

Project Large-scale Data Engineering (LDE) Kick-off Meeting

Dr.-Ing. Patrick Damme

Technische Universität Berlin Berlin Institute for the Foundations of Learning and Data Big Data Engineering (DAMS Lab)





Announcements/Org

- Hybrid Setting with Optional Attendance
 - In-person in MAR 0.010
 - Virtual via zoom

https://tu-berlin.zoom.us/j/67376691490?pwd=NmlvWTM5VUVWRjU0UGI2bXhBVkxzQT09



ZOOM



About Me

Since 10/2022: Postdoc at TU Berlin, Germany

- FG Big Data Engineering (DAMS Lab) headed by Prof. Matthias Böhm
- Continuing work on integrated data analysis pipelines
- Research interests in the fields of database and ML systems (especially compiler & runtime techniques, extensibility)

• 2021-2022: Postdoc at TU Graz & Know-Center GmbH, Austria

- Data Management group headed by Prof. Matthias Böhm
- Started work on integrated data analysis pipelines

2015-2020: PhD student at TU Dresden, Germany

- Dresden Database Research Group headed by Prof. Wolfgang Lehner
- PhD thesis on making complex analytical database queries more efficient through lightweight compression of intermediate results













FG Big Data Engineering (DAMS Lab) – Teaching

Successfully Established TUB Teaching Portfolio (modules, slides)







Agenda



- Course Organization, Outline, and Deliverables
- Projects in DAPHNE and Apache SystemDS
- How to Approach the Project
- List of Project Topics (Proposals)





Course Organization, Outline, and Deliverables





 \rightarrow In the context of systems for data engineering, data management, machine learning

 \rightarrow In combination: Ideal preparation for a bachelor/master thesis with our group

Course Organization



General Contact Person

Dr.-Ing. Patrick Damme (<u>patrick.damme@tu-berlin.de</u>)

Course Website

- https://pdamme.github.io/teaching/2025_summer/lde/lde_summer2025.html
- One site for seminar and project
- All material, schedule, deadlines

ISIS course

- https://isis.tu-berlin.de/course/view.php?id=41969
- Announcements, discussion forum, topic selection poll, submission of summary paper and presentation slides

Language

- Lectures and slides: English
- Communication: English/German
- Submitted paper and presentation: English
- Informal language (first name is fine), immediate feedback is welcome



Semester Schedule & Deadlines

- Kick-off Meeting Apr 14 (optional)
- Recommended Introductory Lecture (optional)
 - May 05, 14:00: Experiments, Reproducibility, and Giving Presentations
- Self-organized Project Work
 - Consultation hours for any questions (optional)
- Intermediate Presentations (prerequisite)
 - Jun 30, 16:00-18:00, B 106: All teams/individuals
- Final Presentations (mandatory)
 - Aug 04, 14:00-18:00, FH 301: All teams/individuals



- List of Project Topics
 - Presented today, take your time to select afterwards

Topic Selection

- Deadline: May 02, 23:59 CEST (in 2½ weeks)
- Ranked list of 5 topics via poll on the ISIS course
 + pref on individual/team work [+ team members]
- Global topic assignment based on preferences
- Notification of assigned topics: May 12 (in 4 weeks)
- Submission of Initial Prototype (prerequisite)
 - Implementation and tests
 - Deadline: Jun 29, 23:59 CEST (in 11 weeks)
 - As a pull request on GitHub (exceptionally by email)
- Submission of Final Prototype (mandatory)
 - Implementation, tests, docs, experiments
 - Deadline: Jul 28, 23:59 CEST (in 15 weeks)
 - As a pull request on GitHub (exceptionally by email)
- Submission of Pres. Slides (Intermediate & Final Pres.)
 - Deadline: The day before the presentation, 23:59 CEST
 - Upload PDF in the ISIS course

Project Deliverables: Initial Prototype & Intermediate Presentation

Introduced in Response to Students' Feedback (Course Evaluation)

Initial Prototype

- 80% functionally complete prototype including good set of test cases
- Basis for further improvements driven by experiments and feedback

Intermediate Presentation

- Slide presentation of 5-10 min per individual/team
- Briefly present the problem you work on
- Give an overview of your initial prototype (concepts and crucial changes to the code base)
- Outline your planned experiments
- Should be the result of prior discussions with your project mentor

Benefits for You

- Improved time management (retain enough time for experiments)
- Exchange with the other students in the project
- Get feedback by project mentors and other students for improving the quality of your prototype.





Ungraded Prerequisites for the Portfolio Exam

to be allowed make mistakes and learn from them

Project Deliverables: Final Prototype & Final Presentation



Final Prototype

- 100% functionally complete prototype including good set of test cases
- Efficiency (confirmed by experiments)
- Final Presentation
 - Summarize the problem and give an overview of your final prototype
 - Present your experimental results
 - 1 student: 10 min talk + 5 min discussion = 15 min
 - 2 students: 13 min + 7 min = 20 min
 - 3 students: 16 min + 9 min = 25 min
 - Audience: engage in the discussion

Grading

- #41086 (seminar + project)
 - Graded portfolio exam
 - 25 pts: summary paper
 - 15 pts: presentation
 - 50 pts: design/impl/tests/doc
 - 10 pts: presentation
- #41183 (project-only)
 - Graded portfolio exam
 - 85 pts: implementation/tests/documentation
 - 15 pts: presentation
- Academic Honesty / No Plagiarism

implies that use of LLMs like ChatGPT is prohibited



Portfolio Exam Registration



- Portfolio exam registration: May 12 Jun 09
 - Binding registration in Moses/MTS
 - Including selection of seminar presentation date (first-come-first-serve)

Portfolio exam de-registration

- Until 3 days before the first graded exam part
 - Modules "LDE"/"Seminar LDE": until Jun 20
 - Module "Project LDE": until Jul 25
 - De-register yourself in Moses/MTS
- With sufficient reason: Until the day of the exam
 - In case of sickness etc.
 - Modules "LDE"/"Seminar LDE": until Jun 23/Jul 07/Jul 14
 - Module "Project LDE": until Jul 28/Aug 04

- Missing deadlines/exam without de-registration
 - Zero points in the respective exam part (!)
 - Approach us early in case of problems
- If you don't want to take LDE anymore
 - Let me know asap to give students in the queue a chance to fill in



LDE Project Goals and Mindset



Goals

- Design/implement a prototype in DAPHNE/SystemDS
- AND prove it is a valuable contribution to the system (tests, documentation, experiments)
- Present and defend your work in a presentation & discussion
- Focus on Methodology
 - LDE as a preparation for a bachelor/master thesis at the DAMS Lab
- Mindset: Be Open Learn New Things and to Work on Any Part of the System
 - Whatever it takes to fulfill the task
 - Self-guided acquisition of required technical skills

Grading Criteria

- Design/implementation (functionality)
- Code quality + tests + documentation
- Experiments

High-quality and convincing contribution to an open-source system





Projects in DAPHNE and Apache SystemDS



Overview: Two Systems Developed at the DAMS Lab



- An Open and Extensible System Infrastructure for Integrated Data Analysis Pipelines
 - Intersection of data management, machine learning, high-performance computing
 - Open-source (Apache v2 license) <u>https://github.com/daphne-eu/daphne</u>
 - Originated from DAPHNE EU-project
 - Written mostly in C++, Python, DaphneDSL
 - Since 2020 (open-source since 2022)



[Patrick Damme et al.: DAPHNE: An Open and Extensible System Infrastructure for Integrated Data Analysis Pipelines. CIDR 2022]



- A Declarative ML System for the End-to-End Data Science Lifecycle
 - Data integration/cleaning/prep, model selection/ training/validation/debugging/deployment/scoring
 - Open-source (Apache v2 license) <u>https://github.com/apache/systemds</u>
 - Originated from Apache SystemML (started at IBM)
 - Written mostly in Java, Python, and DML
 - Since 2010 (open-source since 2015) (as SystemML)



[Matthias Boehm et al.: SystemDS: A Declarative Machine Learning System for the End-to-End Data Science Lifecycle. CIDR 2020]



Simplified High-level Architecture of DAPHNE and SystemDS







Simplified High-level Architecture of DAPHNE and SystemDS



Optimizations

(e.g., IPA,

rewrites, operator

ordering, operator

selection, codegen

Parameter

Server

Spark

Inst.

Feder

ated

Inst.



Output Data



Kinds of LDE Project Topics



BIFOI

LDE Project Characteristics



- Individual/Team Project Work
 - Teams of up to 3 students strongly encouraged
 - Unique topic for each individual/team

Ambitious Projects

- 9 ECTS (~270 h of work)
- ≈6.75 weeks of full-time work

Potential for Impact

- Real open issues in existing systems
- If successful: meaningful contributions that will be used by others

Remarks on Topic Descriptions

- Many open topics in DAPHNE and SystemDS
- Initial topic descriptions of varying level of detail
- During topic selection: Approach project mentor directly if interested in more details
- After topic assignment: More detailed descriptions where necessary
- We're open to alternative topic proposals





How to Approach the Project



Getting Started: Setting Up Your Development Environment



- Goals
 - Build the system from source
 - Successfully run the test suite
- Navigate to the GitHub Repos



Know Where to Find the Documentation



Clone, Build, Test According to the Documentation

1	Quickstart for Developers	Fable of contents
lanner i	Follow these instructions if you work to more modification to CMPHR in series of the source code, local, complex, holding in discovered atom. These emply alogs aloud indice to get standarfor work developers. If regardled, proceeded for modifier to motion weight alor to the sourced	Desense to Uney 1. Earl Earling 2. Rev 2000 00 Typics 1. Nac EMPRE a tra- Tomane
	1. Clone the DAIN-INE Source Code Repository	Pagnaria Managerina.
	Close the same code of the man DV2HRC reportury (command below) or your own fork (adapt the command below at resonancy)	Not light
	ger close letter i certain conduzion numation ger	2. Dome the DATH ME Source Date Reporting 2. Domenia the DATH ME
	2. Download the DAPHNE Development Container Image	5 Mat 504-68 and fair the Text Tark
	The development container image already contains all necessary (1) dependencies of a 2003ML development review over as well as a world indexicution of reviewment available etc., such that and whit reading works advect thread thread on these a service investors are dependence on the service advector of the set thread on the set of the se	Next Dage Addresse Details for Castern Senger
	Get the container image	Spalars Begannerich Operating spalars
	State and Antone Barrier der Gabert 314-94.000	Vision Vision
	Het A case of Docker permission evens, to preparely justice to the command.	Hardware.
	Hill From many want to choose another change tag datent an your platform and meets, in g. [constr_min_int_min_from flow (PPC suggest) or [constr_mining man [for 2006 suggest].	Building the GATE ME agreement Defining up the remainment
	Here () Encountries and Pyllinicit are optional for displaced and act included in the inspira- ray contained due to their higher of anomal gapdyrine. Please Solar the enclosure on installing Python iterates in the inspiration on contained proceedings iteration.	Kareng tin Trads Kareng Schinkt Kahtro untificienzy with

https://daphne-eu.github.io/daphne/ GettingStarted/#quickstart-for-developers

SystemDS Install from source
This pack heps in the restal and seture of type=01 time source code
+ 1000
- Without
+ 2. Table the project
 3. Not A Company Tool
Once the mitinitial elements is set an align to the element part of building the spelane.
Install
Windows
First samp jakes and masker to complet that spittery note that jake version in 11, we suggest using Jakes OpenJOR 11.
+ Management and
 Maximum spacing diversal opti-
Solid your anisoteneous examination with URLN (FXME and MWEN) (FXME) shares the solution with the URLN (FXME they and MARENCY/CODECare in the path investment installation. An assergie of analysis for the part and the family even in transformation of the path (FXME and CodeCare).
To non-the spatient we also takes to estable across Hadrog and spaceful Statistics. These can be hard in the Space/CSI requirities if a salt the samp size can be also and the scale context work provide samp. So, unleader for its PTH is and their and its PHH is an intervention of the scale context and across which is the scale and the Scale(Scale) information.



Getting Started: Preparing Your First Contribution



Goals

- Ability to modify the source code and run/test it
- Initial overview of relevant part of the code base
- Set Up Your Editor/IDE etc.

Issue Tracking

sphne-eu/daphne has	Q hotocare V for at 2 Ex (1)	SystemDS	
ode 🔿 Issues (204) 👔 Politequests (33) 🕤 Atlines 📳 Projetts 🔘 becarity	i ≥ malghtas	Filter Acades Santando Epice 21 - Inva Statulates Santando Ensas (Saturd	Z = International States (Second States (Printy) Z = 1
a Materia an	Q. D Labels @ Milestores New Issue	Not Existence Education Development	Plady Cost Prostage
		STUTZENDO FITO SHID-modal algoment. BURKI 600	- 775 Page 19 - 775
284 Cesed #21 Author * Labels * Projects * M	Allestones * Assignees * Types * 74 Newest *	SYSTEMOS 3624 Heliatic Refundency Explanation Meterometer	TR 0704 1
le Data transfer between Daphne and the Pandas Library using Daphnelib in Pyth	hon isn't working 😓 (metrosta)	5Y57EMD51572 Cumpressed Morphing Total 721	1101 101
ISO lithus opened on litter #	61	SYSTEM25-3554 Multi-modal feature transformation	
oin Ordering using UES	D) (9)	SYSTEM253469 Buildin Survey Control of Restrict Oract Systems	Z - Anna Stadoffer Spinishili Annas Bank State
009 - Dertichmidt opened an Jan 9		5Y57EMD5 5326 Unified Memory Wanager	D Inne Type Court Percentage
ow can daphne running as library and execute the DSL as library	p)	Lineage Tracing and Heute of 40 0000000	20 DM 37 - 3%
08 huttingunal spenet on Jan 5		Memedates >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	Ebspendeng 2 Ph
OT or jit run the daphne DSL in c++ server as function library	Di Di	SISTEMOS 2/15 Percentance searches	@bounestation 1 Ph
		1-10-052 1 2 3 4 5 6 2 Main 100	ter 2005
Iementwise binary ops on two frames 202 adamte spend on Sec 5, 2004	D I		Disprovement 50 - 14%
and and field on fee and from months in states (and field on		Pilar Assilts multiple Statest Avjets 21	Dien Fedare 30 • 105
201 planma special on Dec 5, 2024	D1	Kup Summary Status 16 Jan 3025	of Apr 201
ionuniform handling of broadcasting singletons		SYSTEM26-3760 Compression Found Immediate Result in the last 94 days (prosped work	b) View in
927 - MeetTer operand an Nev 29, 2024		Overstanding in the second sec	(7)w/ A
Codegen] Add lowering functions for actract / slice / insert (here)		SYSTEMDS THEE Canadative New Immed O Reselved Issues (52)	Basteda 3 Ph
225 - MeedTer operand an Nev 29, 2026		Agyrgains	Bas
	lata internet and l	https://issues.apache	orgliiralsocur
https://gitplin com/	dannne-ell/		$\log \left(\frac{1}{2} \right)$

Make Your First Modifications to the Code

- DAPHNE: "good first issues"
- SystemDS: Write initial test cases for your task

Read the Contribution Guidelines

Contributing to the DAPHNE System	Contribution Guidelines and Standards for SystemDS	
Those you for your interest in contribution to the DAPHNE system. Our could is to build an onen and	Thrank one for contribution in Testaw 77	
inclusive community of developers around the system. Thus, contributions are highly welcome.	indiak you not commonting to systemics.	
both from within the DAPHNE project consortium and from external researchers/developers.	The following are mostly guidelines, not rules. Use your best judgement, and feel free to propose the proposed of the second se	
In the following, you find some rough guidelines on contributing, which will most likely be extended	Changes or this out.	
and further clarified in the future.	Before contributing a pull request for review, let's make sure the changes are consistent with the	
Ways of Contributing	guidentes and cound some.	
ways or contributing	General Guidelines and Philosophy for contribution	
There are various ways of contributing including (but not limited to):	and the second second second second	
actual implementation	 Inclusion or unit tests when contributing new reacures, will neep i, prove that the code works correctly, and 	
writing test cases	ii. guard against future breaking changes.	
writing documentation	 Formatting changes can be handled in a separate PR. Example an analysis 	
 reporting bugs or any other kind of issue 	 New features (e.g., a new cutting edge machine learning algorithm) typically will live in wright 	
contributing to discussions	staging or its equivalent folder for specific feature to get some airtime and sufficient testing	
We encourage open communication about the system through comments on issues and pull requests	 When a new contribution is made to SustemDS, the maintenance hunders is the default) 	
directly on GitHub. That way, discussions are made accessible and transparent to everyone interested.	transferred to the SystemDS team. The benefit of the contribution is to be compared ap	
directly on GitHub. That way, discussions are made accessible and transparent to everyone interested. This is important to involve people and to avoid repetition in case multiple people have the same	 when a new community is made to system 35, the maintenance burden is (by default) transferred to the System 35 team. The benefit of the contribution is to be compared agains 	

blob/main/CONTRIBUTING.md

ttps://github.com/apache/systemds/ blob/main/CONTRIBUTING.md



Initial Prototype: Design/Implementation/Tests (~2/3 of the semester)



Goals

- 80% functionally complete prototype including good set of test cases
- Basis for further improvements driven by experiments and feedback

Mindset

Understand the topic

(task description, mentor, additional material)

Understand the code base

(overview and relevant parts)

- Understand the employed libraries/frameworks
- Design and implement step-by-step
- Not always "the" right solution: explore alternatives

Typical Pitfalls

- Start as the deadline approaches
 - -> don't underestimate the effort, start immediately
- Don't talk to project mentor
 - -> your mentor can give you valuable guidance



Final Prototype: Creating a Convincing Contribution (≈1/3 of the semester)



Goals

 High-quality code contribution whose value can easily be appreciated and understood

Typical Pitfalls

- Untidy, hardly documented, hardly tested code
- View experiments as nice-to-have addendum, start them as the deadline approaches
 - -> experiments show the value of your contribution
 - -> you need time to incorporate your insights
- Don't talk to project mentor
 - -> your mentor can give you valuable guidance

Countermeasures

• Focus primarily on investigating and improving your initial prototype after the intermediate presentation



Final Prototype: Code Quality



Goals

- Make your code easy to read/understand
- Others will have to maintain it after your contribution is merged

General Guidelines

- Clearly structure your code into meaningful units (classes, functions, etc.)
- Use clear yet concise identifiers (variable/function/class names)
- Stay consistent with the existing code base (e.g., use the same patterns) OR refactor if necessary
- Adhere to the code base's coding style/formatting

Keep Your Pull Request Tidy

- Stay focused: Avoid changes unrelated to your task (can be contributed as individual small pull requests)
- Don't submit anything that's useless for others (e.g., build artifacts, generated files (e.g., logs), IDE projects)
- Exclude specifics of your local setup: Avoid local paths, usernames, passwords, IP addresses etc.



Final Prototype: Tests



- Goals
 - Show functionally correct behavior
 - Experiments don't make sense if prototype doesn't do what it should

General Hints

- Unit tests and script-level tests
- Test cases that should work
- Test cases that should not work
 (e.g., invalid DSL scripts, invalid input data, ...)
- Construct simple and complex scenarios
- Think of corner cases
- Small input data is often fine (but some bugs only triggered by large inputs)

- Integrate with the Existing Test Suite
 - DAPHNE
 - -> directory: test/
 - -> see the documentation on writing test cases



- development/Testing/
- SystemDS
 - -> directory: src/test/



Final Prototype: Documentation



Goals

Make your contribution understandable for users and developers

User Documentation

- What's not documented doesn't exist
- Add high-level explanation of features and concepts behind them
- Update documentation of language abstractions (e.g., new DSL built-in functions, new types, ...)
- Update documentation of user APIs (e.g., new command-line arguments)

Developer Documentation

- Negative example: "The code is the documentation"
- High-level explanation of your contribution; justify design decisions
- API documentation of classes, functions, members, etc.

(integrate with the existing source code documentation of the system, e.g., doxygen or javadoc style)

Comments within function bodies (e.g., high-level steps of an algorithm)



Final Prototype: Experiments



Goals

- Understand your prototype to discover potential for improvement (e.g., performance bottlenecks)
- Demonstrate functional improvements (new features)
- Showcase non-functional improvements (performance compared to status quo and state-of-the-art baselines)

Design Experiments

- Don't just conduct any random experiments
- Think about which questions you want/need to answer, design experiments accordingly
- Types of experiments: Exploratory, micro benchmarks, benchmarks, end-to-end applications
- Aspects of an experiment: Data, workload, baselines, hardware & software stack, metrics

Conduct Experiments

- Automate as much as possible for repeatability (shell scripts etc.)
- Visualize and Interpret the Results
 - Automate the visualization based on raw experimental data
 - Draw conclusions and react

See 3rd seminar intro lecture on "Experiments, Reproducibility"





List of Project Topics (Proposals)

See list at https://pdamme.github.io/teaching/2025_summer/lde/ProjectTopics.pdf



Summary and Q&A

berlin

- Course Organization, Outline, and Deliverables
- Projects in DAPHNE and Apache SystemDS
- How to Approach the Project
- List of Project Topics (Proposals)
- Remaining Questions?
- Reminder: Seminar Introductory Lecture Recommended for the Project
 - 03 Experiments, Reproducibility, and Giving Presentations [May 05, 14:00]
- See you during the consultation hours and intermediate presentations ③

