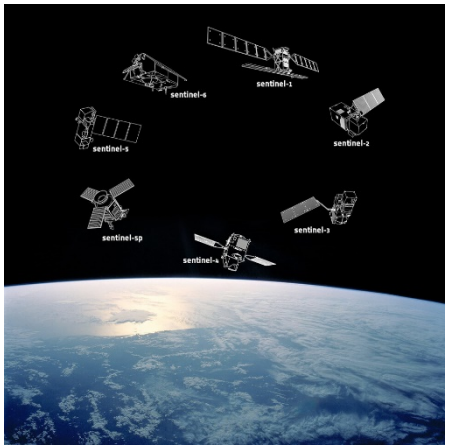
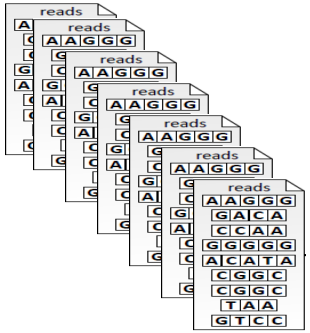
Workflows for Scientific Data Analysis

Ulf Leser

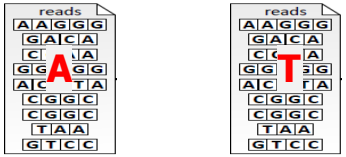
Big Scientific Data

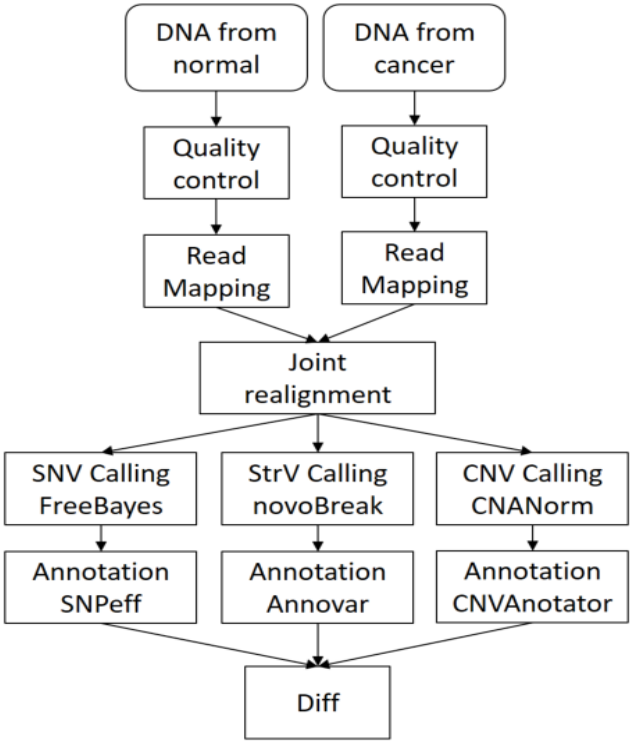


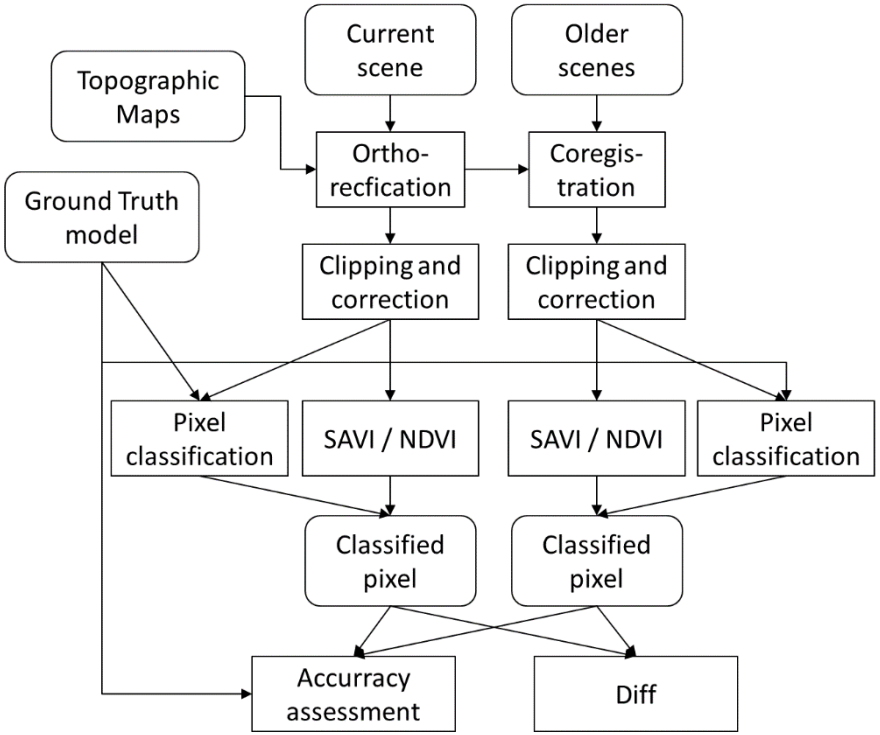


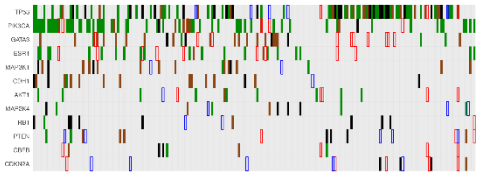
Note petabytes every day, but easily a few terabytes per week

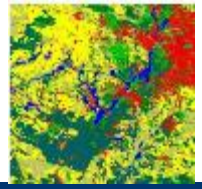
Data Analysis Workflows (DAWs)



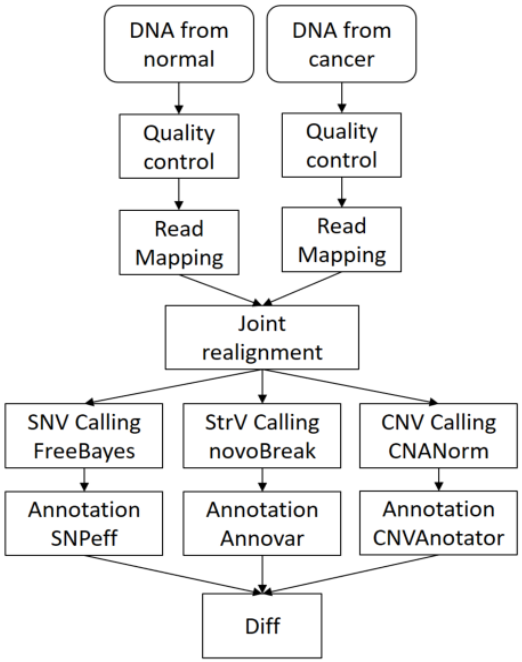








Distributed DAW Infrastructure



slurm workload manager

SAASFEE

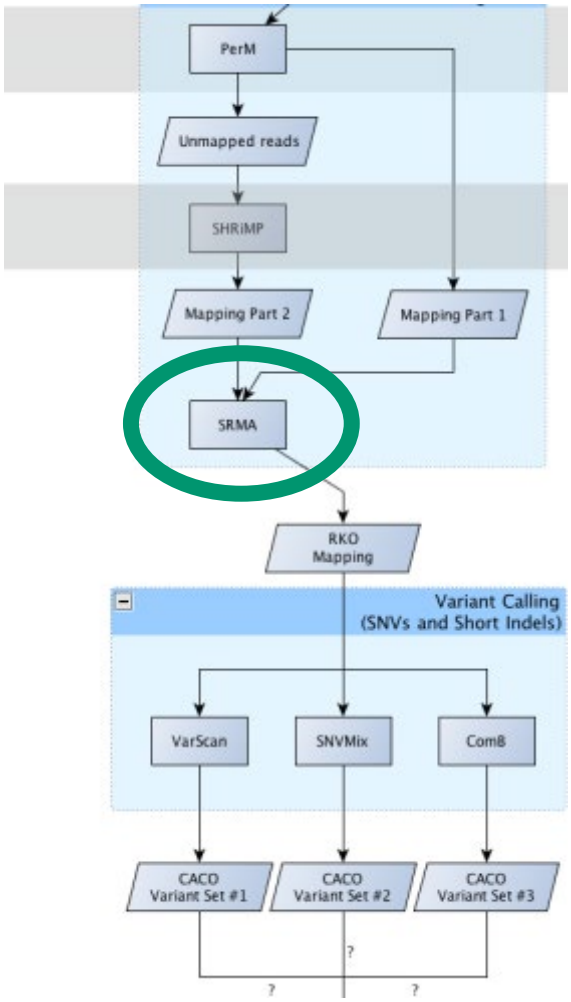
APACHE hadoop

APACHE Spark™

DAWs

Tasks

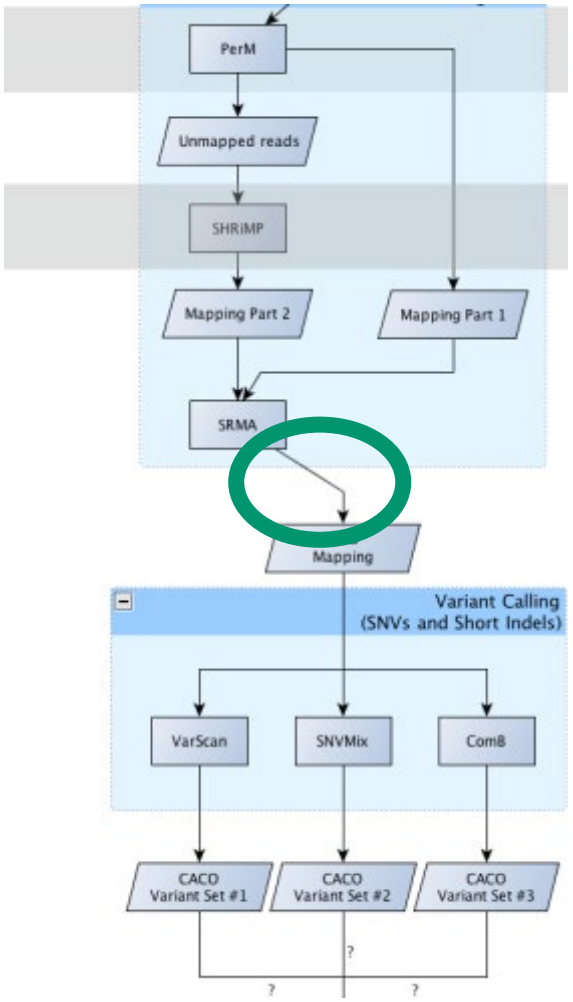
- Have (multiple) input and output "ports" (parameter)
- Must be executable
  - Or web services – deprecated in the Big Data area
- Black box: Infrastructure has no notion on what a task does



DAWs: Tasks and Dependencies

Tasks

- Have (multiple) input and output "ports" (parameter)
- Must be executable
  - Or web services – deprecated in the Big Data area
- Black box: Infrastructure has no notion on what a task does

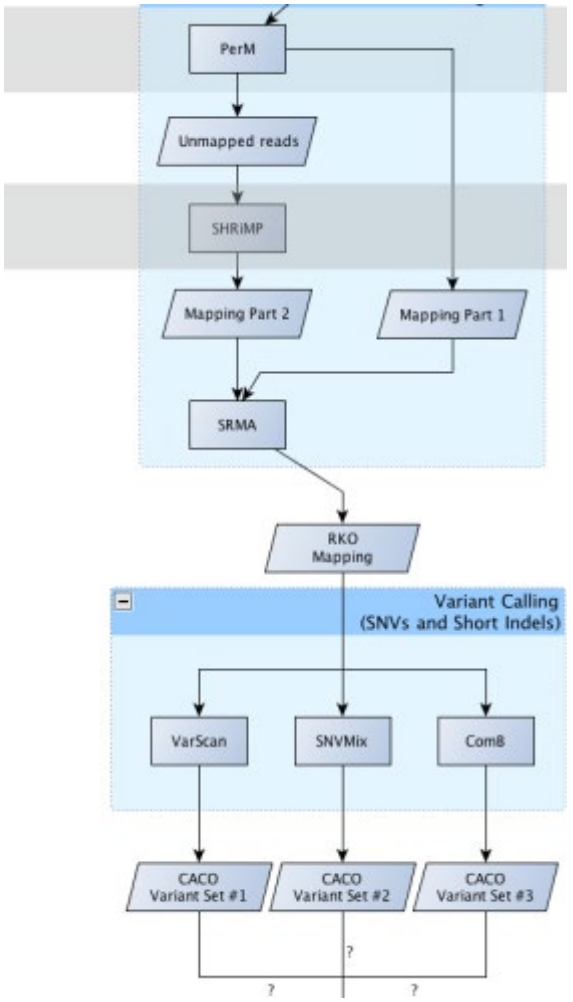


Dependencies

- Connect one input (upstream) with one output (downstream) port
- Implemented as files, pipes, in-memory, …
- Constrain the possible order of execution
- Black box: Infrastructure has no notion on content (or format)

Dependency Graph

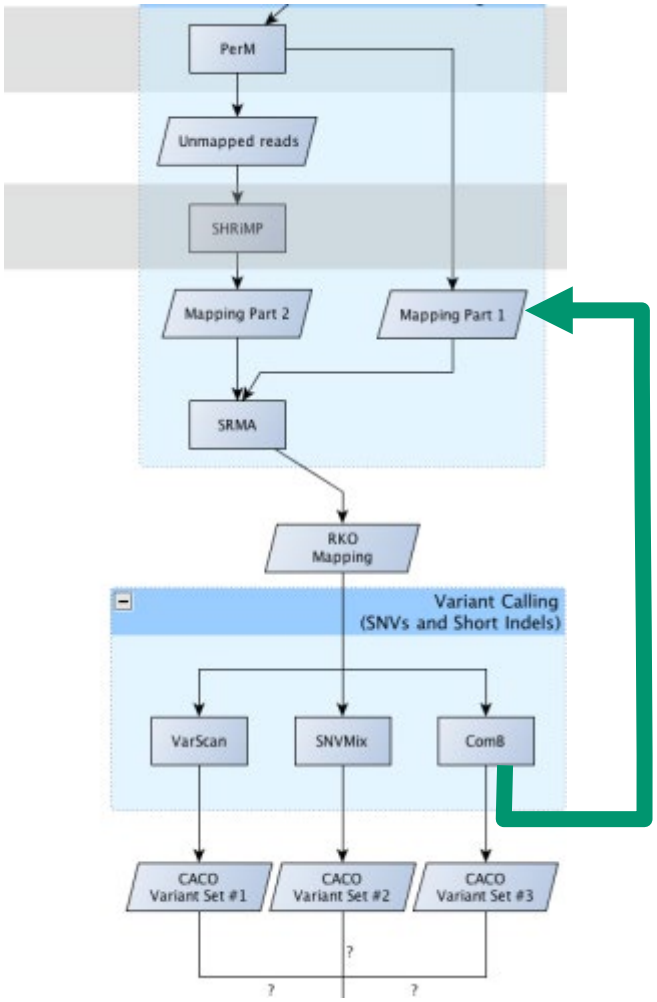
- Tasks become nodes (vertices)
- Dependencies become edges (arcs)
- Together a directed graph G = (T,D)



Dependency Graph

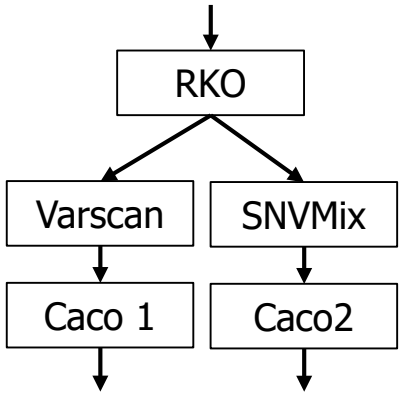
- Tasks become nodes (vertices)
- Dependencies become edges (arcs)
- Together a directed graph G = (T,D)

- Mostly a simple graph: No two arcs between the same pair of nodes
  - But: A task T1 may produce two files F1, F2 which both are input for task T2
- Mostly not a hyper-graph
  - But: Broadcasts can be modelled as n-ary arcs
- Mostly a DAG
  - But: Iteration / recursion introduces loops

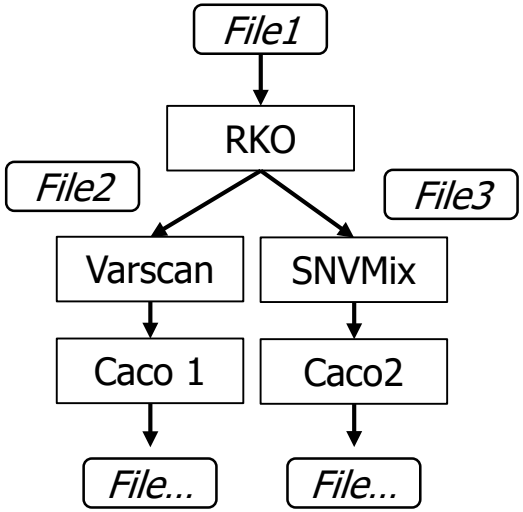


Abstract versus logical DAWs

- So far, our DAWs were abstract
  - Nodes have names, but there are no concrete files
  - Cannot be executed, but are easy to understand

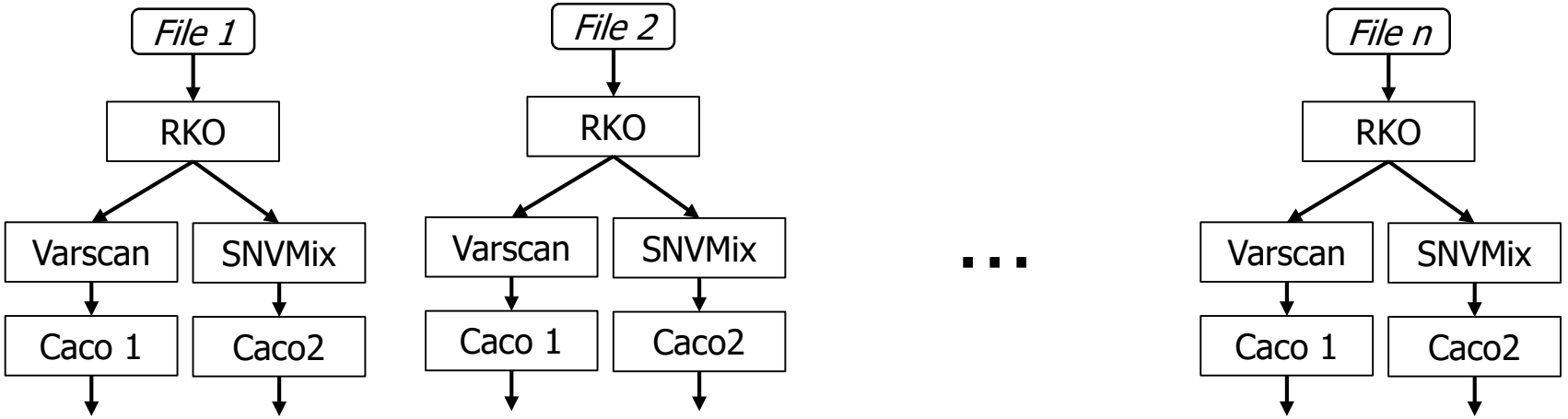


- A logical DAW also has concrete input files and produces concrete output files
  - A logical DAW instantiates an abstract DAW



Real Life

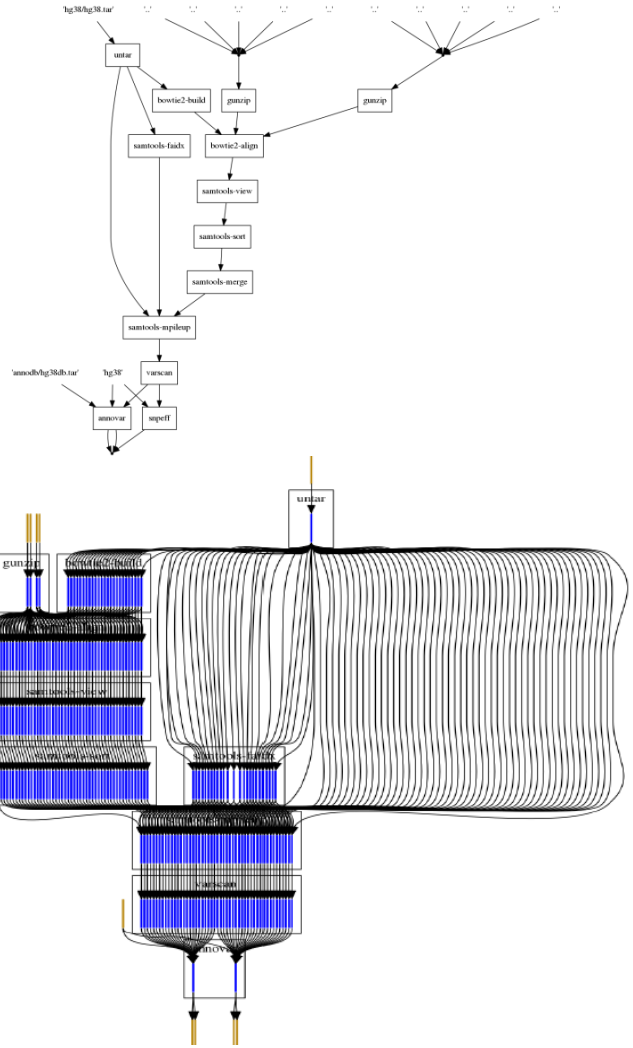
- Typically, DAWs are executed (from start or intermediate) over many files



- With multiple inputs, a single abstract (partial) DAW produces many logical (partial) DAWs
- More complicated as one may think, as the multiplicity of intermediate result files is only determined at runtime

Real Life with Real DAWs

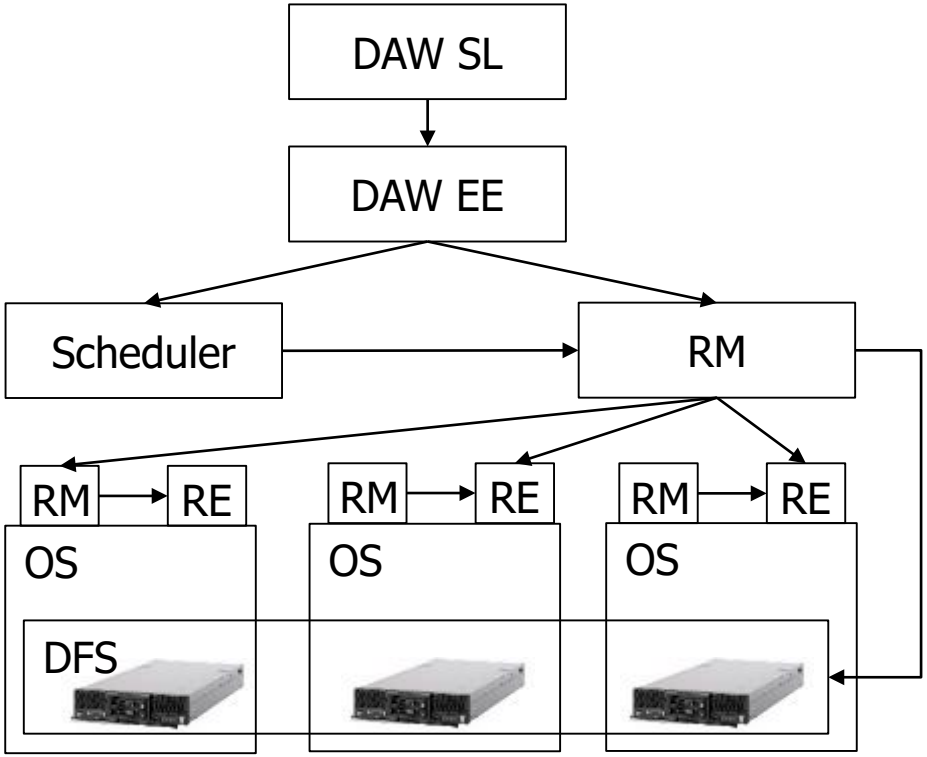
- Multiplicities
  - Some tasks read one file and produce one
    - Read image and produce normalized image
  - Some tasks read one file and produce many
    - Read large image and partition into smaller ones
  - Some tasks read many files and produce one
    - Read partitions and combine into one file



Physical DAWs

- A physical DAW is a logical DAW plus
  - Every (logical) task is assigned to a node for execution
  - Every dependency is assigned to a method of communication
- Logical DAW + schedule + data exchange = physical DAW

DAW Components



Who should be here

- Mono- / Kombibachelor Informatik
- Ability to read English papers
- Knowledge in
  - Distributed systems (e.g. scheduling, file systems)
  - Data science (applications)
  - Software engineering (e.g. languages, optimization)
- Willingness to independent work
  - Search suitable papers covering a topic, prepare presentations, write seminar thesis

How it will work

- Today: Presentation and choice of topics
  - If desired, we will group teams of 2 students
- 1.6.2021: Send an outline of your topic (next slide)
- ~10.6.21: Present your topic in 5min flash-presentation
- ~1.7.21: Meet your advisor to discuss slides
- ~10.7.21: Present your topic (30min) in a Blockseminar
- 30.8.2021: Write seminar thesis (10-15 pages)

The outline

- Topics are rather abstract
- Find a set of suitable papers covering the topic
  - Focus is allowed and welcome
- Extract the most important information
- Structure into an outline of your thesis
  - Chapters, Sections, 1-2 sentences per section to describe the content, 2-4 figures
- Write an abstract
  - Roughly 20 lines – what is the topic, what will the thesis describe?
- Send me abstract + outline + references

The 5-min flash talk

- Focus on marketing – drag students into your topic
  - What is the topic?
  - Why is it challenging?
  - Why is it cool?
  - What are important applications?
  - What will your talk be about?
- At most 5 slides
- Focus on figures & examples; omit all details or algorithms

Presentation

- 30min presentation
- German or English
- Explain topic, methods, experimental results
- Compare different approaches (if enough time)
- Aim: Your audience should understand what you say
- No need to cover the topic entirely – select best topics

Teams

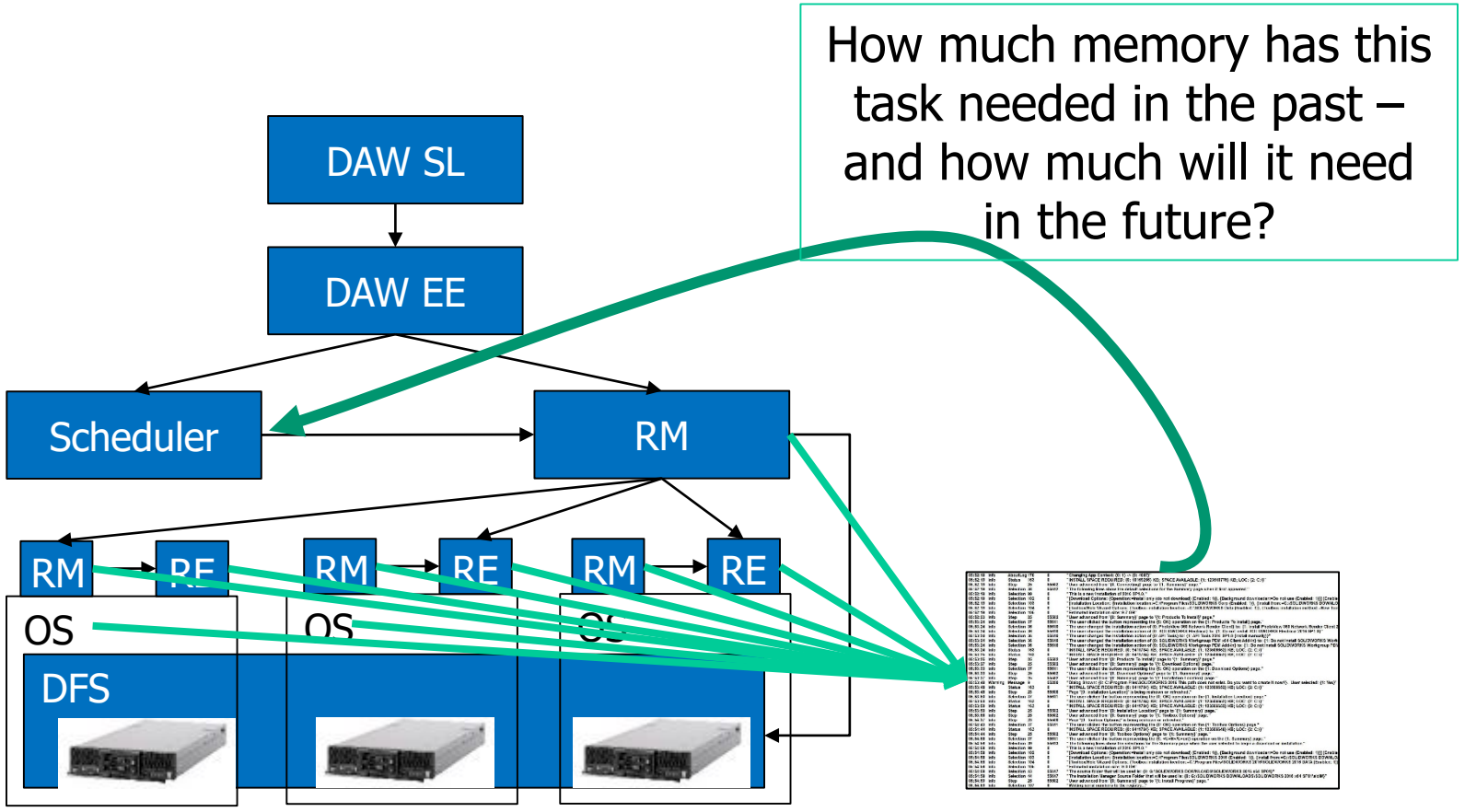
- If a topic is addressed by a team of two students, I expect
  - Read more papers
  - Have more topics in your outline and thesis
  - Presentations times remain the same – choose wisely

ToC

- Introduction
- Topics
- Assignment
- Hints on presenting your topic and writing your thesis

Topic	Assigned to
Provenance Management and Analysis	
Scheduling of Scientific Workflows	
Declarative Workflow Languages	
Container Technology for SWF (Docker, Singularity)	
Distributed File Systems and SWF (CEPH, HDFS, ...)	
Resource Managament for SWF (Yarn, Mesos, ...)	
Concrete system: NextFLow	
Concrete system: Apache AirFlow	
Concrete system: sankemake	
Concrete system: CWL	
Concrete system: Apache TEZ	

DAW Components

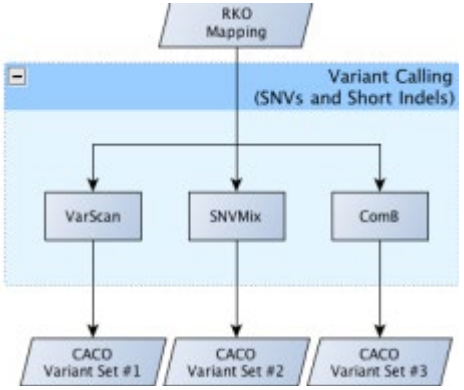


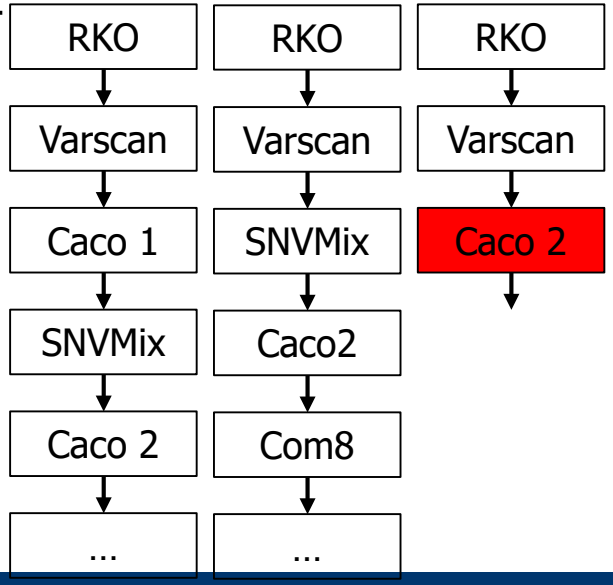
Provenance

- Provenance is all information and their temporal order used to derive some analysis result
  - Input files, intermediate files, user interactions, parameter, configuration & binary of tasks, scheduling decisions, resource manager decision, OS events, …
- Provenance management: Systems for obtaining, storing, and analyzing provenance information
- Provenance can become extremely large
- Typical PM systems constrain the information being managed

Scheduling and Execution Orders

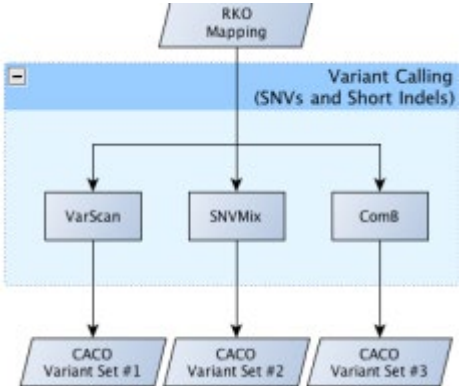
- Executing a physical DAW means: Executing each task exactly once
- Execution order: Order in which tasks are executed
  - For n tasks, there are n! orders
- But: Every dependency is a constraint
  - T1→T2: T2 must not be executed before T1
- One a single node / thread
  - Every execution order must be linear
  - Valid orders: All topological sorts of G

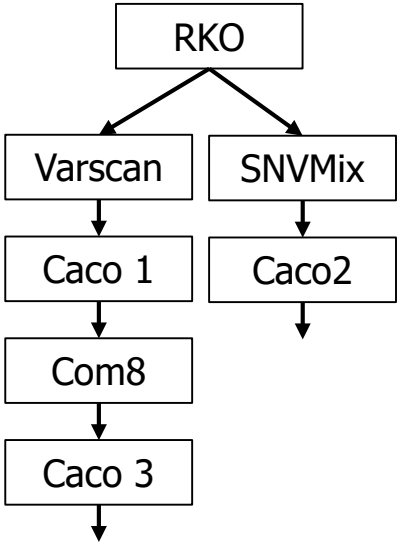




Execution Orders

- Executing a DAW means: Executing each task exactly once
- Execution order: Order in which tasks are executed
  - For n tasks, there are n! orders
- But: Every dependency is a constraint
  - T1→T2: T2 must not be executed before T1
- One a single node / thread
  - Every execution order must be linear
  - Valid orders: All topological sorts of G
- With many nodes / threads
  - Scheduling
  - Import: Number of nodes / threads





Flavors of DAW languages

- Every DAW must be specified – specification language
- Many exist, which fall in four classes
  - Graphical
  - Imperative
  - Declarative
  - Functional
  - Formal process models
  - Embedded
  - Domain-Specific

Declarative / Functional DAW Languages

Declarative

- Consist of goals, preconditions, and facts – rules
  - To fulfill a goal, its preconditions must be fulfilled, which contain goals … recursion down to facts
  - You need "build-in" goals to do real action – read, compute, …
- Compilation is equivalent to building a proof tree
- Examples: CWL, YAWL

- Not easy to grasp if no prior knowledge in logic programming languages, claims higher maintainability

Functional

- Based on Lamba calculus – first and second order functions
  - Like functional programming languages: Scala, Lisp, Haskell
- Compilation is similar as in declarative languages
- Examples: Cuneiform, Swift

- Niche systems with enthusiastic supporters

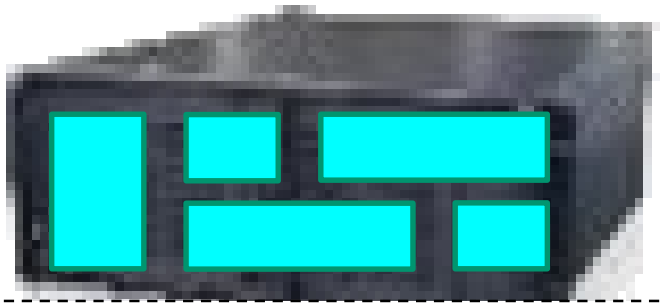
Cluster

- A cluster is an interconnected set of computers
- Ambiguous concept: Node
  - A node typically has multiple cores, threads, hyperthreads
  - Beware: "Node" often used for "computer" and for "smallest compute unit"
- Every node has resources
  - CPU (runtime)
  - Memory
  - Bandwidth
  - Other: GPU, …
  - Beware: Nodes may share resources (threads share memory, all processes on one machine share GPUs, …)

- A cluster is homogeneous if all nodes have same resources
  - Many clusters are not

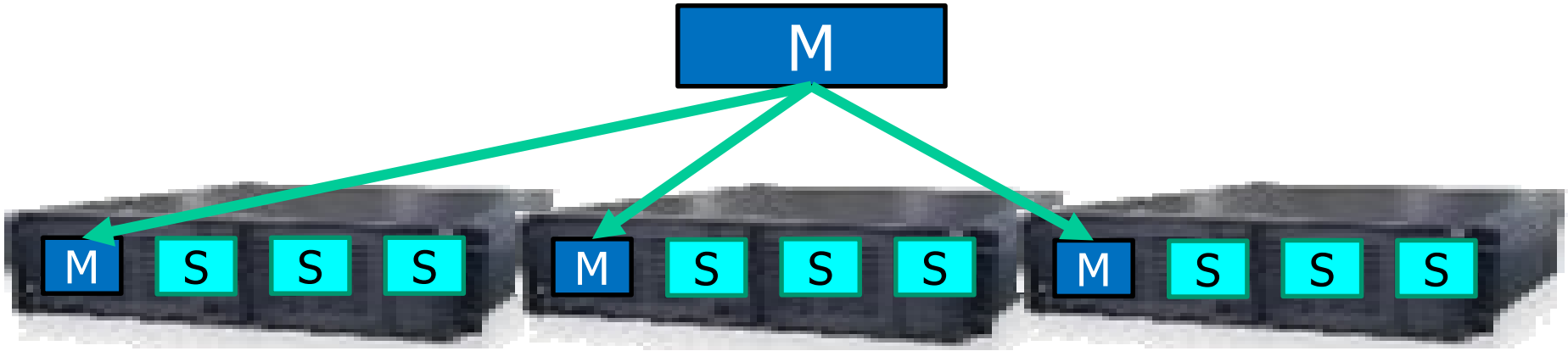
Virtualization

- Virtualization: Simulation of many independent compute nodes on one hardware node
  - Different degrees of separation: Classical virtual machines, containers, …
- DAW systems today typically orchestrate only virtualized nodes
- VMs can be tailored to a task's need
  - Virtualized clusters are very often heterogeneous



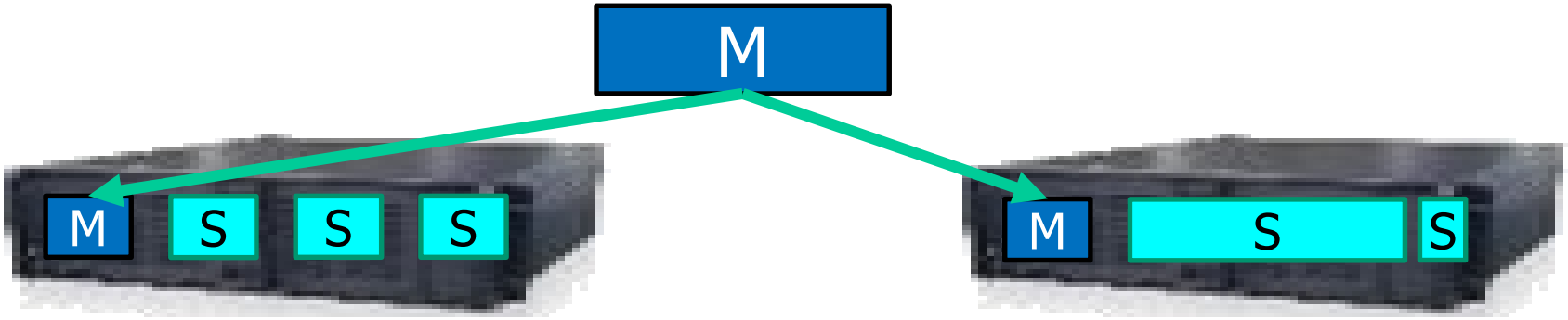
Kurtzer, G. M., Sochat, V., & Bauer, M. W. (2017). Singularity: Scientific containers for mobility of compute. *PloS one*, *12*(5), e0177459.

Resource Manager



- Every node needs a resource manager
  - Creates and destroy VMs
  - Determines resources per VM
  - Grants access to VMs
  - Negotiates with host OS
- Cluster resource manager: Coordination of per-node RMs
  - "Where can I get a VM with 500GB main memory?"

Partitioning a Node

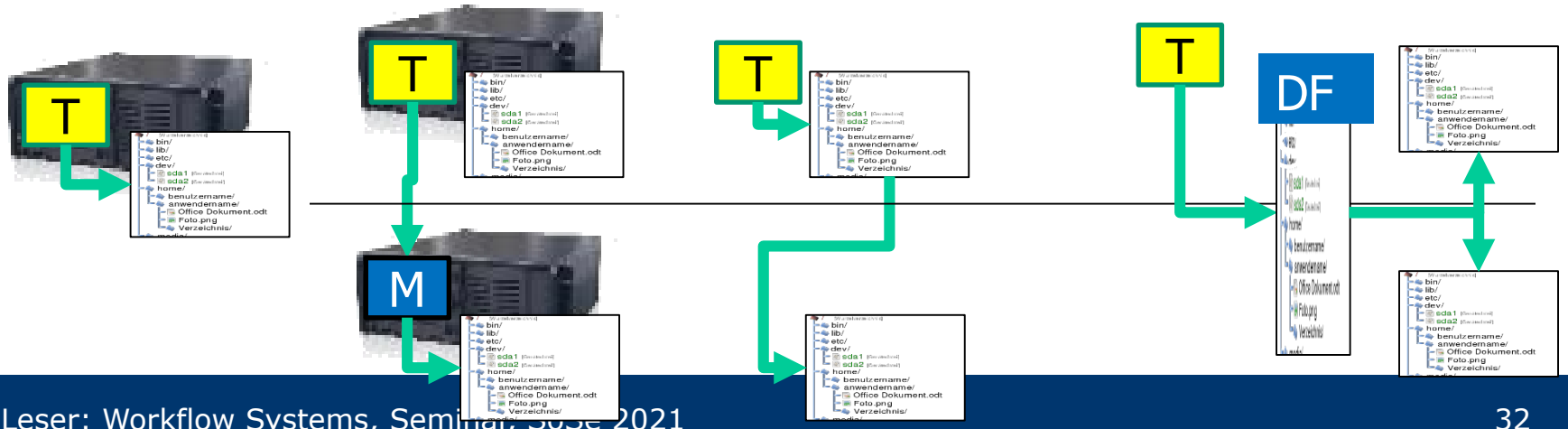


- RM enforces homogeneous VMs
  - Resources equally partitioned
  - Preconfigured, pre-started
  - Optimal saturation of node possible
  - Simpler administration
  - Tasks may be impossible to execute
- RM allows heterogeneous VMs
  - Resources are partitioned on request at DAW runtime
  - VMs created at runtime
  - May lead to underutilization of node
  - Complex administration
  - "All" tasks are possible

AWS solution: A small set of different types of VMs (large, small, huge, …)

File Systems – where is "somewhere"

- Tasks communicate via network or via files
- Files are stored in a file system
  - Local: Disks/SSD are on same hardware node as compute node and administered by local OS
  - Remote: Discs are on remote hardware node
- Task managers might copy from remote hosts from RM
- Tasks may read from remote hosts via, e.g., NFS
- More comfortable: Distributed (virtualized) file systems



Five concrete Systems

NextFlow	Groovy DSL	Quickly growing, much bioinformatics
AirFlow	Python Workflow DAGs	Much commercial usage, origin AirBnB
Tez	Java Workflow DAGs	Unclear popularity, origin Yarn
Snakemake	Python / Make – style	Popular in bioinformatics
CWL	Proper language	

- All are open source
- Most work with different back-ends or stand-alone
- Topic also requires programming concrete workflows and measuring parallel / distributed execution
  - Hello world; word count; table-scan and grouping; kmer count

ToC

- Introduction
- Topics
- Assignment
- Hints on presenting your topic and writing your thesis

Allgemeine Hinweise

- Dozenten sind ansprechbar!
  - Vorbesprechung des Themas
  - Folien durchgehen
  - Abgrenzung der Ausarbeitung
- Diskussion erwünscht
  - Keine Angst vor Fragen: Fragen sind keine Kritik
  - Eine Frage nicht beantworten können ist in Ordnung
- Tiefe, nicht Breite
  - Lieber das Thema einengen und dafür Details erklären
- Bezug nehmen
  - Vergleich zu anderen Arbeiten (im Seminar)

Allgemeine Hinweise

- Werten und bewerten
  - Keine Angst vor nicht ganz zutreffenden Aussagen – solange gute Gründe vorhanden sind
  - Begründen und argumentieren
  - Kritikloses Abschreiben ist fehl am Platz
- Literaturrecherche ist notwendig
  - Die ausgegebenen Arbeiten sind Anker
  - Weiterführende Arbeiten müssen herangezogen werden
  - Auch Grundlagen nachlesen

- Wir schicken eine Liste zum Abhaken rum

Wie halte ich einen Seminarvortrag

1. Wenn man nun so einen Seminarvortrag halten muss, dann empfiehlt es sich, möglichst lange Sätze auf die Folien zu schreiben, damit die Zuhörer nach dem Vortrag aus den Folienkopien noch wissen, was man eigentlich gesagt hat.

2. Während so einem Vortrag schaut sowieso jeder zum Projektor, also kann man das selbst ruhig auch tun - damit kontrolliert man gleichzeitig auch, ob der Beamer wirklich alles projeziert, was auf dem Laptop zu sehen ist. Ausserdem kann man so den Strom für das Laptop-Display sparen.

3. Übersichtsfolien am Anfang sind langweilig, enthalten keinen Inhalt und nehmen den Zuhörern die ganze Spannung. Schliesslich gibt‘s im Kino am Anfang auch keine Inhaltsangabe.

4. Powerpoint kann viele lustige Effekte, hat tolle Designs und Animationen. Die sollte man zur Auflockerung des Vortrags unbedingt alle benutzen, um zu zeigen, wie gut man das Tool im Griff hat.

5. Nicht zu wenig auf die Folien schreiben. Man weiß ja nie, ob man sie nicht doch ausdrucken muss, und man kann so wertvolle Zeit sparen, wenn man nicht weiterschalten muss.

6. Man sollte versuchen, möglichst lange zu reden. Die Zeitvorgaben sind nur für die Leute, die nicht genug wissen - eigentlich will der Prüfer sehen, dass man sich auch darüber hinaus mit dem Thema beschäftigt hat.

Bloß keine Hervorhebungen im Text – sonst müssen die Zuhörer ja gar nicht mehr aufpassen!

Hinweise zum Vortrag

- ~30 Minuten inkl Diskussion
- Klare Gliederung
- Ab und an Hinweise geben, wo man sich befindet
- Bilder und Grafiken; Beispiele
- Font: mind. 16pt
- Eher Stichwörter als lange Sätze
- Vorträge können auch unterhaltend sein
  - Gimmicks, Rhythmuswechsel, Einbeziehen der Zuhörer, etc.
- Adressat sind alle Teilnehmer, nicht nur die Betreuer
- Technik: Laptop? Powerpoint?

Hinweise zur Ausarbeitung

- Eine gedruckte Version abgeben
  - Selbstständigkeitserklärung unterschreiben
- Eine elektronische Version schicken
- Referenzen: Alle verwendeten und nur die
  - Im Text referenzieren, Liste am Schluss
- Korrekt zitieren
  - Vorsicht vor Übernahme von kompletten Textpassagen; wenn, dann deutlich kennzeichnen
  - Aussagen mit Evidenz oder Verweis auf Literatur versehen
- Verwendung von gefundenen Arbeiten im Web
  - Möglich, aber VORSICHT
  - Eventuell Themenschwerpunkt verschieben – Betreuer fragen

Format

- Benutzung unserer Latex-Vorlage
- Nur eine Schriftart, wenig und konsistente Wechsel in Schriftgröße und –stärke
- Inhaltsverzeichnis
- Bilder: Nummerieren und darauf verweisen
- Referenzen:
  - [1] Yan, X., Yu, P. S. and Han, J. (2004). "Graph Indexing: A Frequent Structure-Based Approach". SIGMOD, Paris, France.
  - [YYH04] Yan, X., Yu, P. S. and Han, J. (2004). "Graph Indexing: A Frequent Structure-Based Approach". SIGMOD, Paris, France.
- Darf man Wikipedia zitieren?
  - Ja, aber nicht dauernd

Hinweise zur Ausarbeitung –2-

- Gezielt und sachlich schreiben
  - Ausführungen zur „Philosophische Überlegungen zu Vorzügen probabilistischer Verfahren im Vergleich zu Dempster‘s Theory of Evidence“ oder zur „Anmerkungen zur Trivialisierung des politischen Diskurs für soziale Netzwerke unter besonderer Berücksichtigung von Twitter“ möglichst kurz halten
  - Füllwörter vermeiden (dabei, hierbei, dann, …)
  - Knappe Darlegung, präzise Sprache
- Eine gute Gliederung ist die halbe Miete
- Kommen Sie zu Aussagen
  - Vorteile, Nachteile, verwandte Arbeiten, mögliche Erweiterungen, Anwendbarkeit, eigene Erfahrungen, …