Integrated **D**ata **A**nalysis **P**ipelines for Large-scale Data Management, **H**PC, and Machi**ne** Learning; DAPHNE daughter of river god Peneus (fountains, streams), chased by Apollo

DAPHNE

An Open and Extensible System Infrastructure For Integrated Data Analysis Pipelines

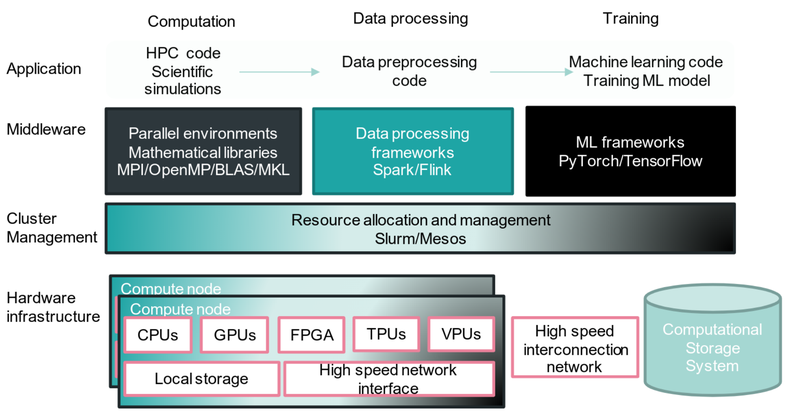
The DAPHNE project is funded by the European Union's Horizon 2020 research and innovation program under grant agreement number 957407 for the time period from Dec/2020 through Nov/2024.

https://daphne-eu.eu/

Contact Us: Daphne_all [at] know-center [dot] at

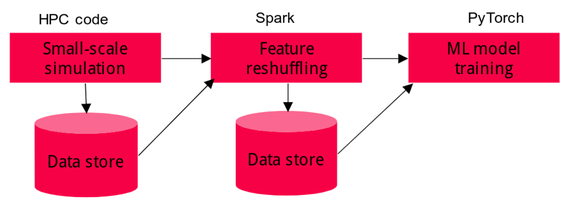
1. Integrated Data Analysis Pipelines

- Integrated Data Analysis (IDA) pipelines are increasingly common in practice, sharing compilation and runtime techniques, and cluster hardware.



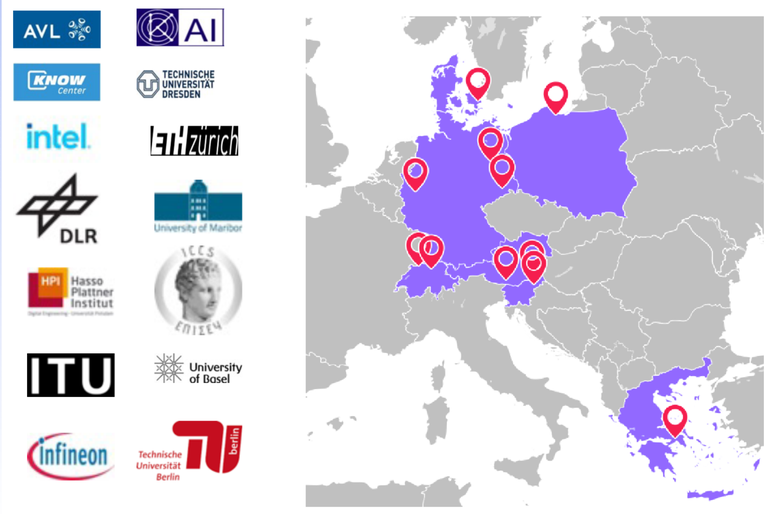
- Different system libraries, programming models

- Data exchange between IDA components



Ecosystem for an integrated data analysis pipeline, from [1].

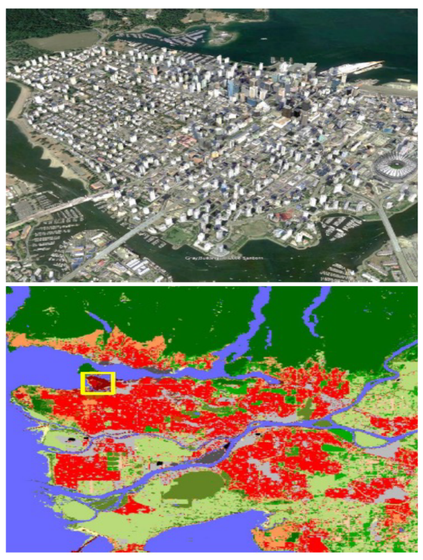
2. DAPHNE Consortium



3. Motivating Use Cases

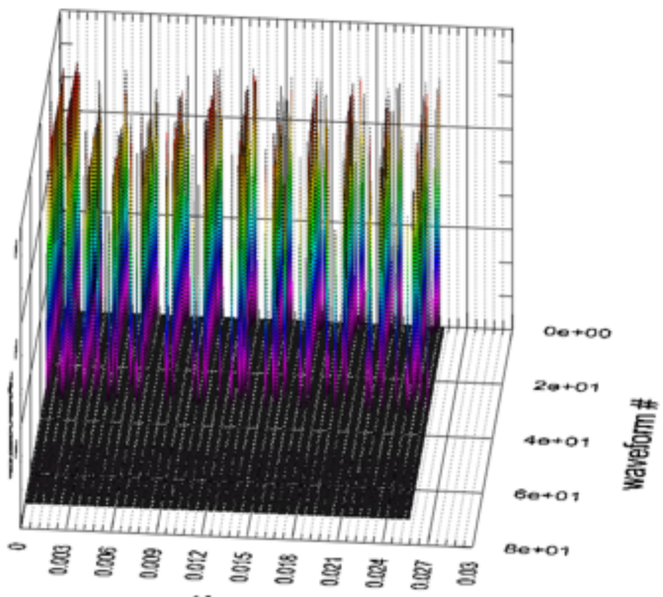
EARTH OBSERVATION

This use case aims at developing a deep learning pipeline for local climate zone classification, based on 4 PB of satellite images.



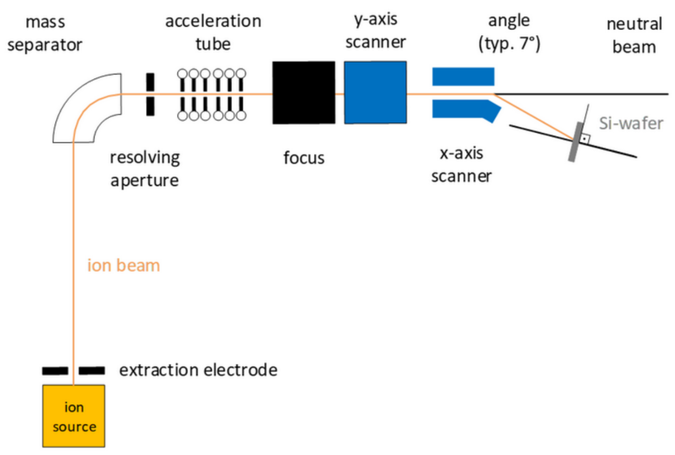
MATERIAL DEGRADATION

This use case focuses on understanding and modeling material degradation during the operation of semiconductor devices.



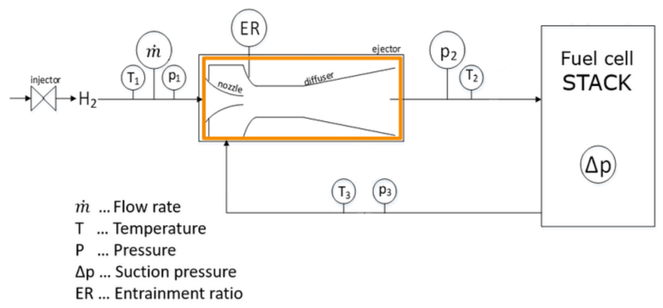
SEMICONDUCTOR MANUFACTURING

This use case aims to optimize implantation equipment's stability and utilization. Ion implanters generate many sensor readings, making them perfect for ML algorithms to learn from.

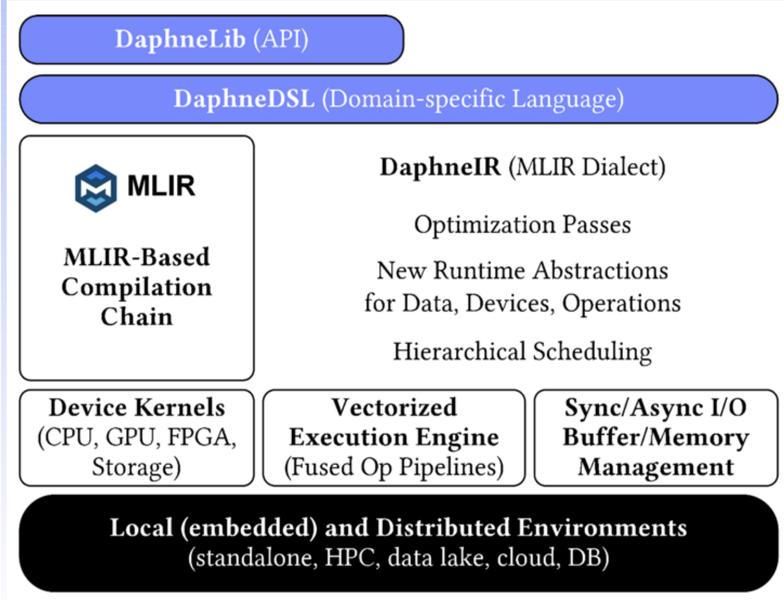


AUTOMOTIVE VEHICLE DEVELOPMENT

This use case focuses on developing a closed-loop high-dimensional optimization problem supported by physics-based simulations and behavioral modeling.



4. DAPHNE System



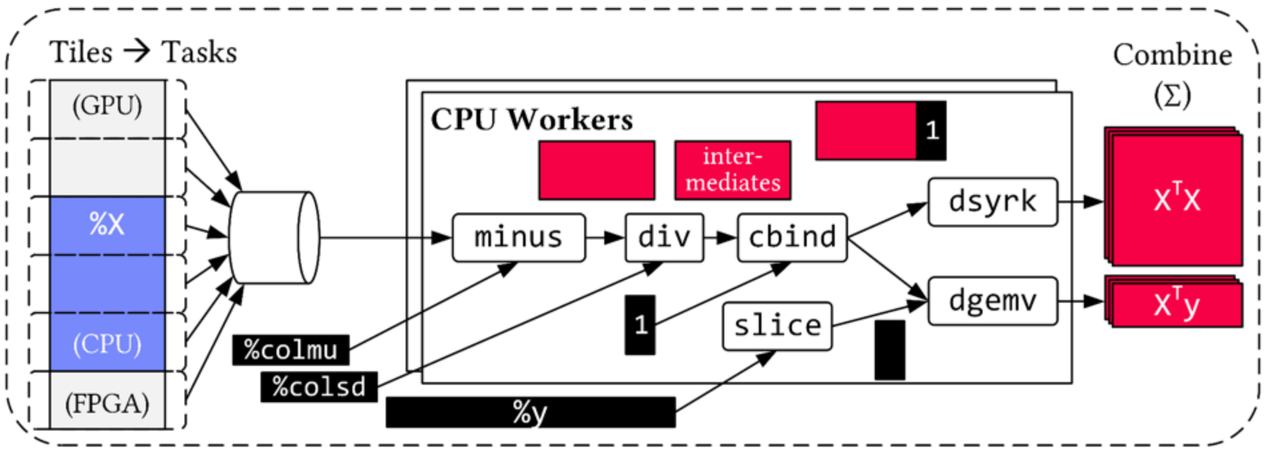
5. DAPHNE DSL Code Example

Connected Components Algorithm [3]

```
maxi = 0;
verbose = true;
n = as.f64($n);
e = as.f64($e);
UT = upperTri(rand(n, n, 1.0, 1.0, 2.0*e/n^2.0, -1), false, false);
G = UT + t(UT);
if( sum(sum(G,0) != t(sum(G,1))) > 0.0 ) {
    print("Connected Components: input graph needs to be "
        + "symmetric but rowSums and colSums don't match up.");
}
c = seq(1.0, as.f64(nrow(G)), 1.0);
diff = as.f64(nrow(G));
iter = 1;
while( as.si64(diff > 0.0) && (maxi==0
    || iter<=maxi) ) {
    u = max(aggMax(G * t(c), 0), c);
    diff = sum(u != c);
    c = u; # update assignment
    if( verbose ) {
        print("Connected components: iter = ",0,0);
        print(iter+", #diff = "+diff);
    }
    iter = iter + 1;
}
writeMatrix(c, $C);
```

6. Vectorized Execution Engine

```
(%9, %10) = fusedPipeline1(%X, %y, %colmu, %colsd) {
```



Locality-aware, Multi-device Scheduling

Default Parallelization Frame & Matrix Ops

Fused Operator Pipelines on Tiles/Scalars + Codegen

7. Project Timeline

- December 2020 — Project Start
- December 2021 — 1^{st} Prototype
- July 2022 — 1^{st} Review meeting
- October 2022 — Release 0.1
- $1^{st}Q$ 2023 — Release 1.0
- January 2024 — 2^{nd} Review meeting
- $1^{st}Q$ 2024 — Release 2.0
- December 2024 — Project End

References

[1] N. Ihde, P. Marten, A. Eleliemy, G. Poerwawinata, P. Silva, I. Tolovski, F. M. Ciorba, and T. Tilmann "*A Survey of Big Data, High Performance Computing, and Machine Learning Benchmarks*", In Proceedings of the Technology Conference on Performance Evaluation and Benchmarking, 2021.

[2] P. Damme, M. Birkenbach, C. Bitsakos, M. Boehm, P. Bonnet, F. M. Ciorba, M. Dokter, P. Dowgiallo, A. Eleliemy, C. Faerber, G. Goumas, D. Habich, N. Hedam, M. Hofer, W. Huang, K. Innerebner, V. Karakostas, R. Kern, T. Kosar, D. Krems, A. Laber, W. Lehner, E. Mier, M. Paradies, B. Peischl, G. Poerwawinata, S. Psomadakis, T. Rabl, P. Ratuszniak, A. Starzacher, P. Silva, N. Skuppin, B. Steinwender, I. Tolovski, P. Tözün, W. Ulatowski, Y. Wang, I. Wrosz, A. Zamuda, C. Zhang, and X. Zhu. "*DAPHNE: An Open and Extensible System Infrastructure for Integrated Data Analysis Pipelines*". In Proceedings of the 12th Annual Conference on Innovative Data Systems Research (CIDR '22), Chaminade, USA, January

[3] https://github.com/daphne-eu/daphne/blob/main/scripts/algorithms/components.daph