Massively Parallel Algorithms
Organisational Stuff

G. Zachmann

University of Bremen, Germany

cgvr.cs.uni-bremen.de

What You (Hopefully) Get Out of This Course

- Most importantly: *mind set* for thinking about massively parallel algorithms
- Overview of some *fundamental* massively parallel algorithms
- Techniques for massively parallel *visual computing*
- Awareness of the *issues* (and solutions) when using massively parallel architectures
- Programming skills in *CUDA* (the language/compiler/frameworks for programming GPUs)

Is This Course For Me ???

- This course is not for you …
  - If you don’t like algorithms
  - If you are not ready to do a bit of programming in C
  - If you're not open to thinking about computing in completely new ways

Otherwise ...

- It will be a richly rewarding experience!

Website

- All important information about this course can be found on:

  http://cgvr.informatik.uni-bremen.de/

  → "Teaching" → "Massively Parallel Algorithms"
- Slides
- Assignments
- Text books, online literature
- Please sign up in StudIP!

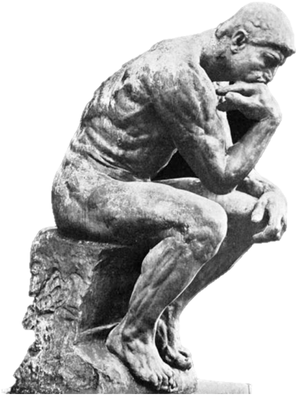
The Exam

1. Either: full oral exam (ca. ½ hour per student)
2. Or: grades from the exercices + mini oral exam ("Fachgespräch")
   - Exercises → grade A , mini oral exam → grade B
     - 95% of all points of the exercises → grade A = 1.0
     - 40% of all points of the exercises → grade A = 4.0
   - Overall grade = 0.5 × A + 0.5 × B
   - Uner the condition: grade A ≥ 4.0 && grade B ≥ 4.0 !
     (Allgemeiner Teil der Bachelorprüfungsordnungen der Universität Bremen, 2010)

- Grading criteria of the exercises:
  1. Labeling variable and function names
  2. "Sufficient" comments in body of functions
  3. Documentation of functions and their parameters (in/out, pre-/post-condition, what does the function do / not do, …)
  4. Functionality (exercise solved? no bugs? …)

Exercises / Assignments

- The two approaches we will pursue in this course:



- Weekly small exercises
  - Due the week after assignment
- Optional: your own programming mini-project in CUDA
  - Due in the last lecture!
  - You give the demo …
  - Before you begin, you need to present your idea in 5 minutes

The SDK, Needed for Working at Home

- IDE (obviously) of your choice
  - Can be as simple as an ASCII editor and compiler on command line
- CUDA for your platform:

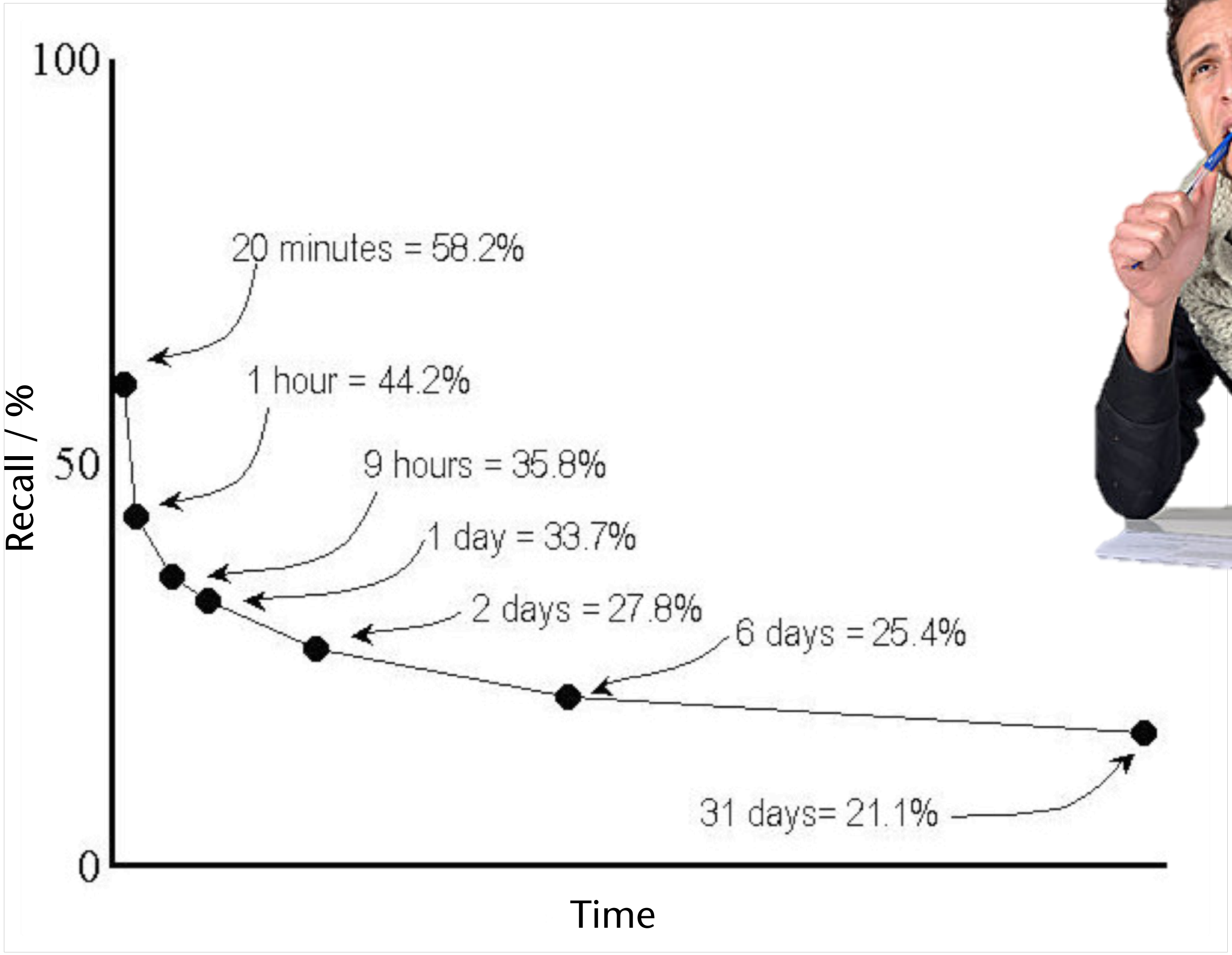
  https://developer.nvidia.com/cuda-downloads
  - Works, of course, only with NVidia graphics cards
  - If your laptop/desktop does not contain NVidia, use the pool or our lab

A Quote

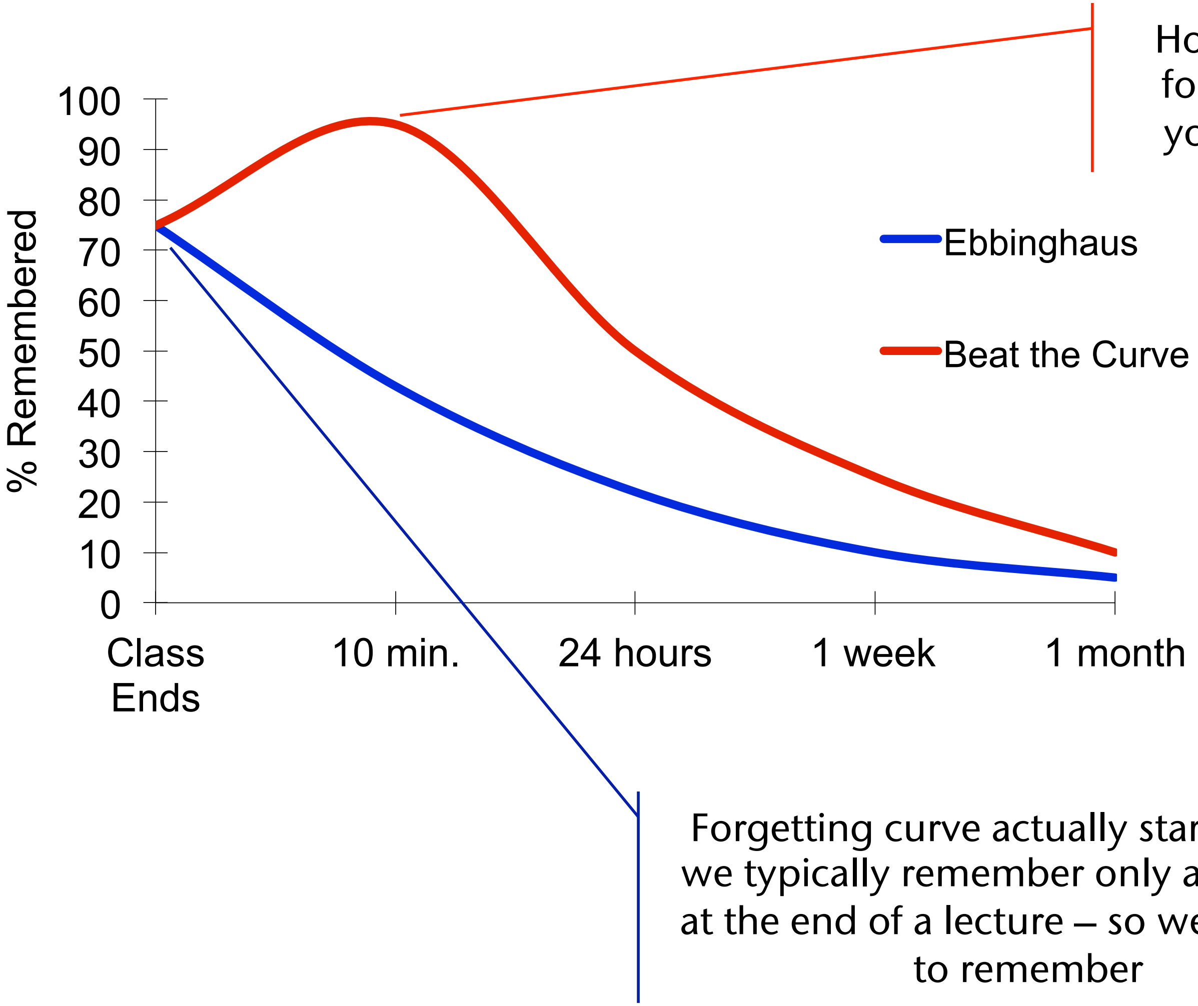
I hear and I forget.
I see and I remember.
I do and I understand.

[attributed to Confucius]

The Forgetting Curve (Ebbinghaus)



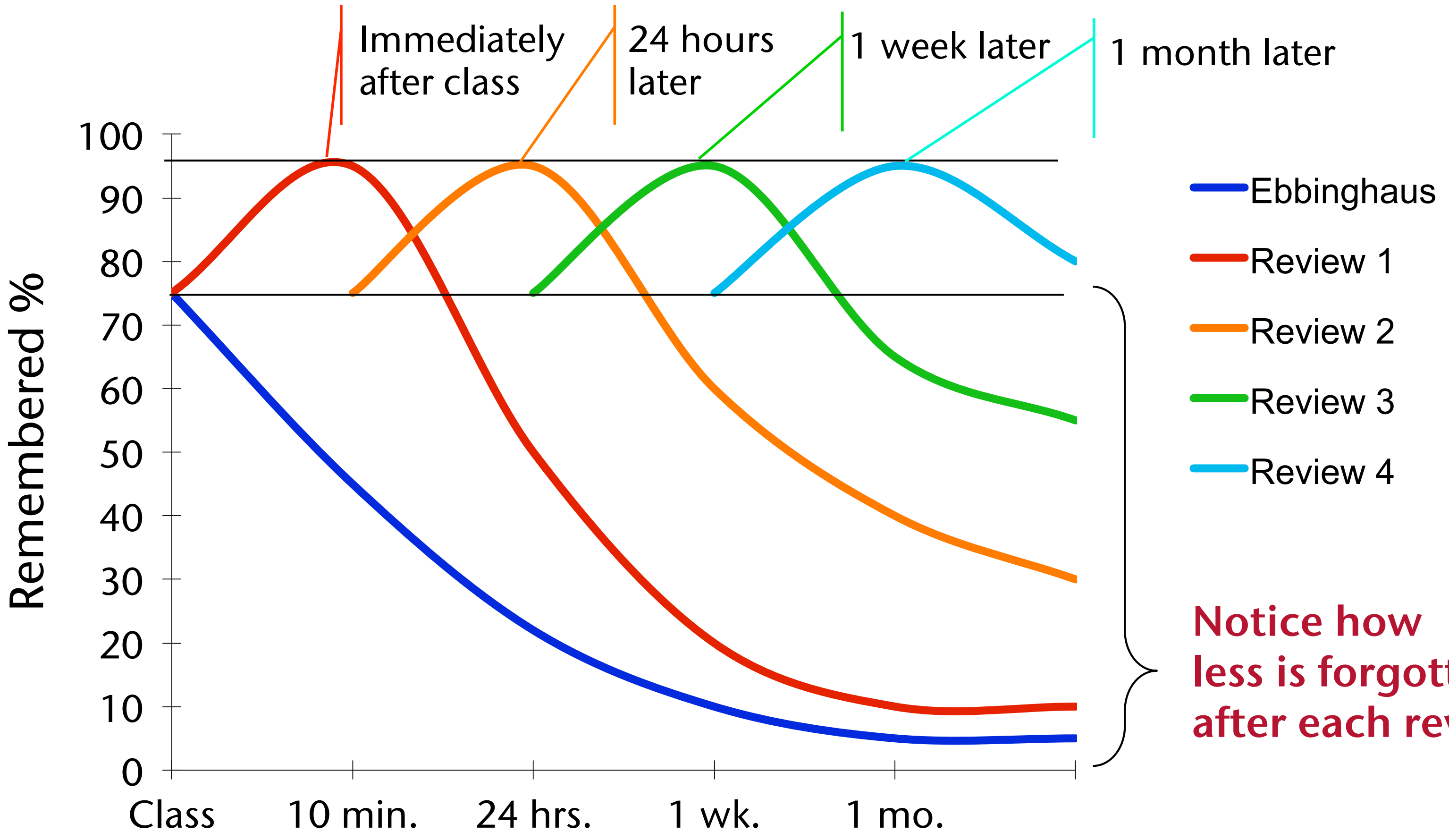
Beating the Forgetting Curve



However, you have the potential to forget less PLUS remember more if you review immediately after class

Forgetting curve actually starts here as we typically remember only about 75% at the end of a lecture – so we have less to remember

Overcoming the Curve



Notice how less is forgotten after each review!

Average Retention Rates

Just listening	5%
Reading	10%
Audio Visual	20%
Demonstration	30%
Discussion	50%
Practice by doing	75%
Teach others	90%

What Lies Ahead (Tentative)