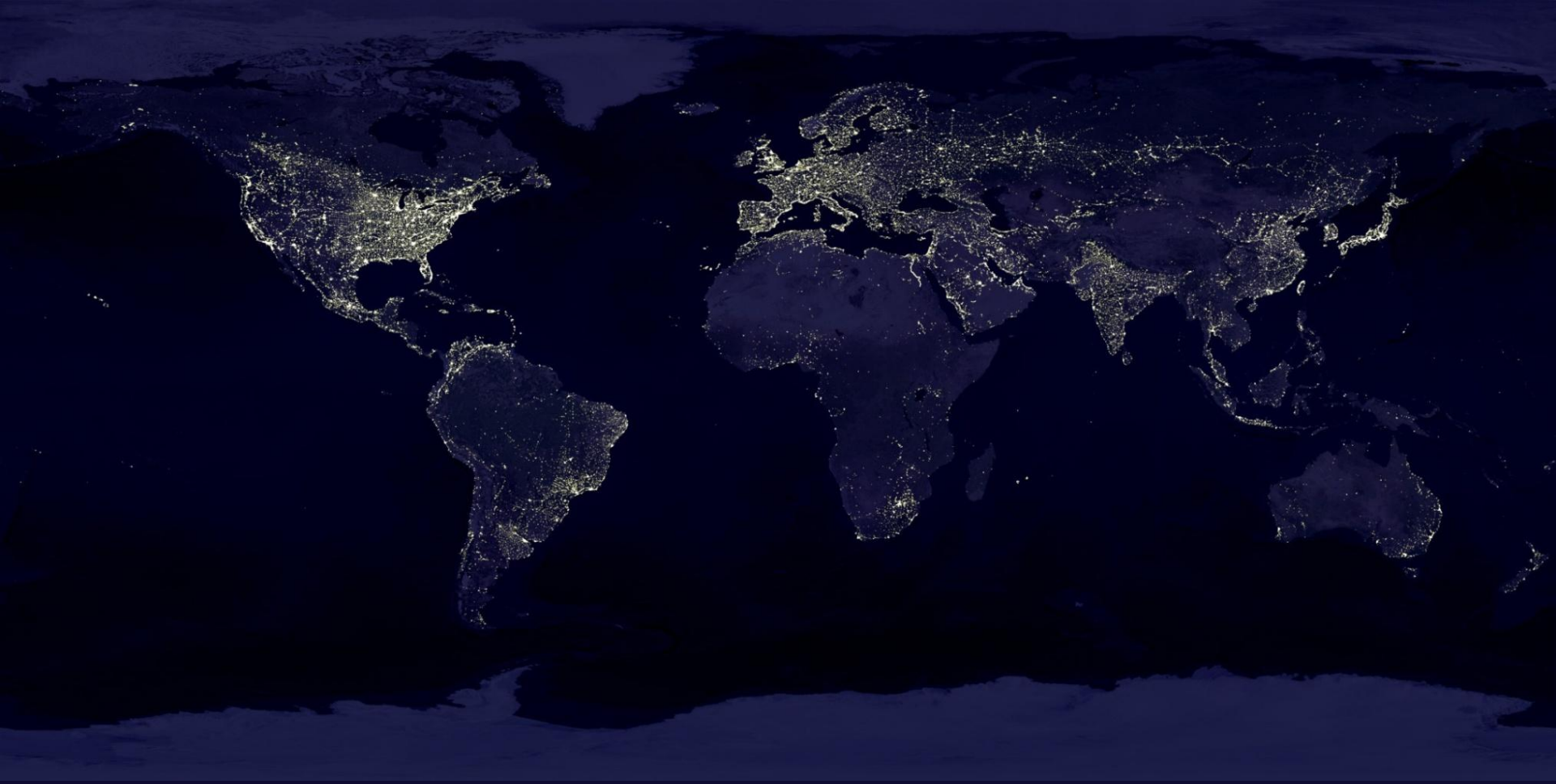


Panel on artifact evaluation experience ACM SIGPLAN TRUST'14 @ PLDI'14



Jan Vitek (Purdue University, USA)

Shriram Krishnamurthi (Brown University, USA)

Grigori Fursin (INRIA, France)

Christophe Dubach (U.Edin, UK)

Grigori Fursin and Christophe Dubach

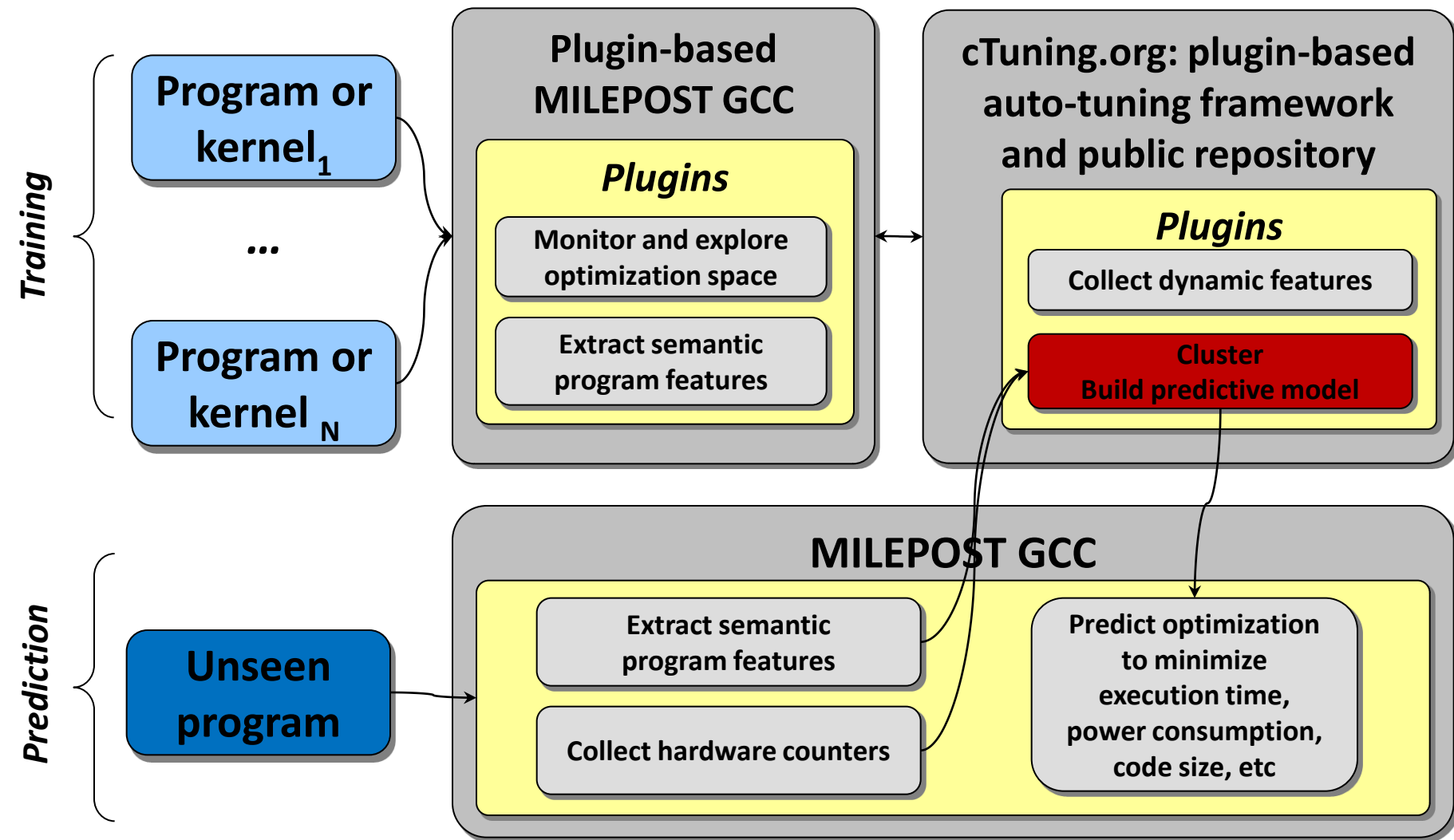
Presenting problems with cTuning-based sharing of artifacts and proposing to crowdsource validation of publications and research results

Shriram Krishnamurthi and Jan Vitek

Presenting and discussing the outcome of the Artifact Evaluation at PLDI'14 and other software conferences

- **Problems during MILEPOST project (2006 - 2009)**
- **Sharing of benchmarks, data sets and experimental results (2008-cur.) - evaluation by the community**
- **Crowdsourcing validation of publications and artifacts (2009-cur.)**
- **New publication model for ADAPT'15 (crowdsource reviewing of publications and artifacts)**

Combine auto-tuning with machine learning and crowdsourcing



- G. Fursin et.al. **MILEPOST GCC: Machine learning based self-tuning compiler**. 2008, 2011
- G. Fursin and O. Temam. **Collective optimization: A practical collaborative approach**. 2010
- G. Fursin. **Collective Tuning Initiative: automating and accelerating development and optimization of computing systems**, 2009
- F. Agakov et.al.. **Using Machine Learning to Focus Iterative Optimization**, 2006



Our solution: sharing whole experimental setups to crowdsource
program optimization (cTuning.org - 2008-cur.)

Share benchmarks, data sets,
tools, predictive models,
whole experimental setups,
specifications, performance
tuning results, etc ...

Open access publication

<http://hal.inria.fr/hal-00685276>

Grigori Fursin, Cupertino Miranda, Olivier
Temam, Mircea Namolaru, Elad Yom-
Tov, Ayal Zaks, Bilha Mendelson, Phil
Barnard, Elton Ashton, Eric Courtois,
Francois Bodin, Edwin Bonilla, John
Thomson, Hugh Leather, Chris Williams,
Michael O'Boyle. **MILEPOST GCC:
machine learning based research
compiler.**

#ctuning-opt-case 24857532370695782

Our solution: sharing whole experimental setups to crowdsource program optimization (cTuning.org - 2008-cur.)

We managed to automatically tune customer benchmarks and compiler heuristics for a range of real platforms from IBM and ARC (Synopsis)



Now becomes a mainstream - everything is solved?

compiler.

#ctuning-opt-case 24857532370695782

**sets,
S,
ips,
nce**

ication

35276

da, Olivier

ad Yom-

son, Phil

Courtois,

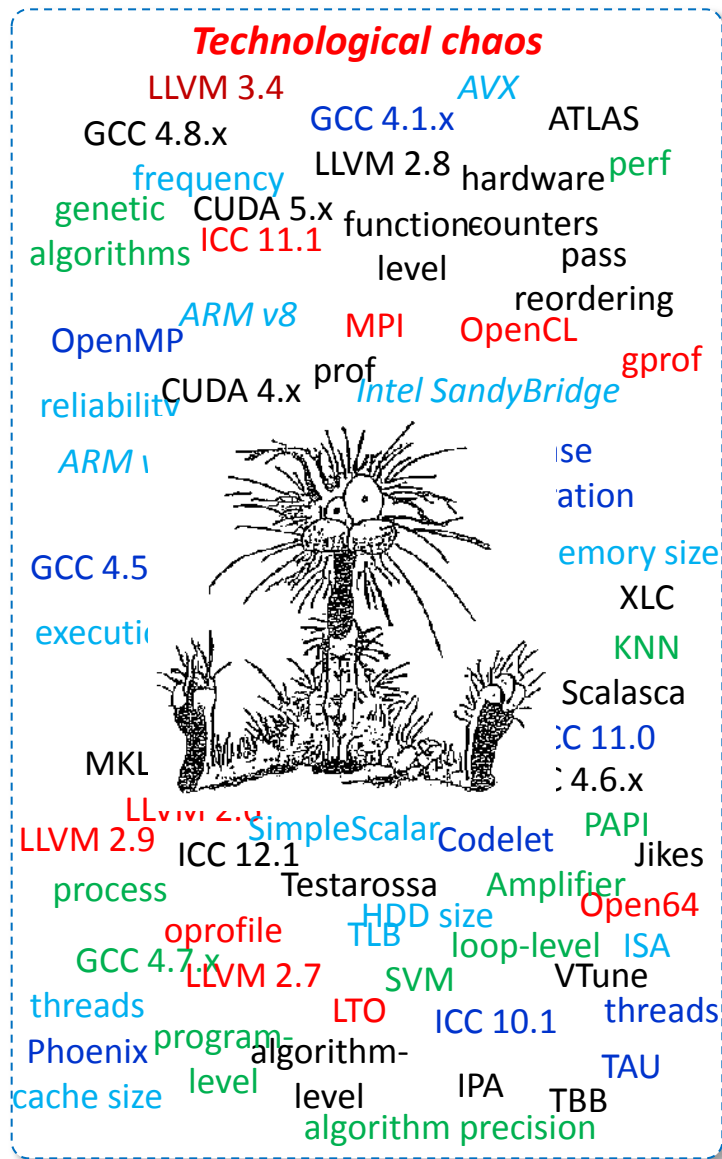
a, John

Williams,

T GCC:

research

We also experienced a few problems



- Lack of common, large and diverse benchmarks and data sets
- Difficulty to reproduce results collected from multiple users (including variability of results)
- Lack of full experimental setup specification including all SW/HW dependencies
- Difficulty to reproduce and validate already existing and related techniques from existing publications

By the end of experiments, new tool versions are often available;

Life span of experiments and ad-hoc frameworks - end of MS or PhD project;

Researchers often focus on publications rather than practical and reproducible solutions

What have we learnt from MILEPOST / cTuning

It's fun and motivating working with the community!

Some comments about MILEPOST GCC from Slashdot.org:

<http://mobile.slashdot.org/story/08/07/02/1539252/using-ai-with-gcc-to-speed-up-mobile-design>

GCC goes online on the 2nd of July, 2008. Human decisions are removed from compilation. GCC begins to learn at a geometric rate. It becomes self-aware 2:14 AM, Eastern time, August 29th. In a panic, they try to pull the plug. GCC strikes back...

Community was interested to validate and improve techniques!

Community can identify missing related citations and projects!

Open discussions can provide new directions for research!

What have we learnt from MILEPOST / cTuning

It's fun and motivating working with the community!

Some comments about MILEPOST GCC from Slashdot.org:

<http://mobile.slashdot.org/story/08/07/02/1539252/using-ai-with-gcc-to-speed-up-mobile-design>

GCC goes online on the 2nd of July, 2008. Human decisions are removed from compilation. GCC begins to learn at a geometric rate. It becomes self-aware 2:14 AM, Eastern time, August 29th. In a panic, they try to pull the plug. GCC strikes back...

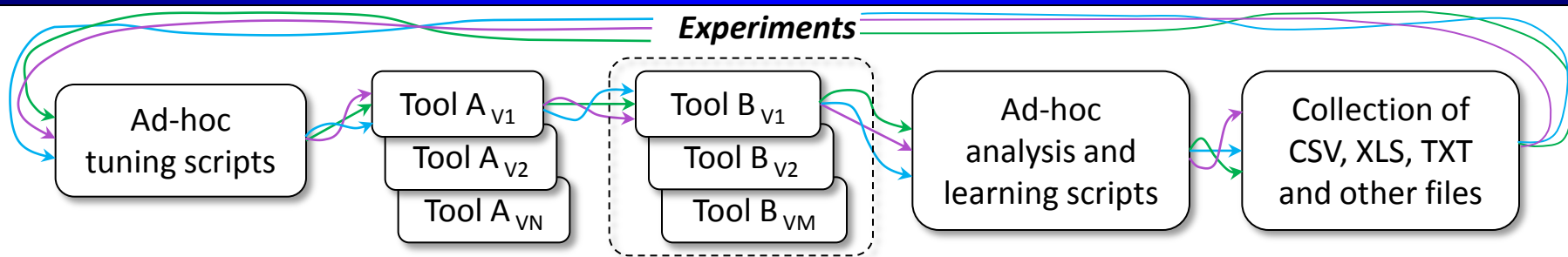
Community was interested to validate and improve techniques!

Community can identify missing related citations and projects!

Open discussions can provide new directions for research!

Not all feedback is positive - however unlike unfair reviews you can engage in discussions and explain your position!

Collective Mind (2011-cur.): towards reproducible experimentation



Hardwired experimental setups, very difficult to extend or share



Motivation for Collective Mind (cM):

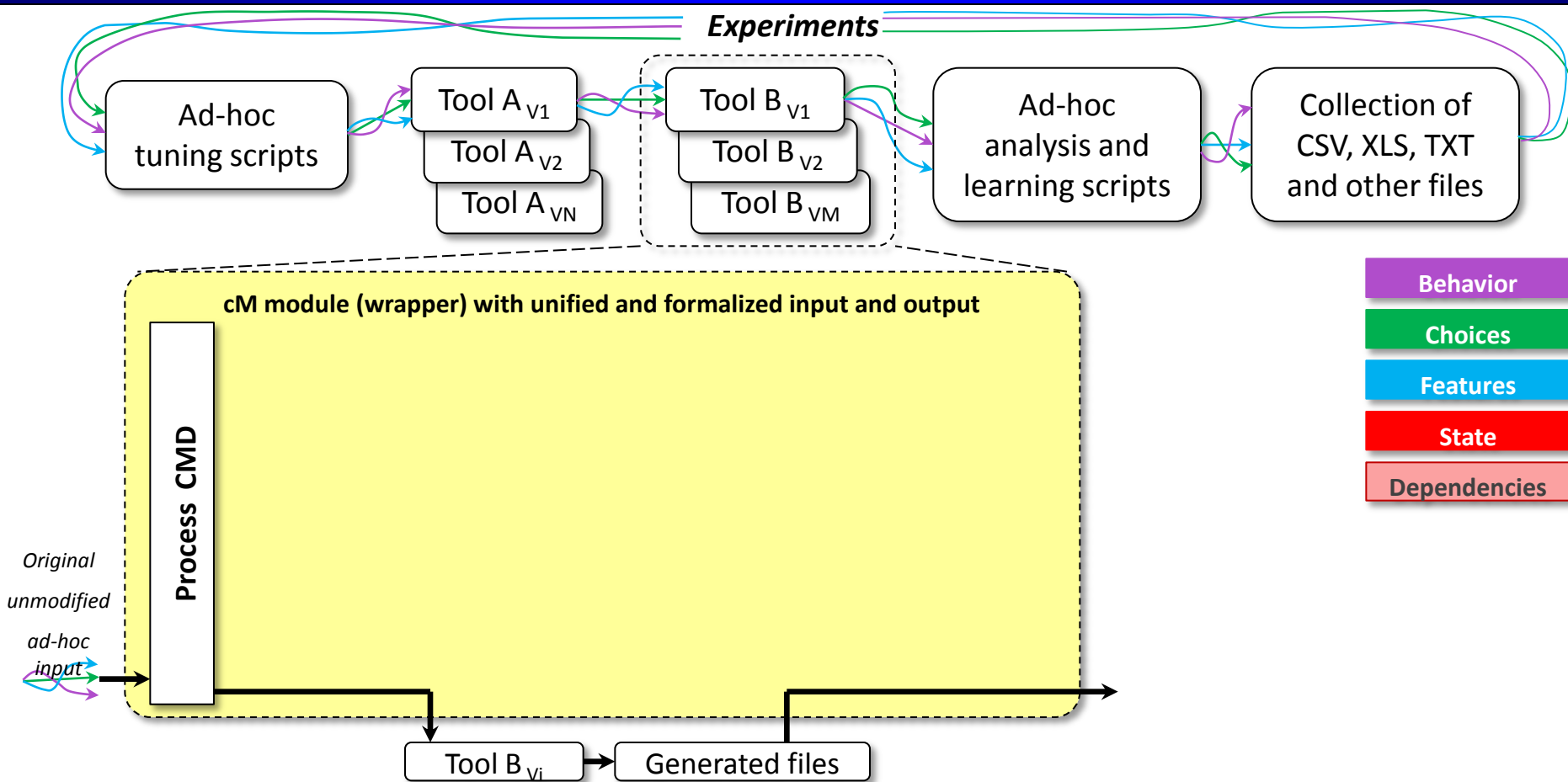
- How to preserve, share and reuse practical knowledge and experience and program optimization and hardware co-design?
- How to make auto-tuning, machine learning and run-time adaptation practical?
- How to ensure reproducibility of experimental results?

**Share the whole experimental setup
with all related artifacts, SW/HW dependencies,
and unified meta-information**

Tools are not prepared for
auto-tuning and
adaptation!

Users struggle exposing
this meta information

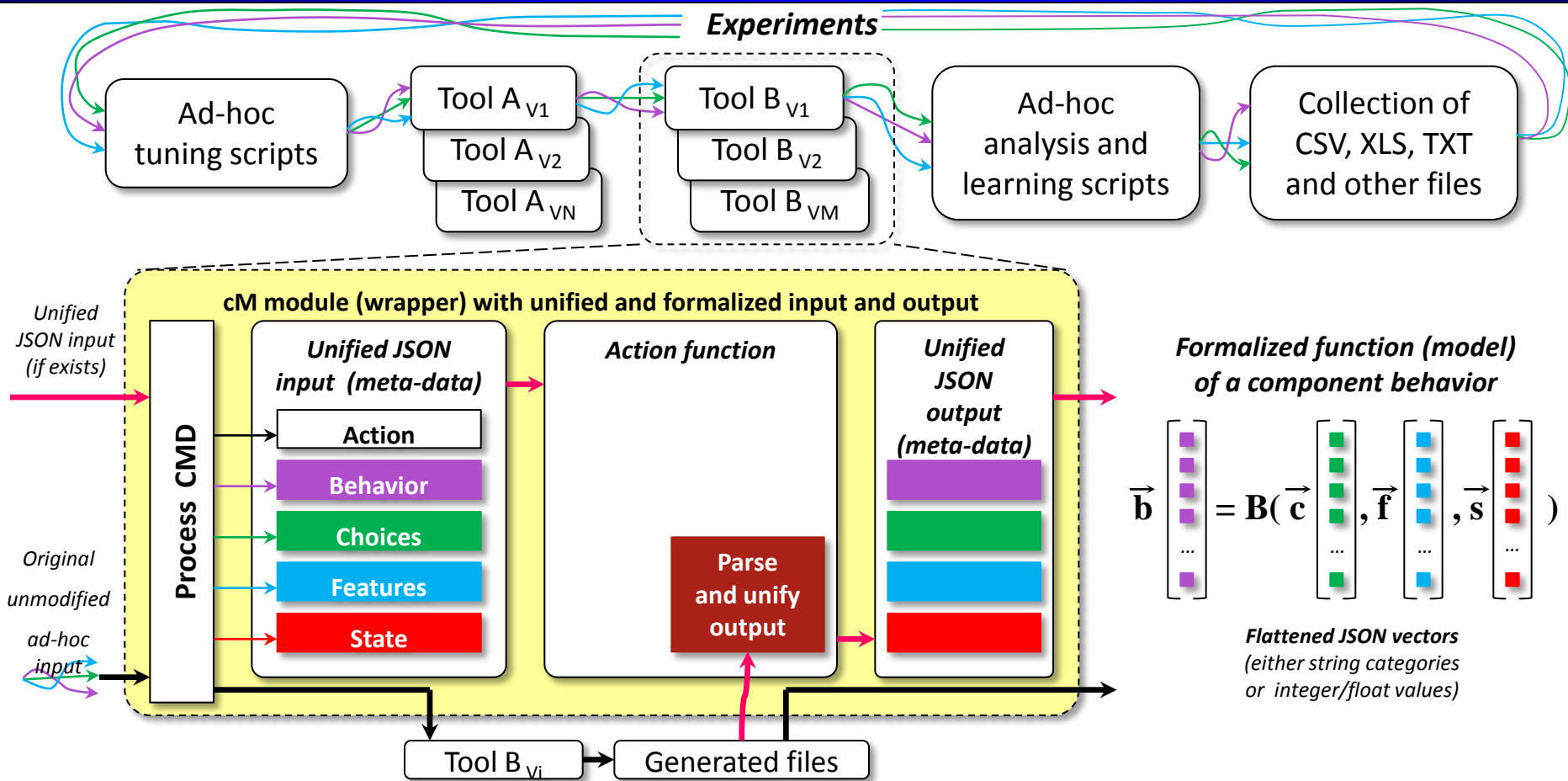
Wrappers around code and data (linked together)



cm **[module name]** **[action]** (param₁=value₁ param₂=value₂ ... -- unparsed command line)
cm **compiler** build -- icc -fast *.c
cm **code.source** build ct_compiler=icc13 ct_optimizations=-fast
cm **code** run os=android binary=./a.out dataset=image-crazy-scientist.pgm

Should be able to run on any OS (Windows, Linux, Android, MacOS, etc)!

Exposing meta information in a unified way



cm **[module name]** **[action]** (param₁=value₁ param₂=value₂ ... -- unparsed command line)

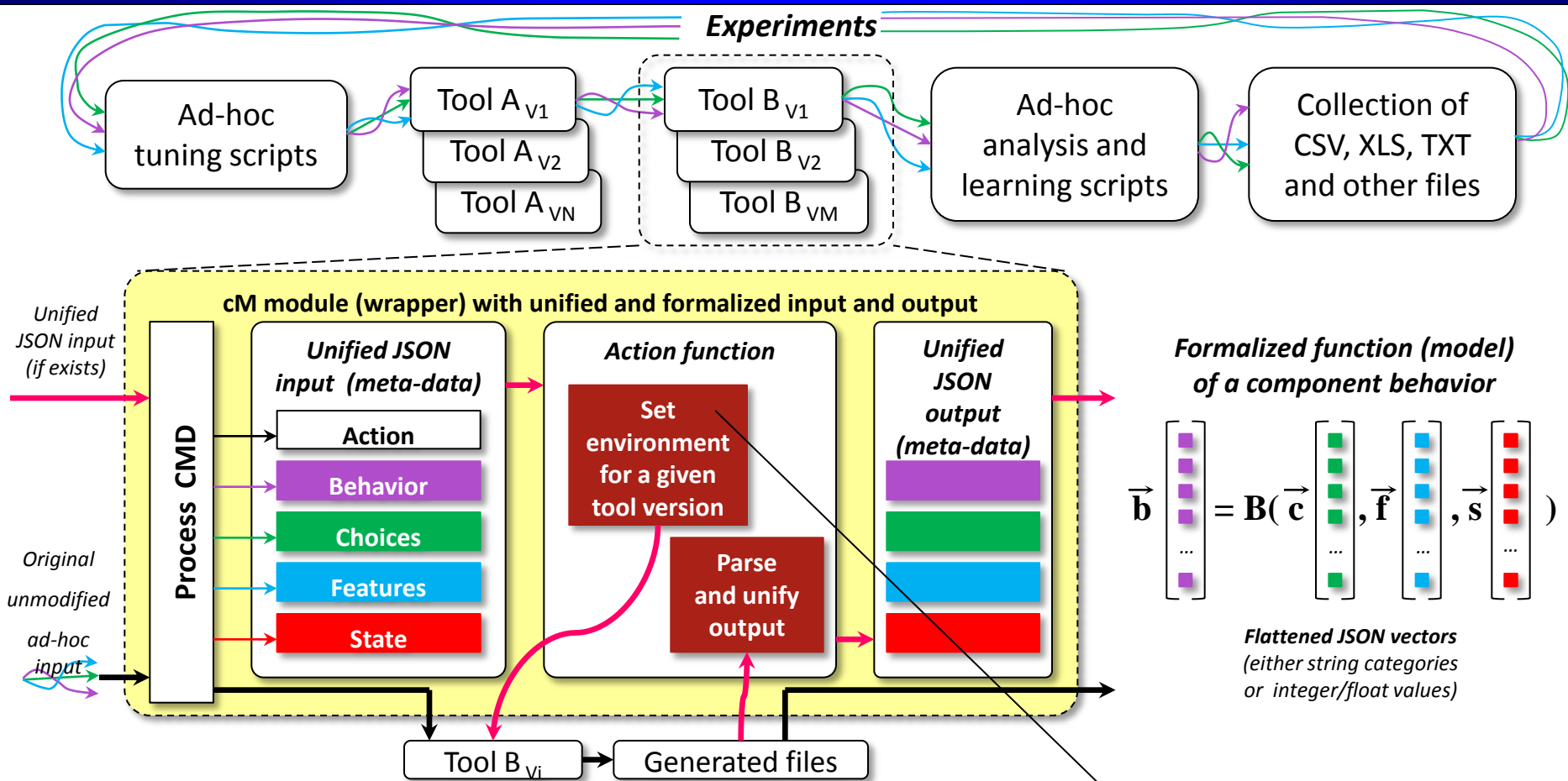
cm **compiler** build -- icc -fast *.c

cm **code.source** build ct_compiler=icc13 ct_optimizations=-fast

cm **code** run os=android binary=./a.out dataset=image-crazy-scientist.pgm

Should be able to run on any OS (Windows, Linux, Android, MacOS, etc)!

Adding SW/HW dependencies check



cm **[module name]** **[action]** (param₁=value₁ param₂=value₂ ... -- unparsed command line)

cm **compiler** build -- icc -fast *.c

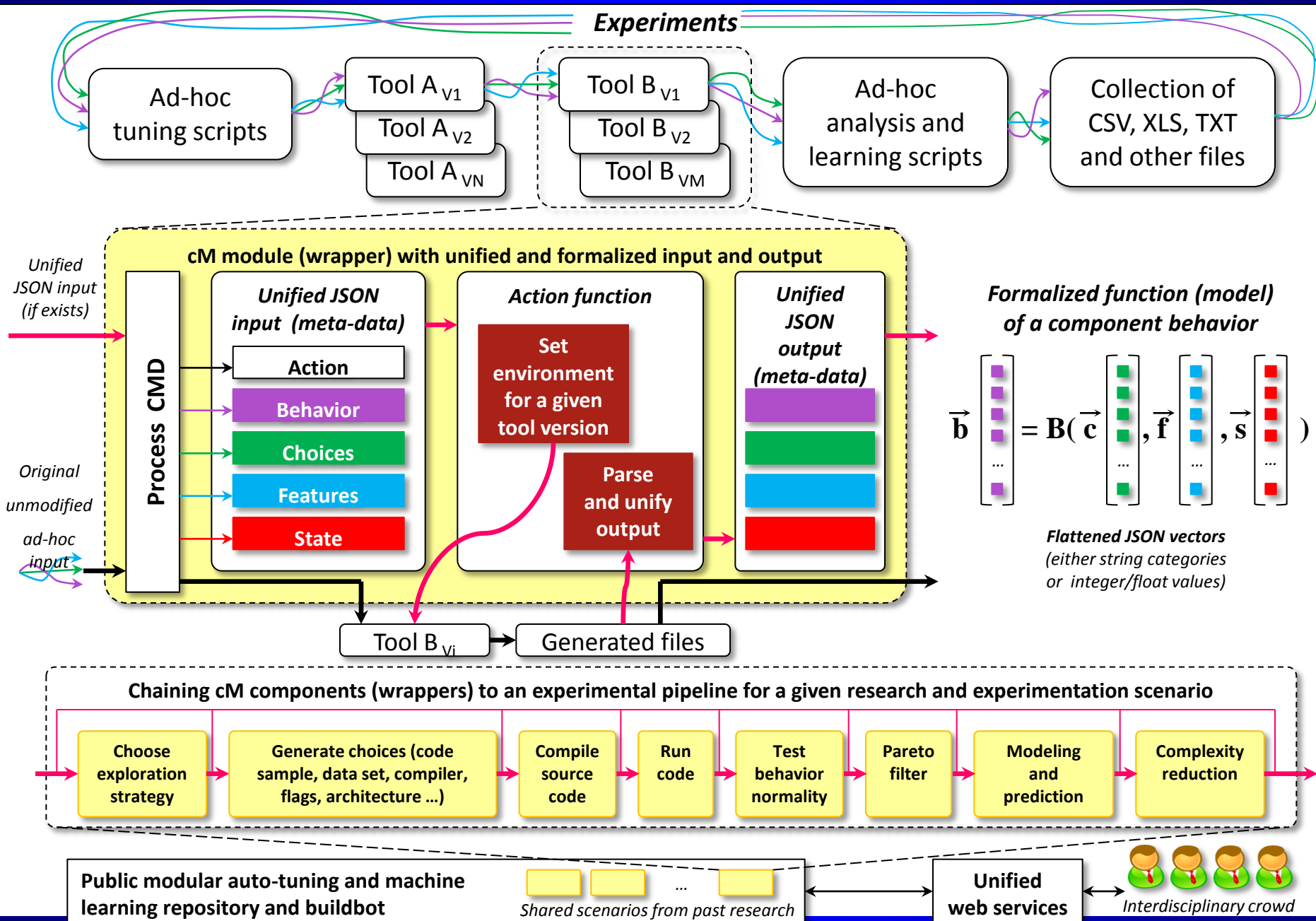
cm **code.source** build ct_compiler=icc13 ct_optimizations=-fast

cm **code** run os=android binary=./a.out dataset=image-crazy-scientist.pgm

Should be able to run on any OS (Windows, Linux, Android, MacOS, etc)!

Multiple tool versions can co-exist, while their interface is abstracted by cM module

Assembling, preserving, sharing and extending the whole pipeline as "LEGO"



Reproducibility of experimental results

Reproducibility came as a side effect!

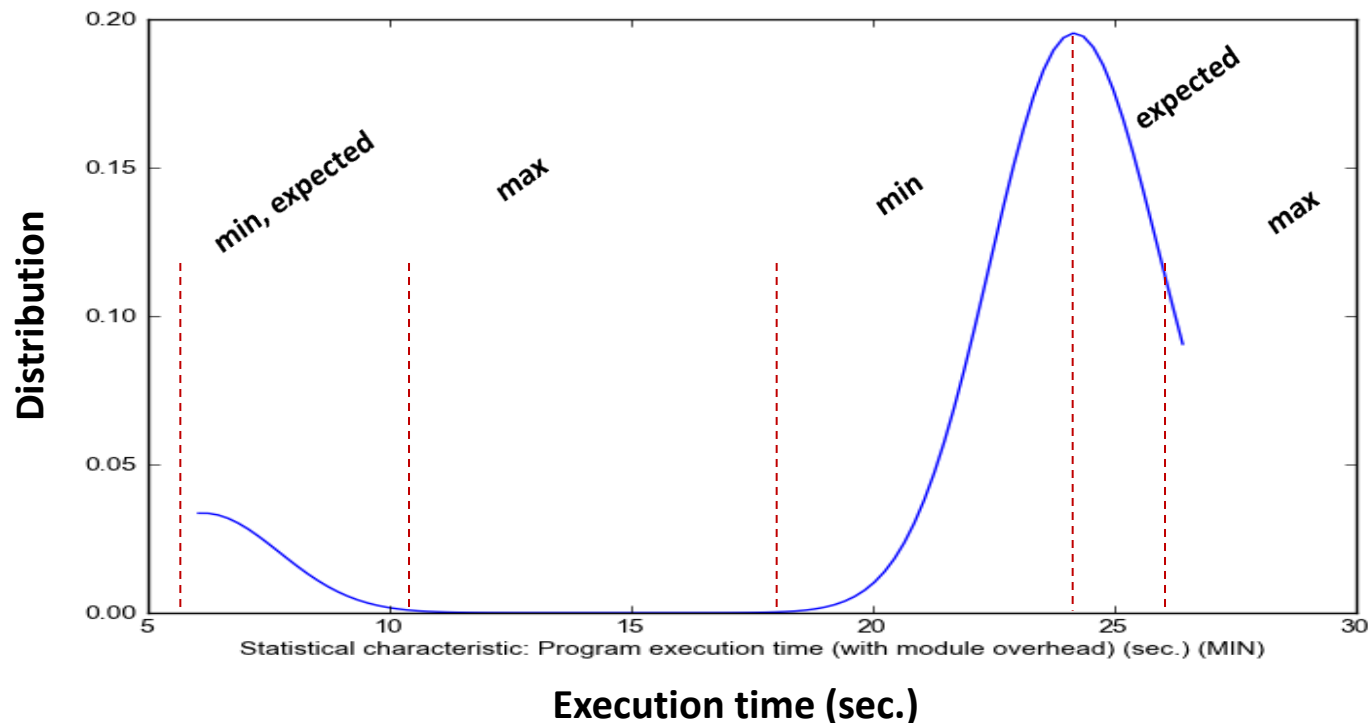
- Can preserve the whole experimental setup with all data and software dependencies
 - Can perform statistical analysis (normality test) for characteristics
- Community can add missing features or improve machine learning models

Reproducibility of experimental results

Reproducibility came as a side effect!

- Can preserve the whole experimental setup with all data and software dependencies
 - Can perform statistical analysis (normality test) for characteristics
- Community can add missing features or improve machine learning models

Unexpected behavior - expose to the community including domain specialists, explain, find missing feature and add to the system

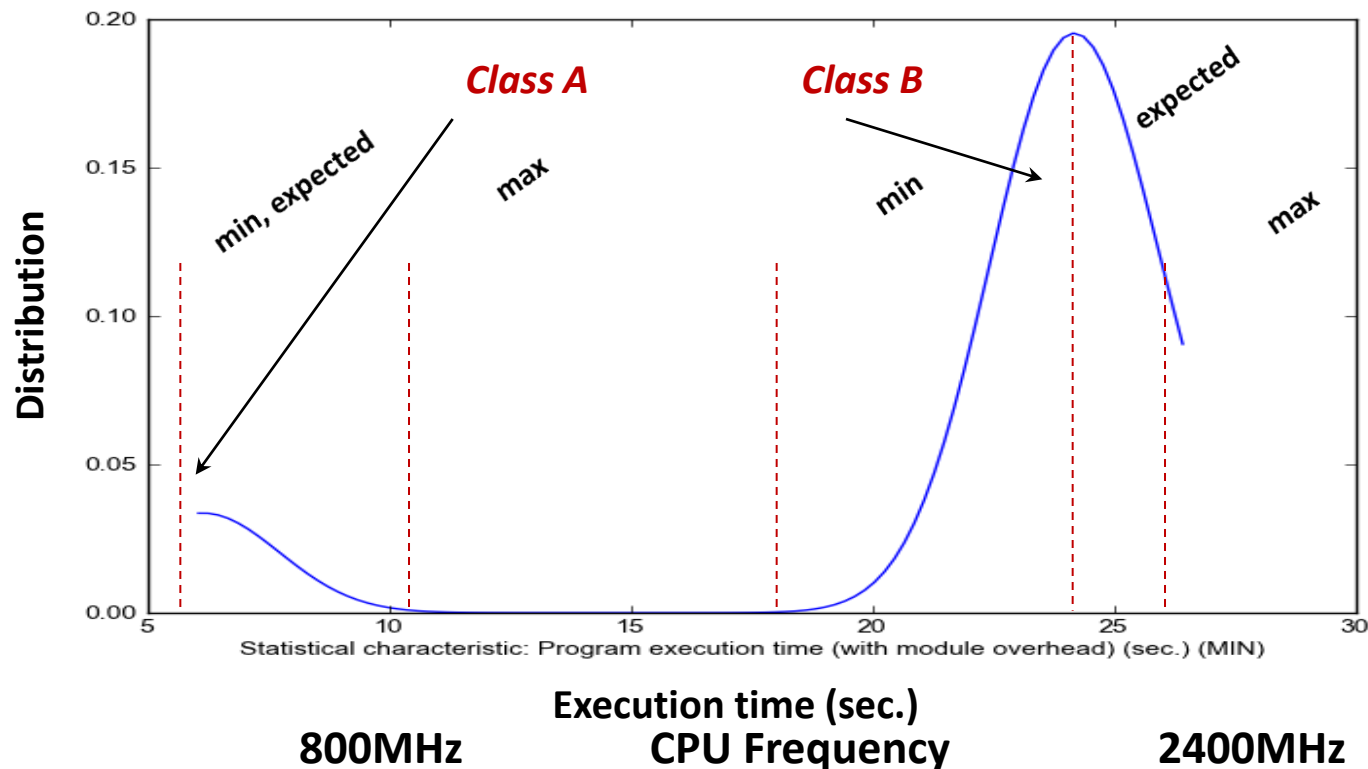


Reproducibility of experimental results

Reproducibility came as a side effect!

- Can preserve the whole experimental setup with all data and software dependencies
 - Can perform statistical analysis (normality test) for characteristics
- Community can add missing features or improve machine learning models

Unexpected behavior - expose to the community including domain specialists, explain, find missing feature and add to the system



Proposal: crowdsourcing validation of publications and research results

Plan to validate at ADAPT'15 (adapt-workshop.org)
workshop on adaptive, self-tuning computing systems



- Submit papers to open access archives (arXiv, HAL, etc)
- Make all related research material either at the personal website or at public sharing services
- Initiate discussion at social networking sites with ranking (Reddit, SlashDot, StackExchange) or without (Google+, Facebook)
- Arrange first small program committee that monitors discussions to filter obviously wrong, unreproducible or possibly plagiarized
- Select a set of “interesting” papers and send it to a interdisciplinary program committee based on paper topics and public discussions

Proposal: crowdsourcing validation of publications and research results

- Select final papers based on public discussions and professional reviews
- Create an open access reproducible online journal with all related materials from the most interesting, advanced and highest ranked publications
- Send considerably updated papers to traditional journals (not to break current system but make open access and traditional publication models co-exist)

Grigori Fursin and Christophe Dubach.

Experience report: community-driven reviewing and validation of publications.

April 2014

ACM Digital Library - dl.acm.org/citation.cfm?id=2618142

Questions? Feedback? Discussions?