Adaptive Resource Allocation for (Bioinformatics-)Workflows

Dr. Lauritz Thamsen

Lecturer in Computer Systems

GLASS – SoCS – UofG

https://lauritzthamsen.org

Data-Intensive Applications

Search

Business Intelligence

Telemedicine

Mobility

Recommendations

Industry 4.0

Scientific Data Analysis

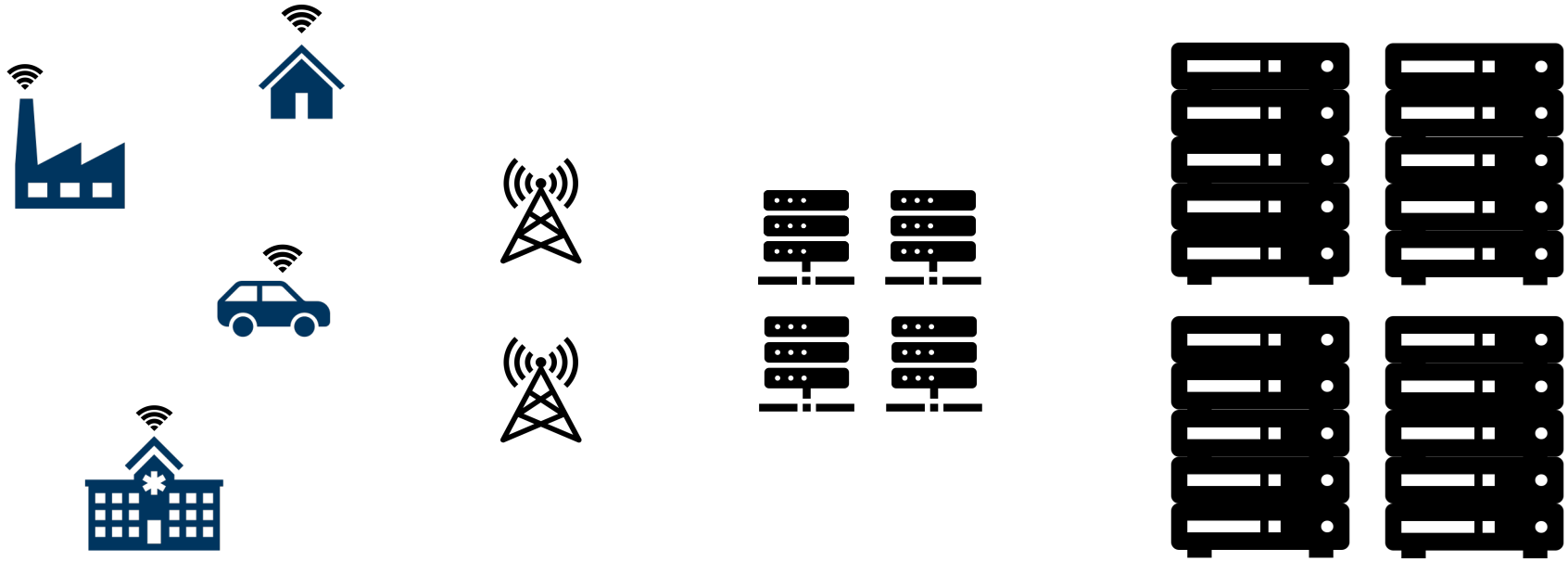
Log Analysis

Smart Homes

Smart Water Networks

Diverse Computing Infrastructures

Heterogeneous and dynamic distributed computing environments from devices to data centers

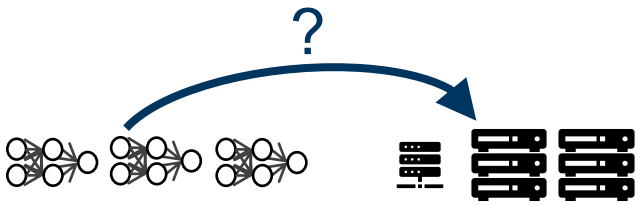


Research Questions

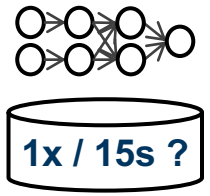
Given a job and objectives/constraints for its execution:

1. What resources to use for a job?

2. When and where to run the jobs?



3. How to set system configurations?



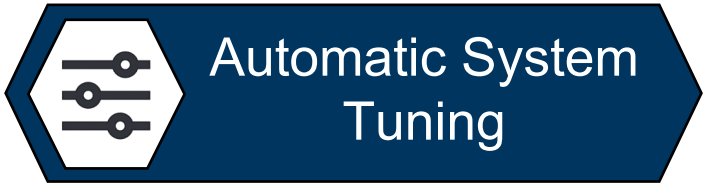
Vision: Adaptive Resource Management

Select resource allocations to meet specific performance objectives and constraints

CloudCom'18, EDGE'19, BigData'20, IPCCC'20, ICFEC'21, EdgeSys'21, IC2E'21, CLUSTER'21, SAC'22, Euro-Par'22, SSDBM'22, IC2E'22

Adjust resource configurations at runtime as workloads change or components fail

CloudCom'17, BigData'18, BigDataCongress'18, CCPE journal'20, IC2E'21, IPCCC'21, ACSOS'21, BigData'21, SPE'21, Middleware'21, ISORC'22



Tune system configurations using monitoring data, profiling, and performance models

BigData'19, BigData'20, IC2E'22, FedCSIS'22, ICWS'22

Resource Allocation for Workflows

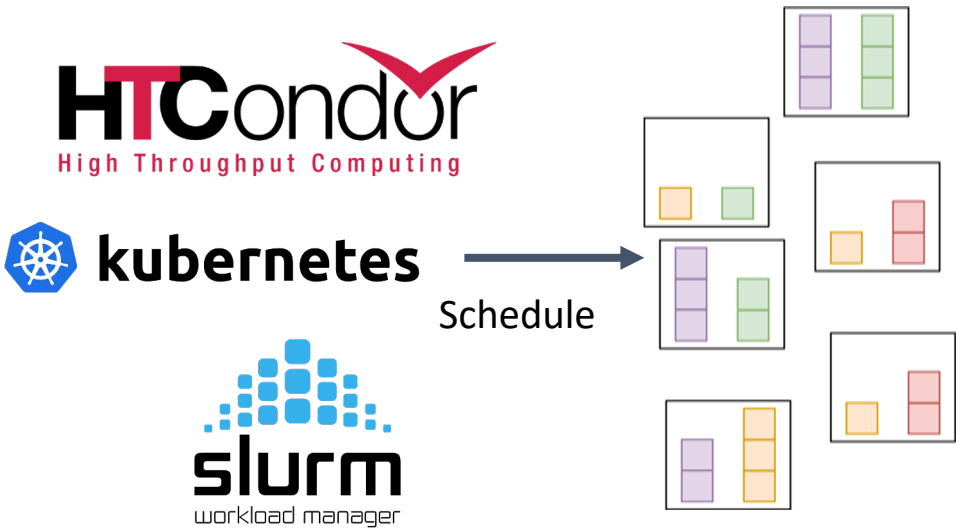
Tarema: Adaptive Resource Allocation for Scalable Scientific Workflows in Heterogeneous Clusters. Bader, Thamsen, Kulagina, Will, Meyerhenke, Kao. Big Data 2021.

Lotaru: Locally Estimating Runtimes of Scientific Workflow Tasks in Heterogeneous Clusters. Bader, Lehmann, Thamsen, Will, Leser, Kao. SSDBM 2022.

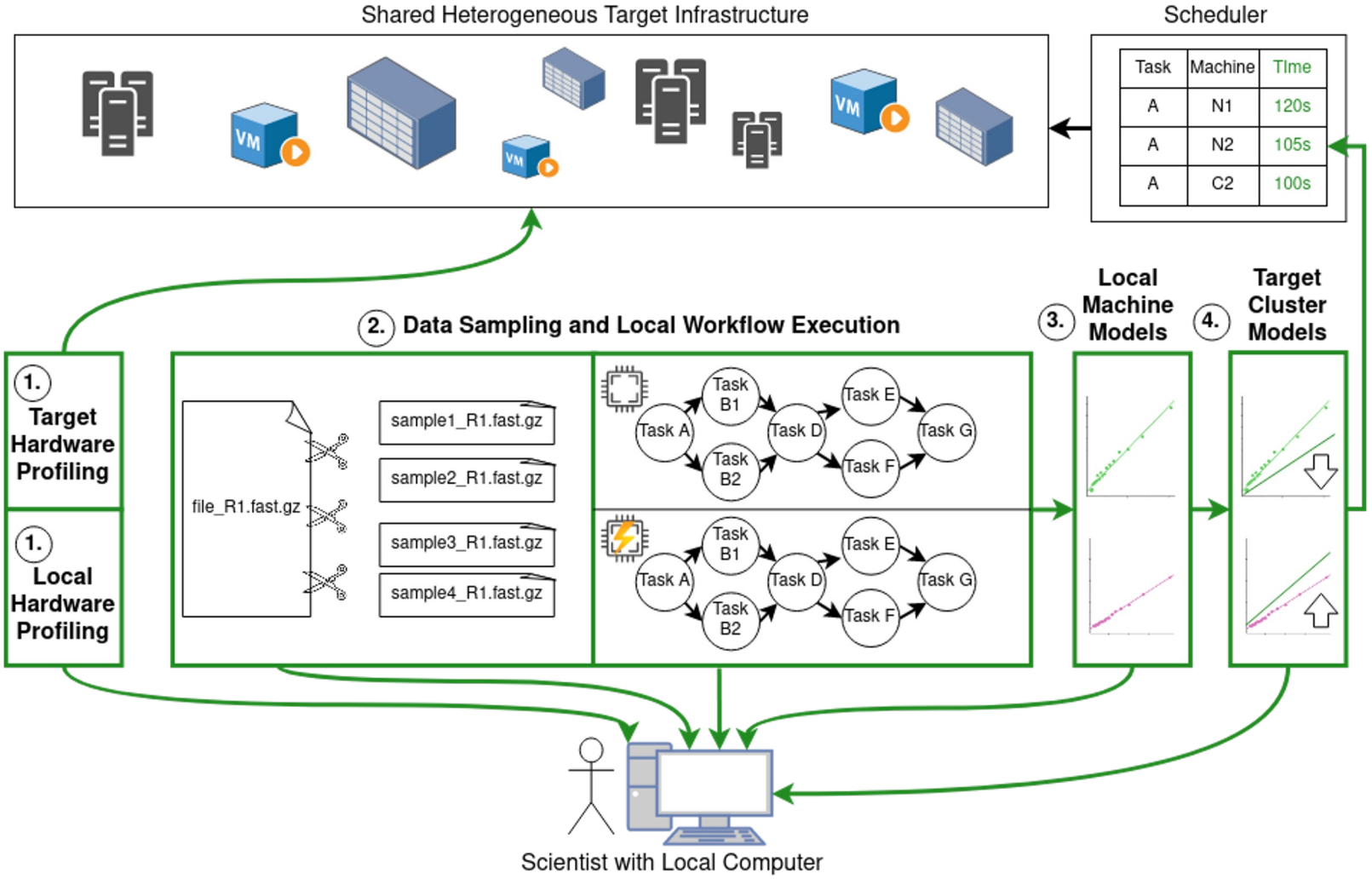
Reshi: Recommending Resources For Scientific Workflow Tasks on Heterogeneous Infrastructures. Bader, Lehmann, Groth, Thamsen, Scheinert, Will, Leser, Kao. Under review. 2022.

Starting Point

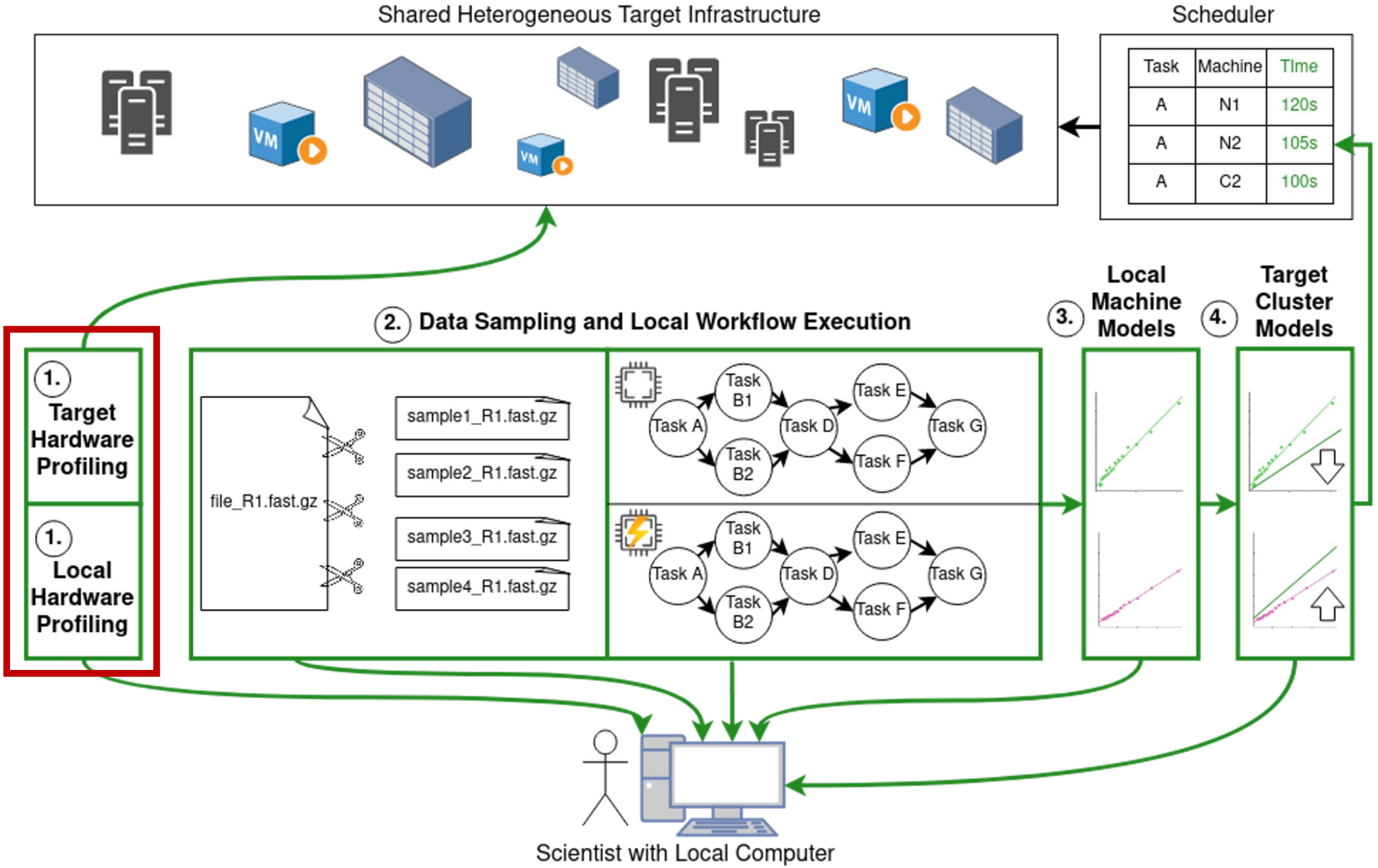
- Clusters are commonly heterogeneous
- Scheduling research, has put forward many interesting methods
- However, widely used cluster resource managers apply simple methods
- So, why is that?
  - Lack of historical executions
  - Knowledge about task runtimes on each machine is required



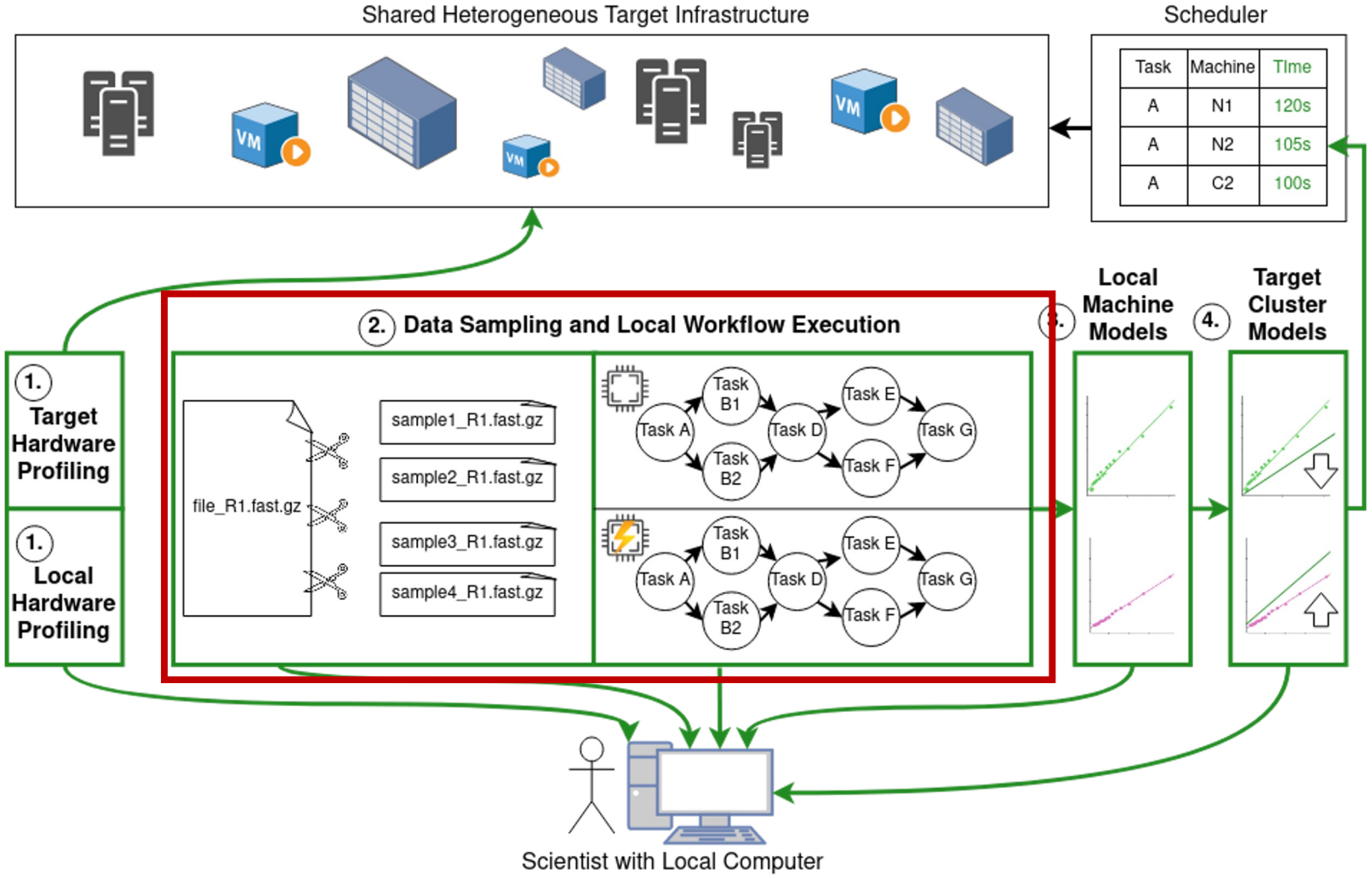
Our Approach “Lotaru”



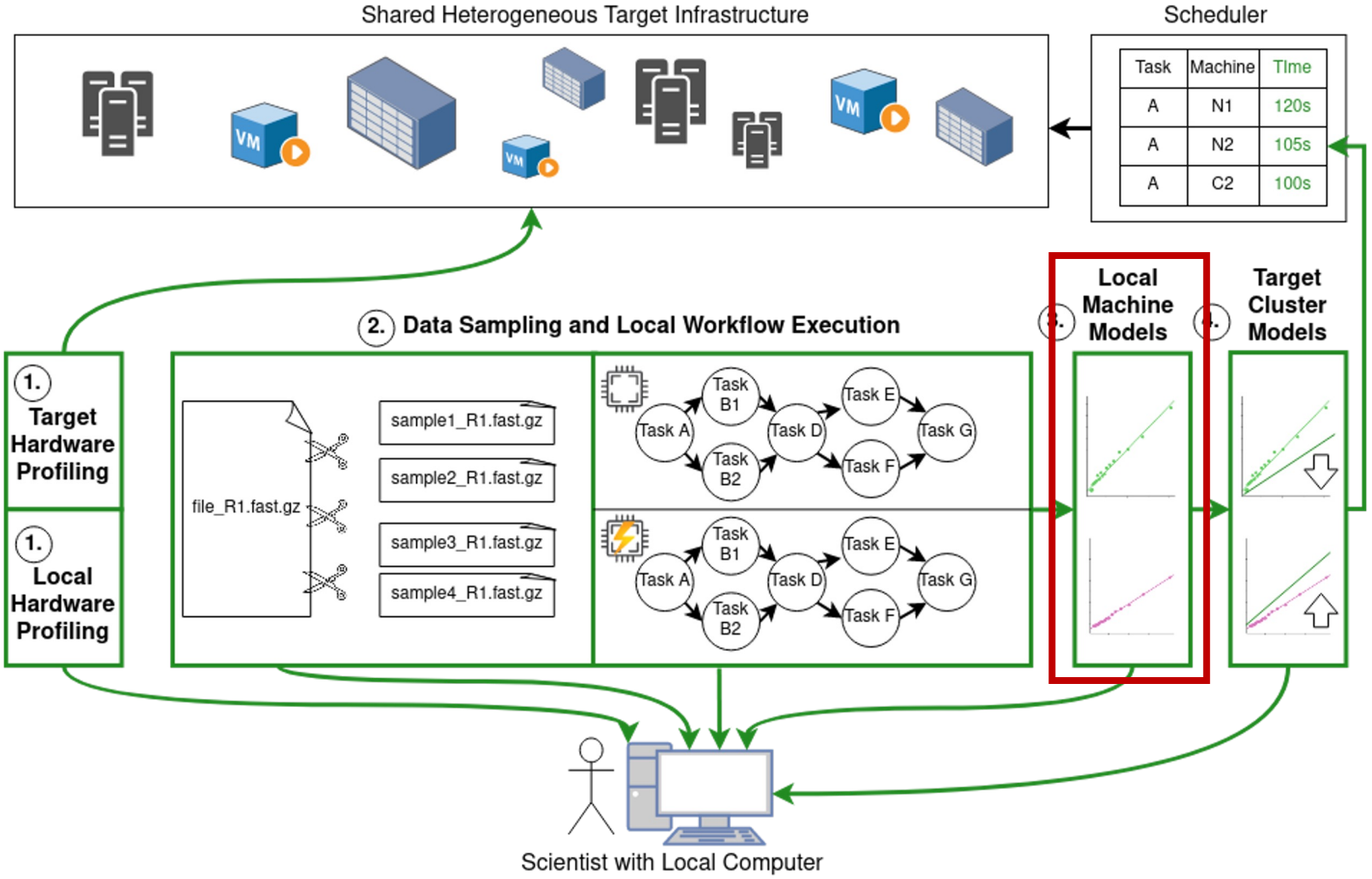
Our Approach “Lotaru”



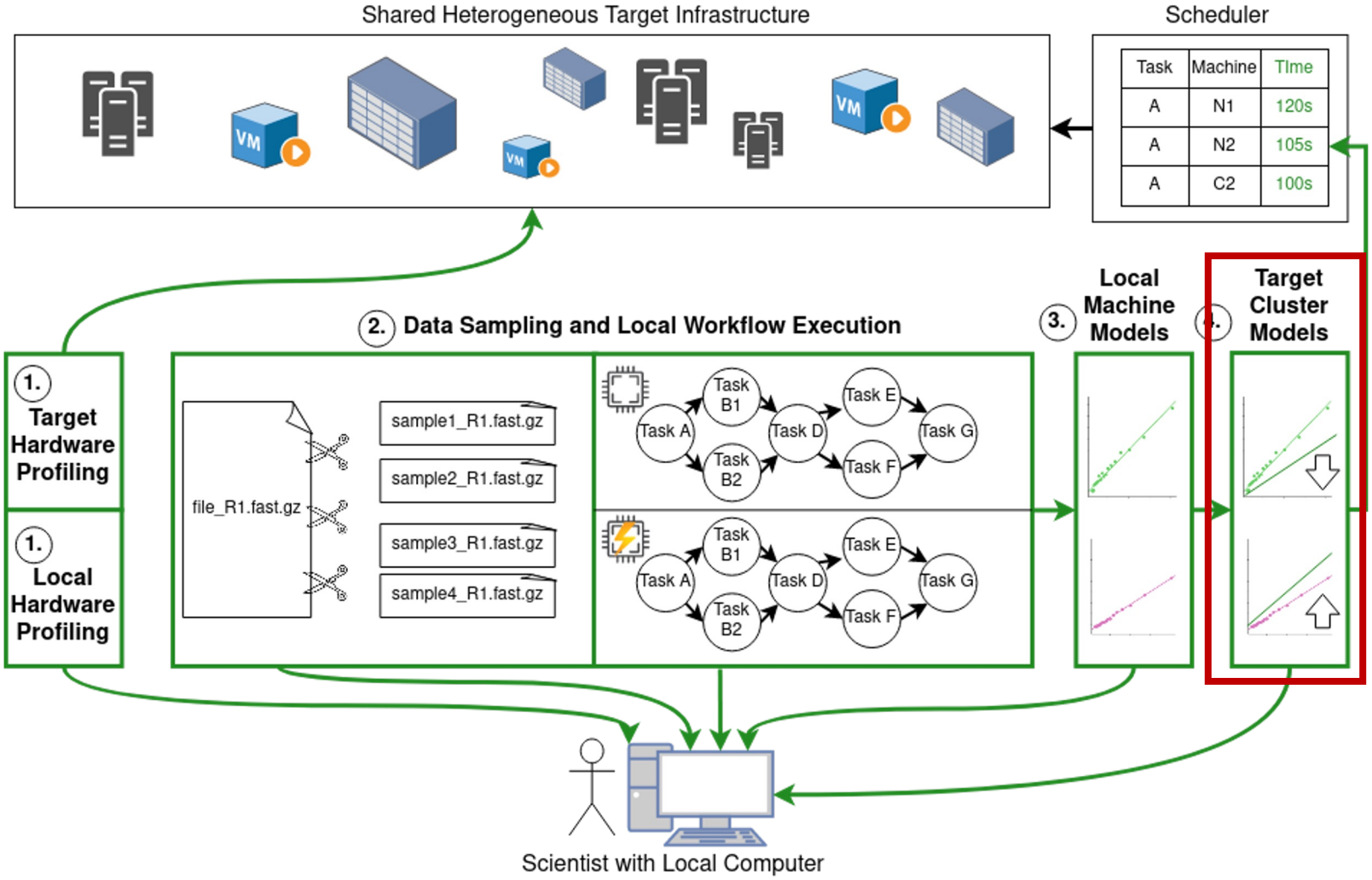
Our Approach “Lotaru”



Our Approach “Lotaru”



Our Approach “Lotaru”



1. Infrastructure Profiling 1/2

- Resources not only differ in capacities (# CPUs / cores or amount of memory), but have different performances
- Idea: Use microbenchmarks (< 1min) to gather node performance
  - CPU
  - Memory
  - read/write I/O
- Rerun profiling when changes are detected

1. Infrastructure Profiling 2/2

- All resources (local machine and cluster nodes) profiled using sys-bench, fio, and LINPACK
- Results for a local machine and five cluster nodes:

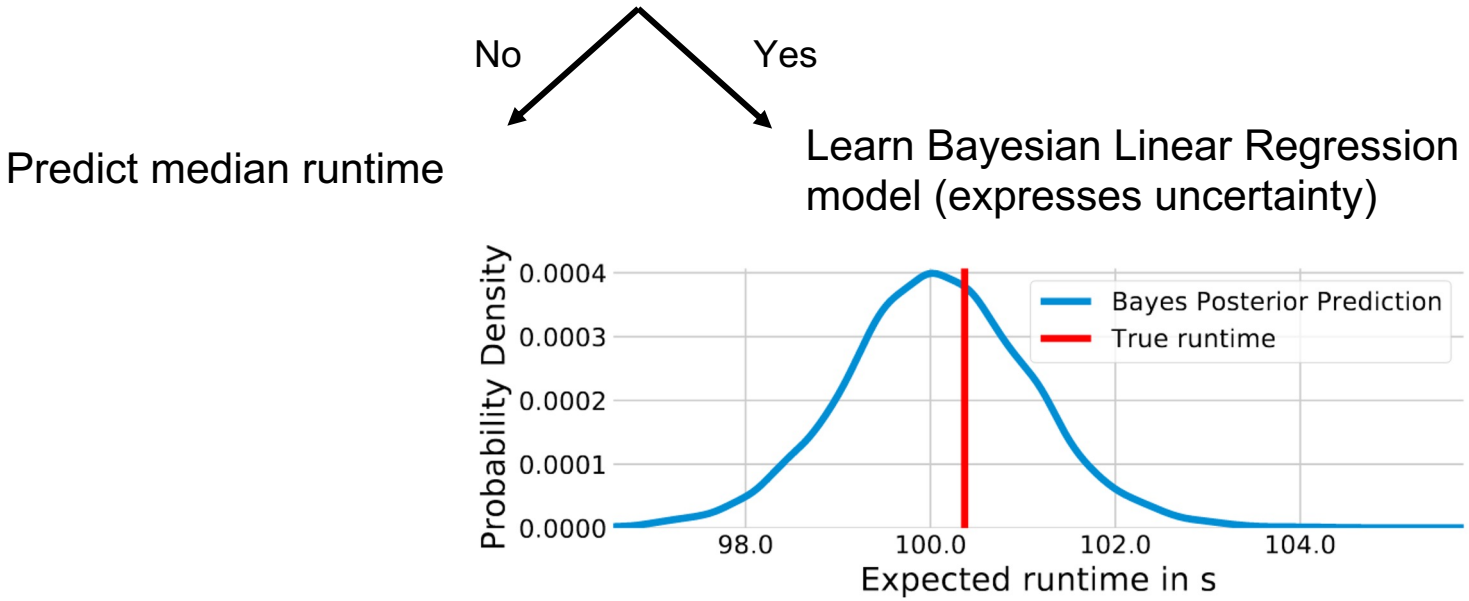
Machine	# CPUs	Memory	Storage	CPU events/s	LINPACK	RAM score	read IOPs	write IOPS
Local	8	16 GB	HDD	458	3,959,800	18,700	414	415
A1	2 x 4	32 GB	HDD	223	-	11,000	306	301
A2	2 x 4	32 GB	HDD	223	-	11,000	341	336
N1	8	16 GB	HDD	369	3,620,426	13,400	481	483
N2	8	16 GB	HDD	468	4,045,289	17,000	481	483
C2	8	32 GB	HDD	523	4,602,096	18,900	481	483

2. Sampling and Local Execution

- Possibly hundreds of input files
  - Select one and sample it down
- Create several small input files to generate training data
  - More samples can lead to better models (but longer profiling)
- Run workflow locally with samples
- Decrease the CPU frequency and run the workflow again locally

3. Local Prediction Model Training

- Estimate runtimes given a certain input data size
- Linear correlation between *uncompressed* input data size and workflow task runtime?



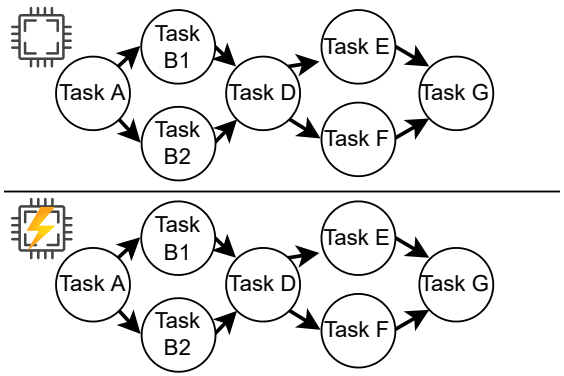
4. Adjusting to Target Node

- Local machine (e.g. a scientist's computer) is different from target cluster nodes
- However, we want estimates for all task-node combinations
  → Translate runtimes based on measured performance

- Deviation: $$dev = \frac{time_{normal} - time_{redFreq}}{time_{redFreq}}$$

- Weighting: $$w = max\left(0; min\left(1; \frac{median_{dev}}{(freq_{old}/freq_{new}) - 1}\right)\right)$$

- Factor: $$f_t = w * \frac{cpu_{local}}{cpu_{target}} + (1 - w) * \frac{io_{local}}{io_{target}}$$



Evaluation – Setup

- 6 different nodes

Machine	# CPUs	Memory	Storage	CPU events/s	LINPACK	RAM score	read IOPs	write IOPS
Local	8	16 GB	HDD	458	3,959,800	18,700	414	415
A1	2 x 4	32 GB	HDD	223	-	11,000	306	301
A2	2 x 4	32 GB	HDD	223	-	11,000	341	336
N1	8	16 GB	HDD	369	3,620,426	13,400	481	483
N2	8	16 GB	HDD	468	4,045,289	17,000	481	483
C2	8	32 GB	HDD	523	4,602,096	18,900	481	483

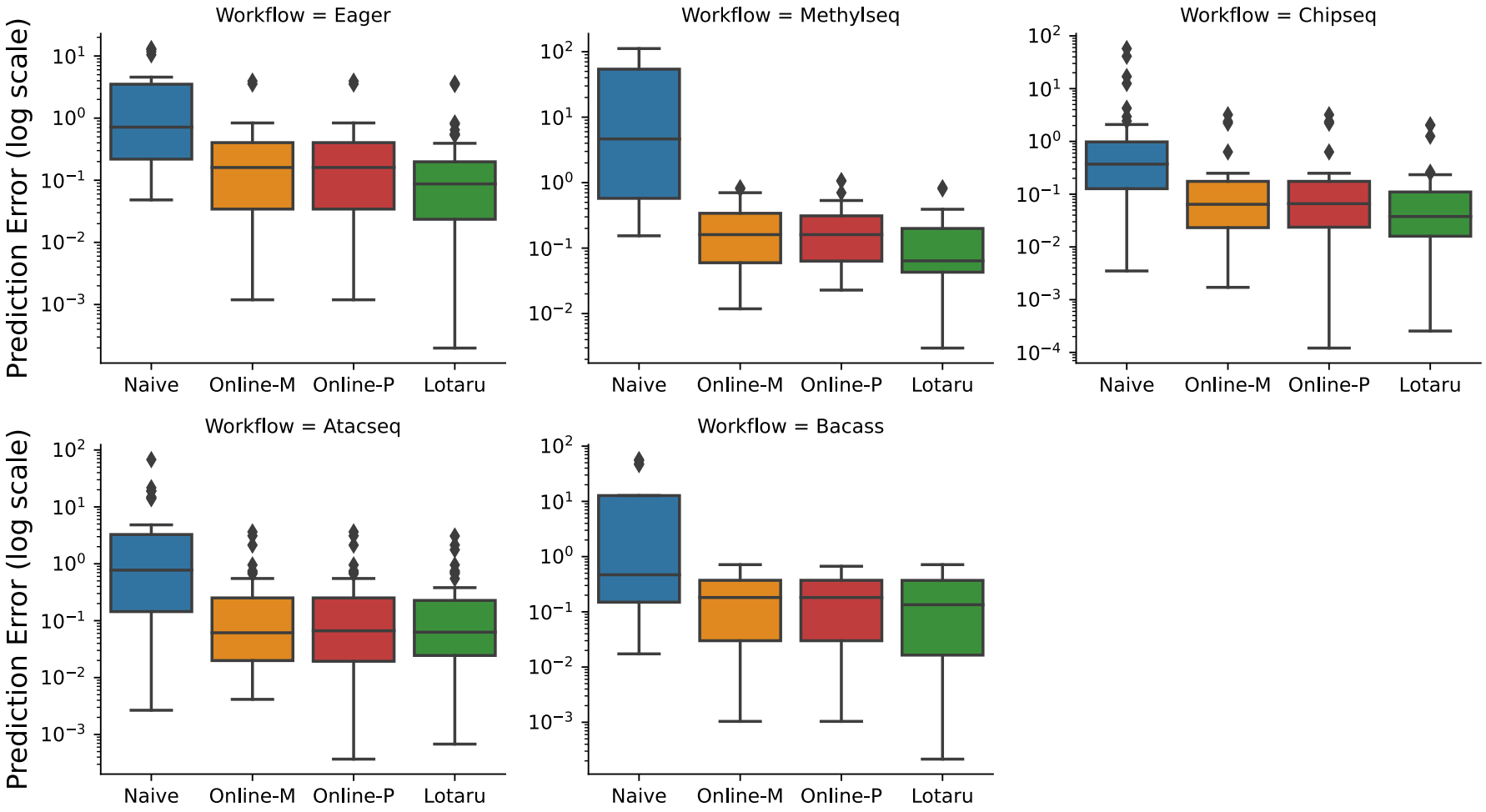
- 5 real-world workflows (nf-core repository)

Workflow	# Abstract Tasks	Sample	Size	Uncompr. Size	Runtime Per Input
Eager	13	1	1.52 GB	8.33 GB	148 min
		2	4.34 GB	25.71 GB	211 min
Methylseq	8	1	3.61 GB	17.03 GB	90 min
		2	4.75 GB	22.50 GB	117 min
Chipseq	14	1	1.33 GB	4.81 GB	140 min
		2	8.71 GB	32.98 GB	948 min
Atacseq	14	1	3.26 GB	14.09 GB	184 min
		2	2.40 GB	11.81 GB	104 min
Bacass	5	1	1.23 GB	3.64 GB	237 min
		2	1.45 GB	4.35 GB	253 min

- Baselines:
  - Naive
  - Toward fine-grained online task characteristics estimation in scientific workflows [1] - Online-M
  - Online task resource consumption prediction for scientific workflows [2] - Online-P

[1] Da Silva et al., Toward Fine-Grained Online Task Characteristics Estimation in Scientific Workflows. WORKS. 2013.
[2] Da Silva et al., Online Task Resource Consumption Prediction for Scientific Workflows. Par. Proc. Letters 25. 2015.

Evaluation on a Heterogeneous Cluster



Outcomes

- New online task runtime estimation method with 15.99% error (vs. 30.90% error of the best baseline)
- Working prototype implementation for Nextflow, https://github.com/CRC-FONDA/Lotaru
- Trace repository with more than 9,000 task executions from 5 different scientific workflows on 6 different machine types, https://github.com/CRC-FONDA/Lotaru-traces

Outlook

- Application domain-specific microbenchmarks: experimenting with common bioinformatics tasks
- How well do our runtime estimates work for SotA scheduling methods that rely on knowing task runtimes upfront (e.g. HEFT)?
- Our own scheduling that takes into account estimates, uncertainty, and resource performance profiles

Acknowledgements

Results from the DFG Collaborative Research Center *Foundations of Workflows for Large-Scale Scientific Data Analysis* (FONDA, https://fonda.hu-berlin.de)

Jonathan Bader (TUB)

Fabian Lehmann (HUB)

Svetlana Kulagina (HUB)

Henning Meyerhenke (HUB)

Ulf Leser (HUB)

Odej Kao (TUB)

References & Contact

Tarema: Adaptive Resource Allocation for Scalable Scientific Workflows in Heterogeneous Clusters. Bader, Thamsen, Kulagina, Will, Meyerhenke, Kao. IEEE Big Data 2021.

Lotaru: Locally Estimating Runtimes of Scientific Workflow Tasks in Heterogeneous Clusters. Bader, Lehmann, Thamsen, Will, Leser, Kao. ACM SSDBM 2022.

Reshi: Recommending Resources For Scientific Workflow Tasks on Heterogeneous Infrastructures. Bader, Lehmann, Groth, Thamsen, Scheinert, Will, Leser, Kao. Under review, but happy to share.

My email: lauritz.thamsen@glasgow.ac.uk

Personal website: https://lauritzthamsen.org

FONDA: https://fonda.hu-berlin.de/