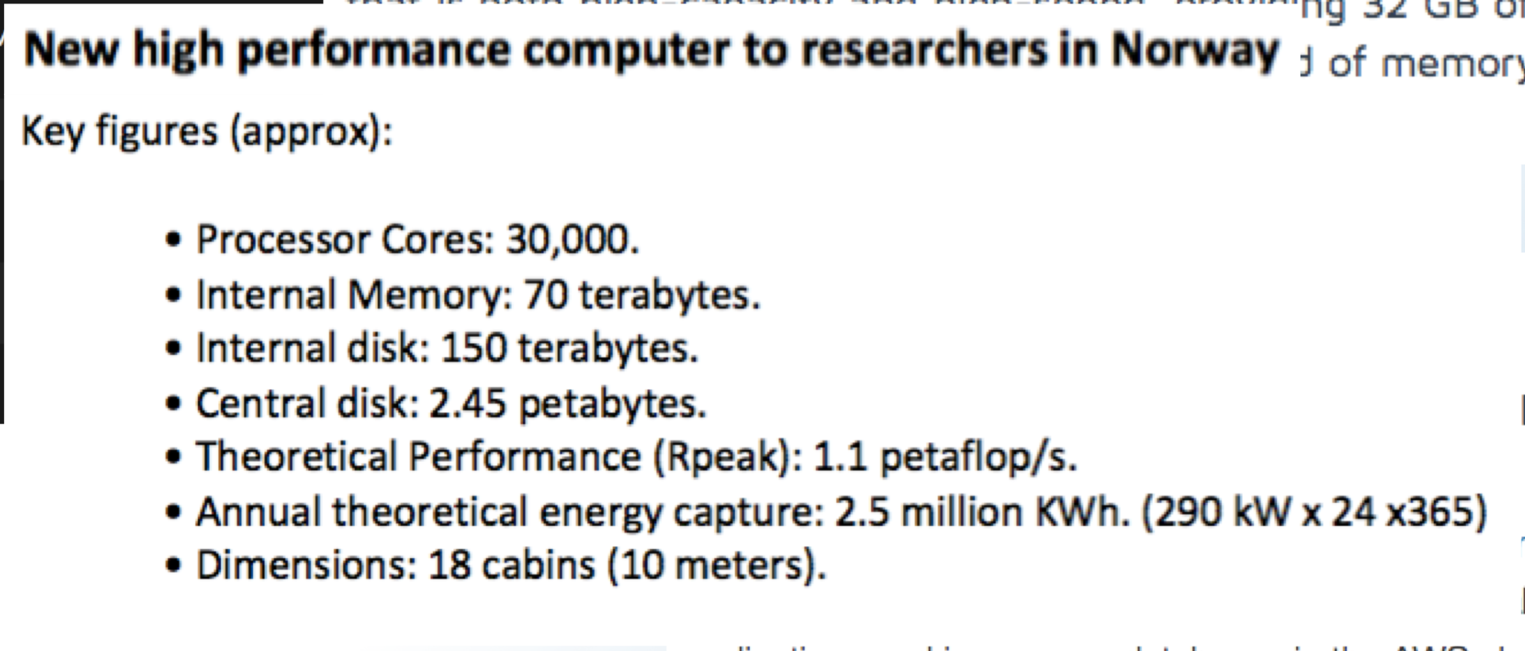
M



Blazingly fast data access via high-bandwith memory

Training deep learning networks involves moving *a lot* of data, and current memory technologies are simply not up to the task. The Nervana Engine uses a new memory technology called High Bandwidth Memory that is both high-capacity and high-speed, providing 32 GB of on-chip

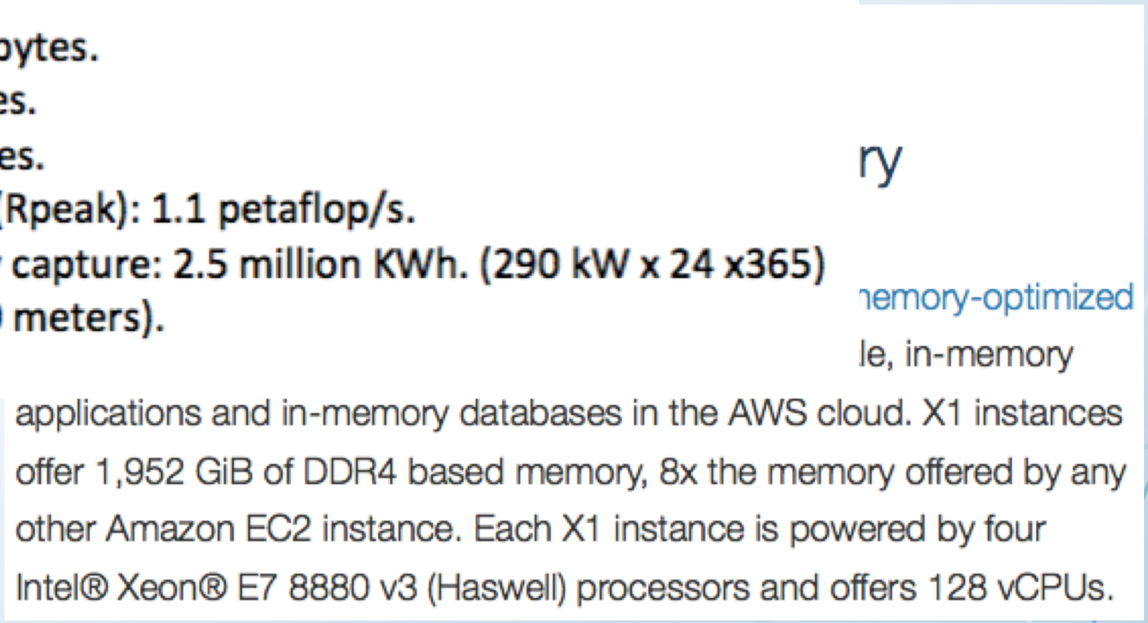
of memory access



New high performance computer to researchers in Norway

Key figures (approx):

- Processor Cores: 30,000.
- Internal Memory: 70 terabytes.
- Internal disk: 150 terabytes.
- Central disk: 2.45 petabytes.
- Theoretical Performance (Rpeak): 1.1 petaflop/s.
- Annual theoretical energy capture: 2.5 million KWh. (290 kW x 24 x365)
- Dimensions: 18 cabins (10 meters).



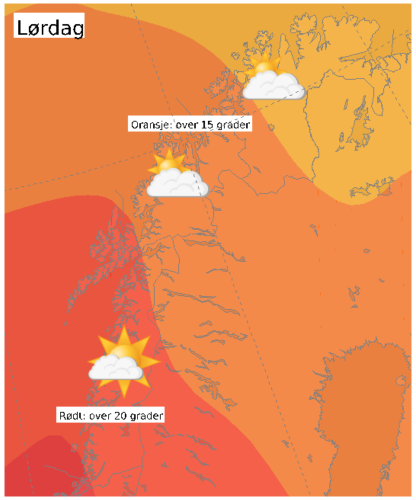
ry

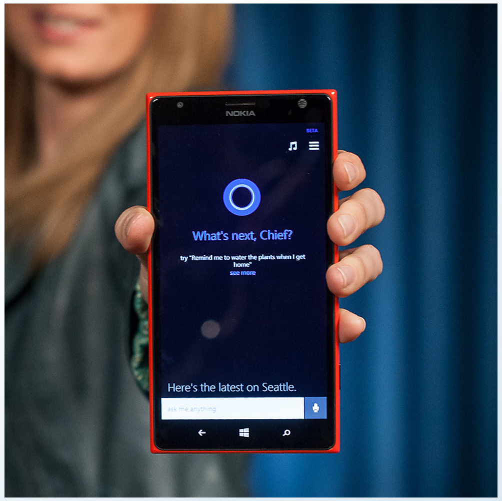
emory-optimized

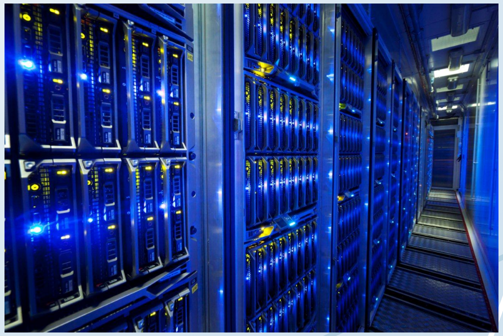
le, in-memory

applications and in-memory databases in the AWS cloud. X1 instances offer 1,952 GiB of DDR4 based memory, 8x the memory offered by any other Amazon EC2 instance. Each X1 instance is powered by four Intel® Xeon® E7 8880 v3 (Haswell) processors and offers 128 vCPUs.

What can we do with all the data and cores?




NOKIA
What's next, Chief?
try "Remind me to water the plants when I get home"
see more
Here's the latest on Seattle.



Motivation

- “The core is the logic gate of the 21st century”
  Anant Agarwal, MIT
  - The number of cores will double every 18th month

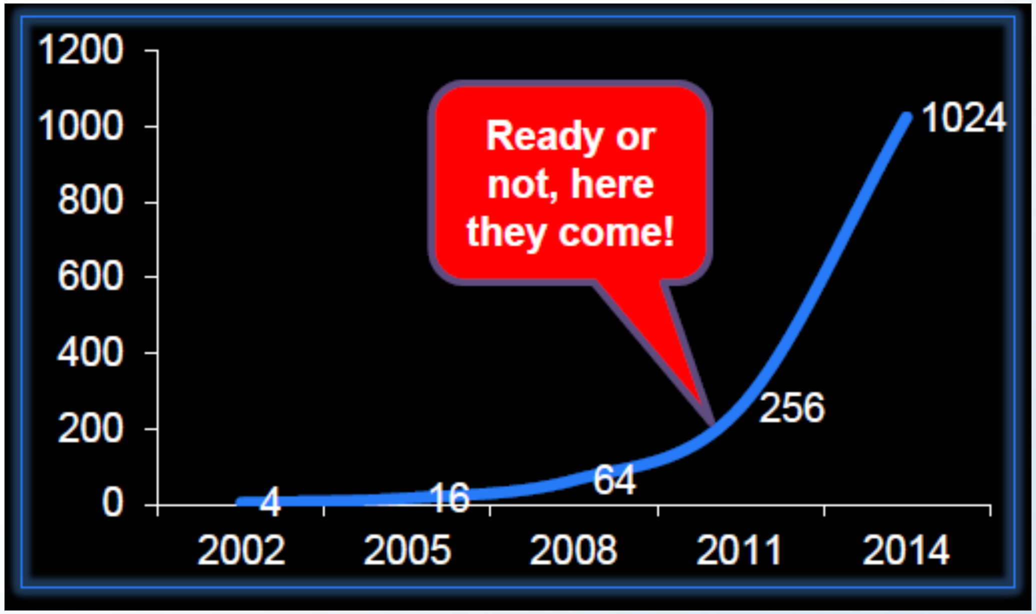
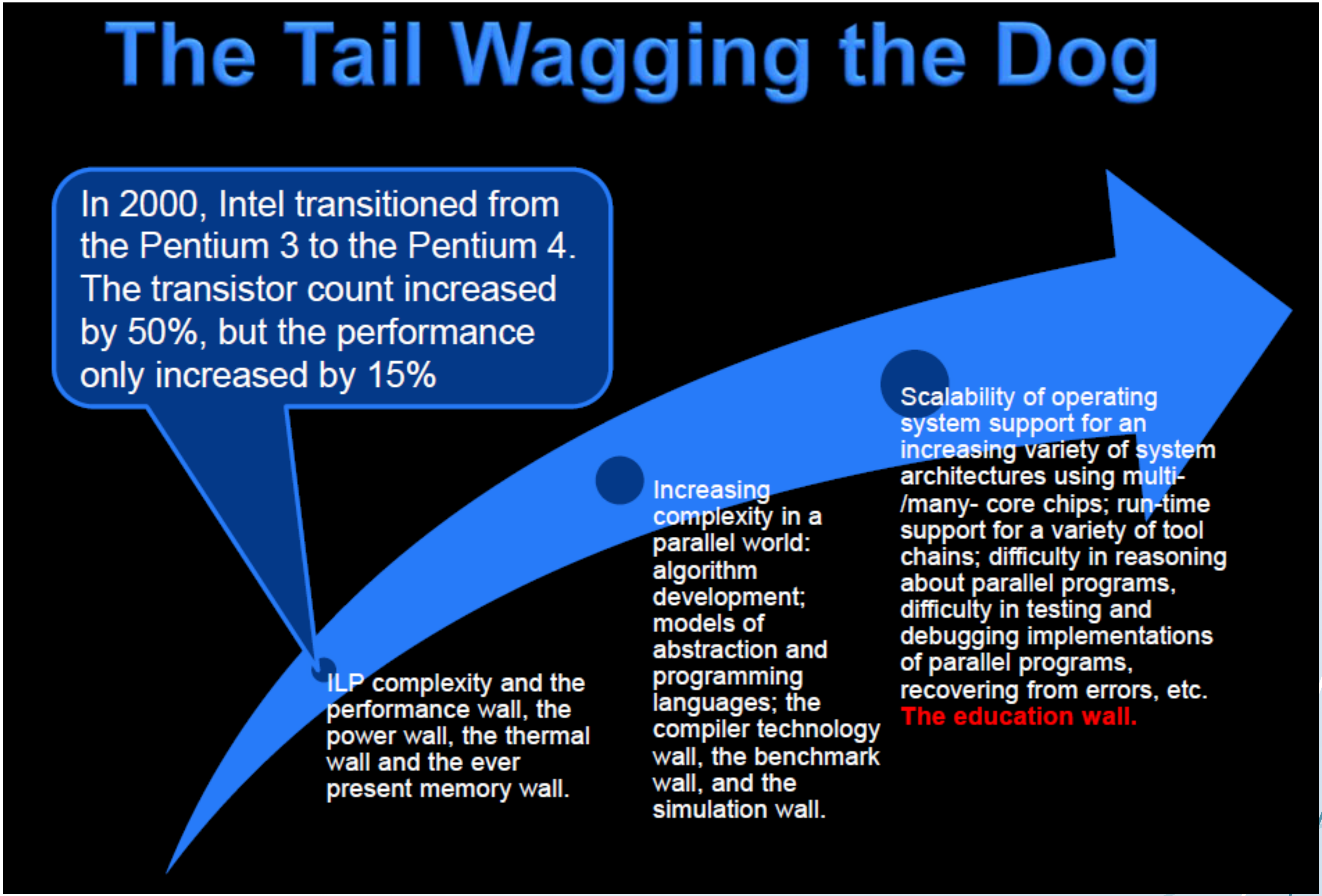


Figure from a lecture by Bruce Shriver, Feb 2012, UiT



The Tail Wagging the Dog

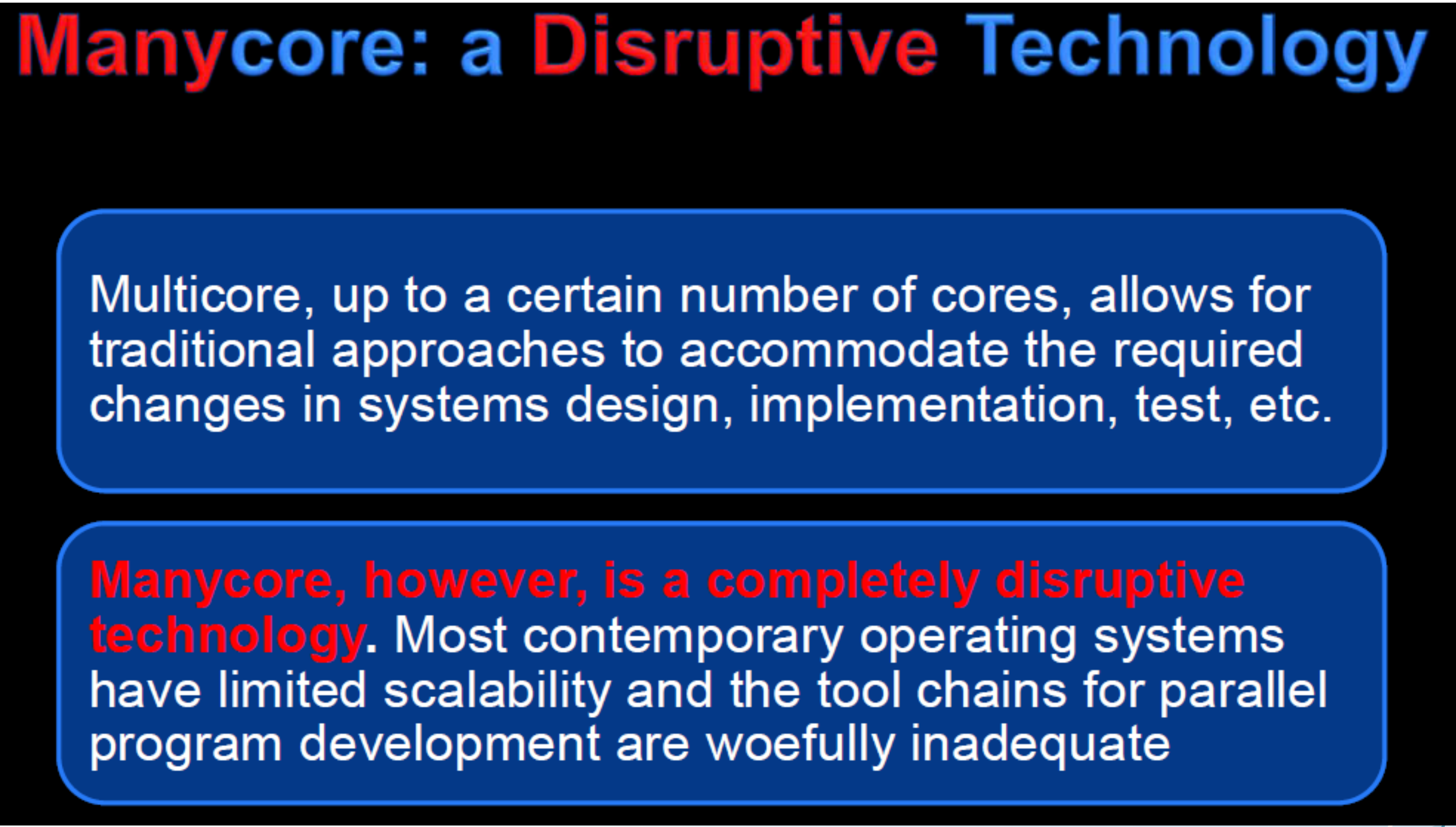
In 2000, Intel transitioned from the Pentium 3 to the Pentium 4. The transistor count increased by 50%, but the performance only increased by 15%

ILP complexity and the performance wall, the power wall, the thermal wall and the ever present memory wall.

Increasing complexity in a parallel world: algorithm development; models of abstraction and programming languages; the compiler technology wall, the benchmark wall, and the simulation wall.

Scalability of operating system support for an increasing variety of system architectures using multi- /many- core chips; run-time support for a variety of tool chains; difficulty in reasoning about parallel programs, difficulty in testing and debugging implementations of parallel programs, recovering from errors, etc. **The education wall.**

Slide from a lecture by Bruce Shriver, Feb 2012, UiT



Manycore: a Disruptive Technology

Multicore, up to a certain number of cores, allows for traditional approaches to accommodate the required changes in systems design, implementation, test, etc.

Manycore, however, is a completely disruptive technology. Most contemporary operating systems have limited scalability and the tool chains for parallel program development are woefully inadequate

Slide from a lecture by Bruce Shriver, Feb 2012, UiT

Motivation - summary

- Application pull
  - Supercomputing
  - Data-intensive computing
  - Games
  - Robotics
  - …
- Technology push
  - How to utilize 1000’s of cores?

Teaching staff

- Associate Prof. Lars Ailo Bongo
- Teaching Assistants:
  - Einar Holsbø
  - Tengel Ekrem Skare
- Guest lectures:
  - Lars Tiede (ITA, UiT)
  - Dag Brattli (Microsoft)
  - Åge Kvalnes (Microsoft)
  - Steffen Viken Valvåg (Microsoft)
  - Tor Kreutzer (Microsoft)
  - Jan-Ove Karlberg (Microsoft)
  - …?

Course content

- 3 main topics:
  - Concurrent programming
  - Data-intensive computing
  - Performance evaluation
- Not included:
  - In-depth study of systems and approaches
  - GPU/ accelerator programming
  - Concurrency theory
- No textbook

Information sources and contact info

- Web page: http://www.cs.uit.no/kursinfo/inf2202
  - Or https://inf-2202-f17.github.io/
- Mailing list: inf-2202-f17@list.uit.no
- Github organization: https://github.com/inf-2202-f17
- Slack team: inf-2202-f17
- We will not use Fronter/ whatever

TODO list 1:

- Make sure you are subscribed to the mailing list
  - https://list.uit.no/sympa/info/inf-2202-f17
  - Can use non-UiT email
- Create github account, send Einar username
- Join slack channel or ask Einar or Tengel for invitation
- Receive invitation to join github organization

Lecture plan

- https://inf-2202-f17.github.io/
- Most lectures on Thursdays 14:15-16:00
- Fridays 12:15-13:00 may also be used
- Colloquium on Tuesdays 14:15-16:00 (A016 or p-lab)

Mandatory assignments

1. Parallel programming
2. Reactive programming
3. Big data processing on cloud

Exercises

- Note! This is a programming course. You need to spend a significant amount of time designing, implementing, testing, and evaluating programs.

- Note! Concurrent and data-intensive programming is easy if:
  - It is a simple problem
  - There is a library for it
  - You do not care about performance/ scalability
  - You do not care about correctness
  - Someone tells you how to do it
- But, usually this is not the case

TODO list 2

- Read:
  - Modern operating systems, 3ed, Andrew S. Tanenbaum. Prentice Hall. 2007. Chapters: 2.2, 2.3, 2.5, 10.3, 11.4
  - Alternative to MOS: another operating systems textbook: the chapters about threading, IPC mechanisms, and classical IPC problems.
- Do:

  a) Compare the overhead of forking a process vs. creating a Pthread

  b) Compare the overhead of forking a process vs. creating a Python thread

  c) Implement a solution the following classical IPC problems using pthreads/Python threads and semaphores/condition variables. Note that you also need to generate a use case, test data, and useful output:

     a) Producer/ consumer

     b) Reader/ writer

     c) Sleeping barber

     d) Dining philosophers

  d) Modify the code in c) to use message passing.