Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

(?:Build|Develop) your own Database

Week 1

Outlook

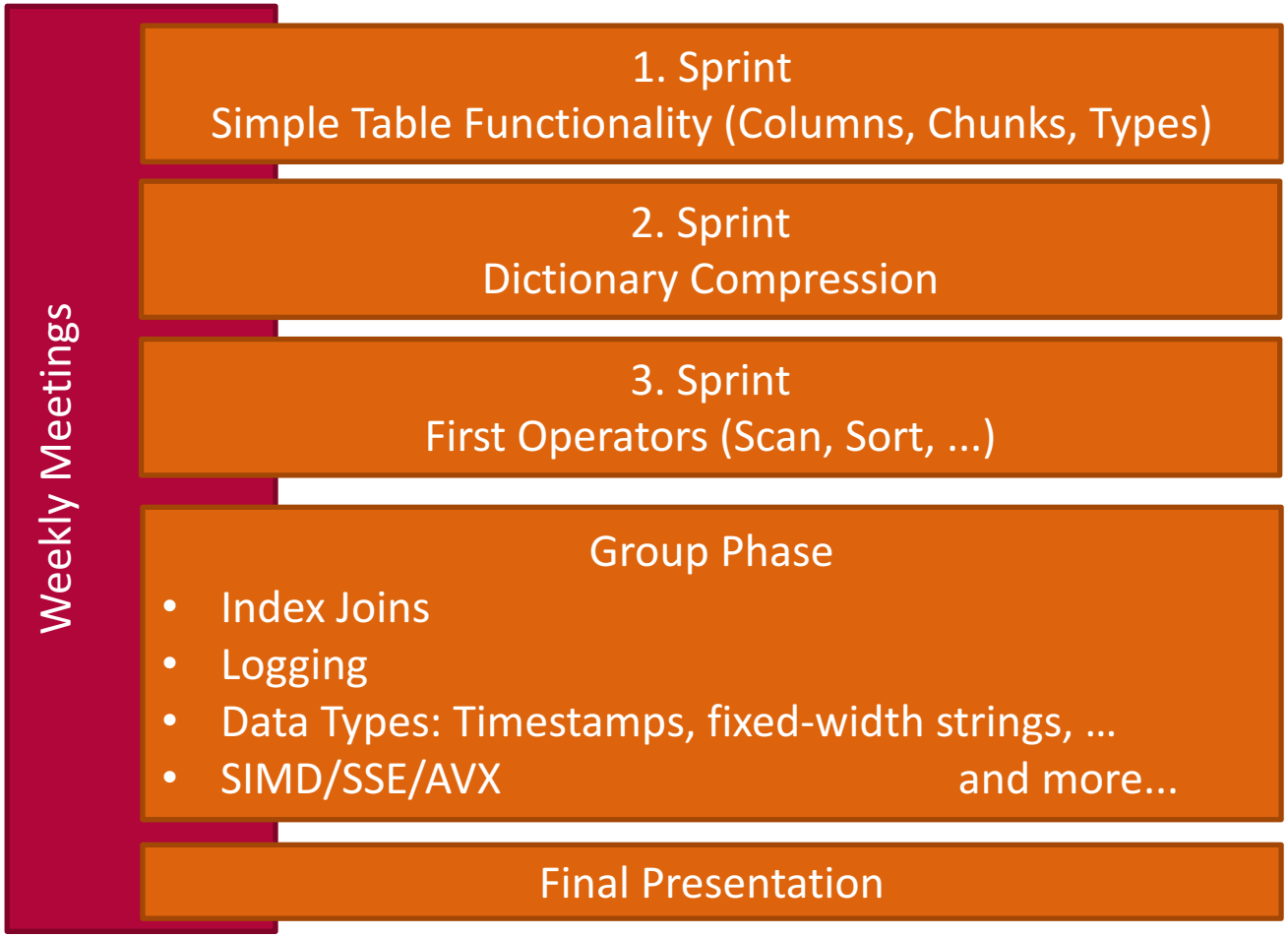
1. High-Level Overview
2. First Work Package
3. Organizational Stuff

What can you expect?

- Better understand how in-memory databases work
- Gain experience in system development
- Improve your C++1xyz skills
- Work in small teams on a larger project

If this sounds interesting to you, you are in the right room.

Timeline


Weekly Meetings
1. Sprint
Simple Table Functionality (Columns, Chunks, Types)
2. Sprint
Dictionary Compression
3. Sprint
First Operators (Scan, Sort, ...)
Group Phase
• Index Joins
• Logging
• Data Types: Timestamps, fixed-width strings, …
• SIMD/SSE/AVX
and more...
Final Presentation

What do we expect?

- Fruitful discussions about why we do things the way we do
- Active participation in the group work and our meetings

What do we hope for?

1. Generate interest in our research
2. Continue to work with you in Master‘s theses, Hiwi jobs, ...
3. ???
4. Profit.

Who are we?

Martin Boissier

- Data Aging and Tiering
- Pricing

Markus Dreseler

- New Hardware
  - NVRAM
  - SGI

Stefan Klauck

- Replication
- SSI-CLOPS

Jan Kossmann

- Database Automatization

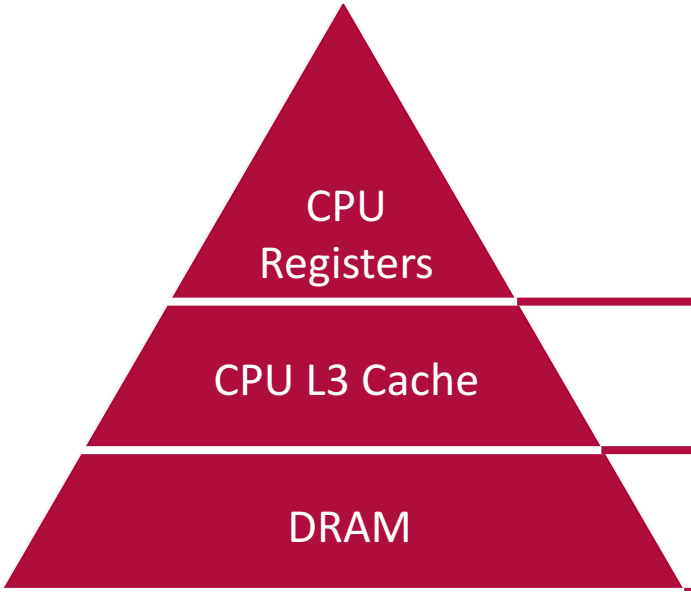
Introducing Opossum

Introducing Opossum

- Opossum is the (1) prototypical, (2) columnar (3) in-memory database that we will build in this class

- Prototypical: We do not plan for Opossum to be used in a productive environment
- Columnar: We exclusively use columnar orientation for data
- In-Memory: All data that we work with is stored in RAM

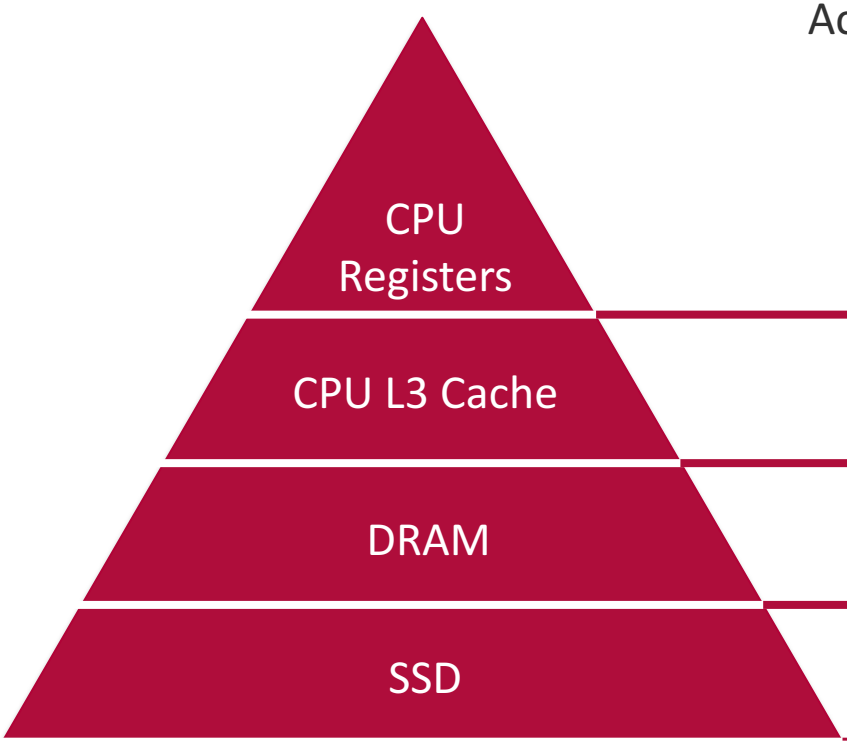
Why In-Memory?



	Access Time	If a nanosecond was a second, similar to retrieving the data from...
CPU Registers	1 ns	your memory (1 sec)
CPU L3 Cache	10 ns	your desk (10 sec)
DRAM	100 ns	the next office (01:40 min)

http://people.eecs.berkeley.edu/~rcs/research/interactive_latency.html
http://www.montana.edu/cpa/news/wwwpb-archives/yuth/pigeon.html

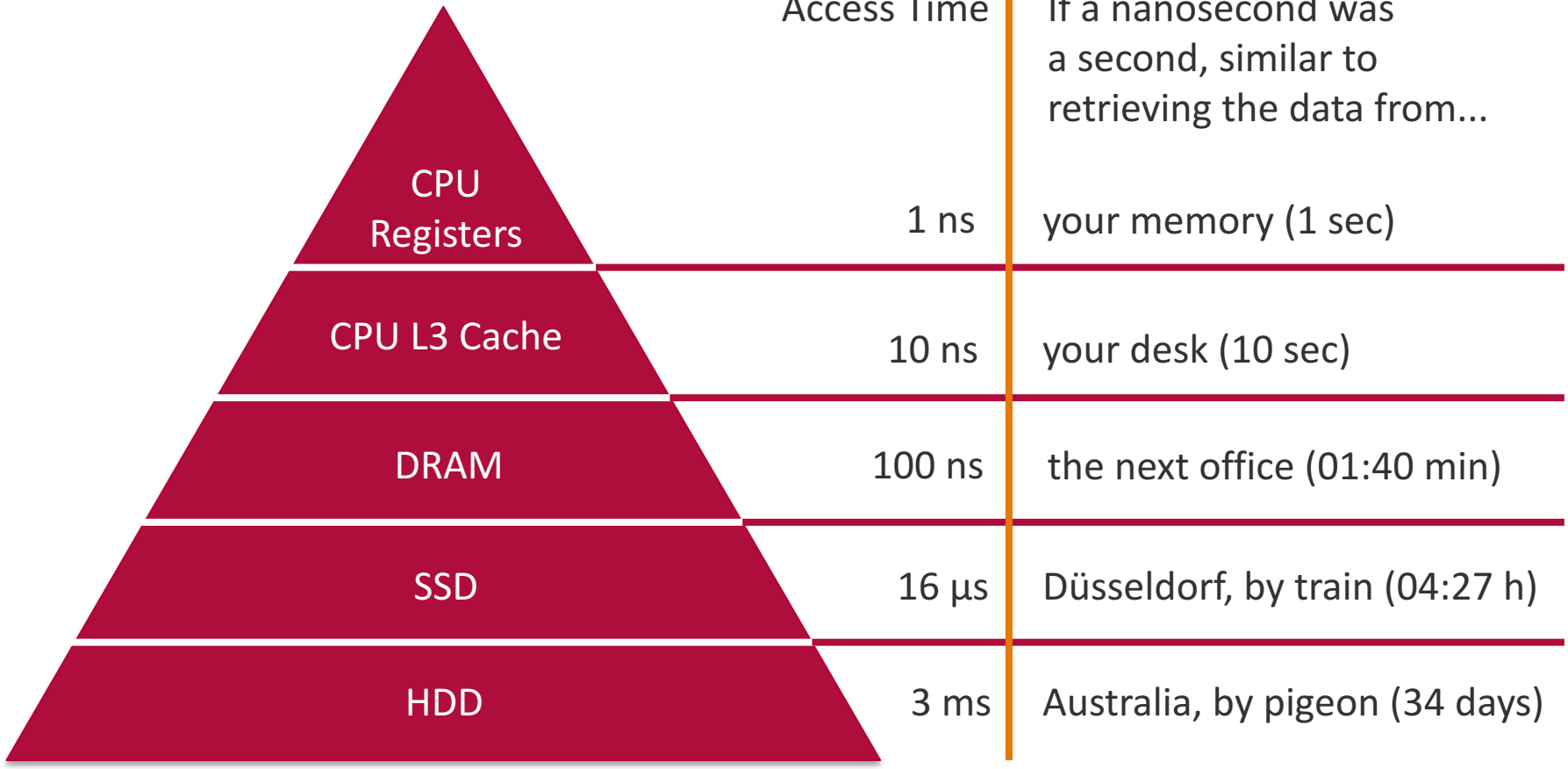
Why In-Memory?



	Access Time	If a nanosecond was a second, similar to retrieving the data from...
CPU Registers	1 ns	your memory (1 sec)
CPU L3 Cache	10 ns	your desk (10 sec)
DRAM	100 ns	the next office (01:40 min)
SSD	16 µs	Düsseldorf, by train (04:27 h)

http://people.eecs.berkeley.edu/~rcs/research/interactive_latency.html
http://www.montana.edu/cpa/news/wwwpb-archives/yuth/pigeon.html

Why In-Memory?



http://people.eecs.berkeley.edu/~rcs/research/interactive_latency.html
http://www.montana.edu/cpa/news/wwwpb-archives/yuth/pigeon.html

HYRISE?

- You might have heard of our research database Hyrise
- The database started last year and was extended by student projects
- It will replace Hyrise[1] on Thursday
  - Hyrise[2] is significantly easier to understand, extend, and use
- Allowing students to start from scratch makes the first steps easier

First Work Package

Description

- You can find the description of the work package online:
  - https://hpi.de/plattner/teaching/winter-term-201718/build-your-own-database.html

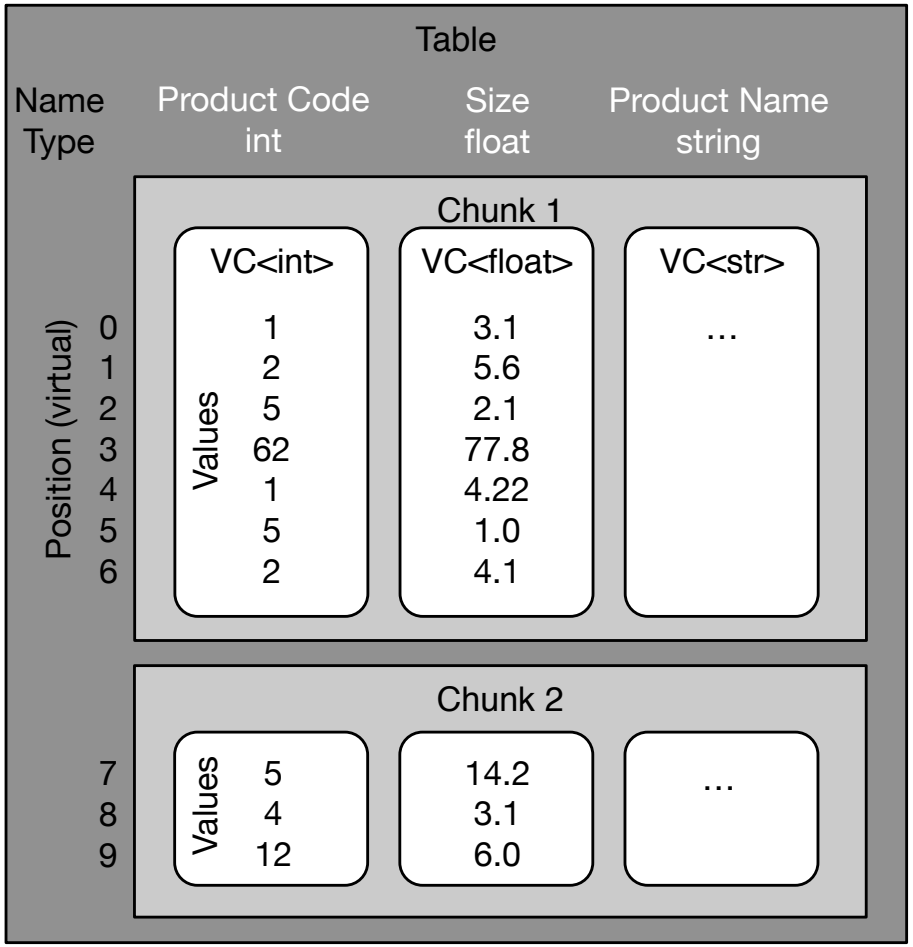
First tasks

1. Set up your build environment
2. Implement a single column
3. Group columns into a chunk
4. Append data to a chunk
5. Group chunks into a table
6. Store tables in a StorageManager

Setting up your Environment

- Demo (git clone, install, cmake, make test -j)

The Opossum Table Model



VC: ValueColumn

Document Walkthrough

Organizational Stuff

About Correctness

- Instead of reusing last year‘s material, we have stripped down the Hyrise codebase to the template that we are giving you
- We have tested that everything works the way we expect it to, but this does not mean that everything is perfect
- If something looks wrong, or if you have any issues about the course itself, please do not hesitate to talk to us

Einschreibung und -fristen, Leistungserfassungsprozess, Vertiefungsgebieteinordnung

Allgemeine Information

- Semesterwochenstunden : 4
- ECTS : 6
- Benotet : Ja
- Einschreibefrist : 27.10.2017
- Programm : IT-Systems Engineering MA
- Lehrform : PS
- Belegungsart : Wahlpflicht

Module

- ITSE-{Analyse, Entwurf, Konstruktion, Maintenance}
- BPET-{Konzepte und Methoden, Spezialisierung, Techniken und Werkzeuge}
- OSIS-{Konzepte und Methoden, Spezialisierung, Techniken und Werkzeuge}
- SAMT-{Konzepte und Methoden, Spezialisierung, Techniken und Werkzeuge}

Einschreibung und -fristen, Leistungserfassungsprozess, Vertiefungsgebieteinordnung

Kriterium	Gewichtung
Sprint 1-3	30 %
Gruppenphase	60 %
Aktive Mitarbeit	10 %

Piazza

- Most likely, there will be remaining questions about the architecture or the implementation
- Waiting for a week is not an option
- Your classmates may have the same question or be able to help you

Piazza

- We use Piazza as an online plattform for all of these discussions:
- https://piazza.com/class/j8vgbo26s8g689
- Please use common sense in how much of your implementation you should share

Groups

- We would like for you to work in groups of two or three, depending on how many students are in the class
- Please send an email to markus.dreseler@hpi.de with your group members and the name of your git repository by the date of the Einschreibefrist
- You can also use Piazza to find team members

Weekly Meetings

- We will use one of our two slots for presentations given (mostly) by the teaching team – attendance here is highly recommended
- The other slot can be used for your group work
- As needed, we might use it for further clarifications of the material – this will be announced in advance and is optional
- Which slot do you prefer? Vote on Piazza!

Deliverables

- 1 Nov — Code Sprint 1
- 8 Nov — Review Sprint 1

- 15 Nov — Code Sprint 2 (tbc)
- 22 Nov — Review Sprint 2
- 29 Nov — Code Sprint 3
- 6 Dec — Review Sprint 3

(Group phase)

- 7 Feb — First Code Group Phase
- tbd — Review and Final Code Group Phase

Next Week

- Deep Dive into some of the used C++ concepts and beyond
  - Templates
  - Smart Pointers
  - RAII