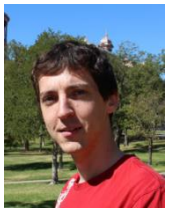
Performance-Influence Models: Prediction, Optimization, Debugging


Norbert Siegmund


Alexander Grebhahn

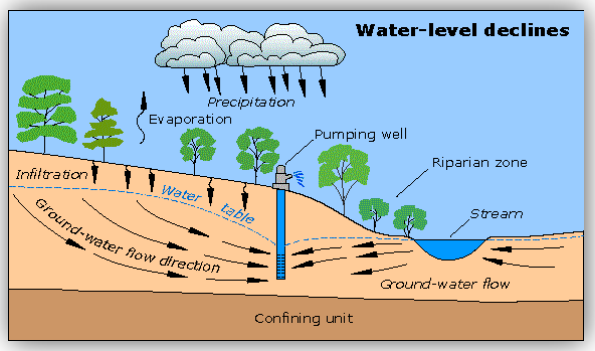

Sven Apel

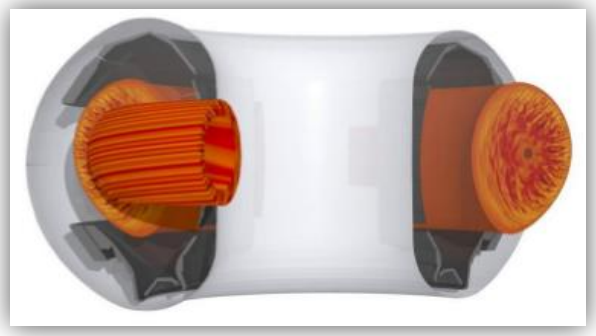

Christian Kästner

UNIVERSITÄT PASSAU

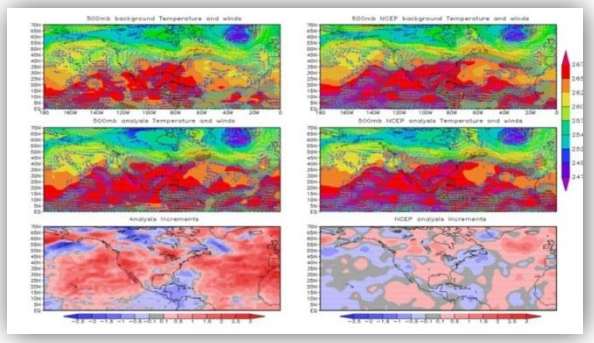
Carnegie Mellon University

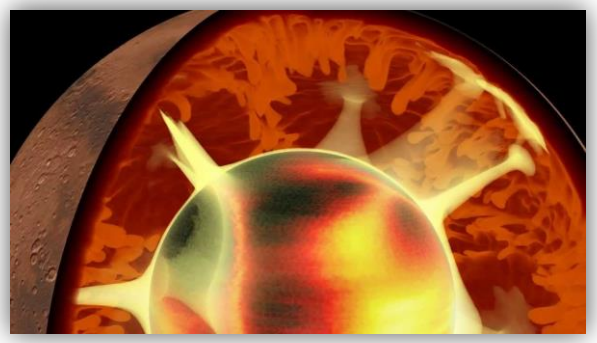
Even Domain Experts Struggle with Performance!



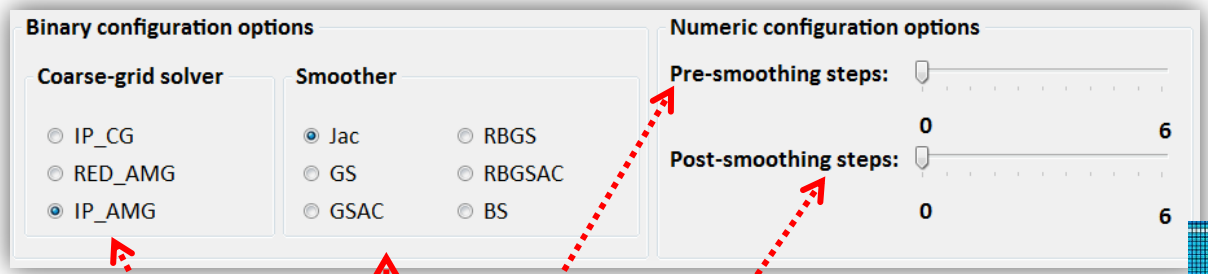


HPC Experts Meeting at Dagstuhl

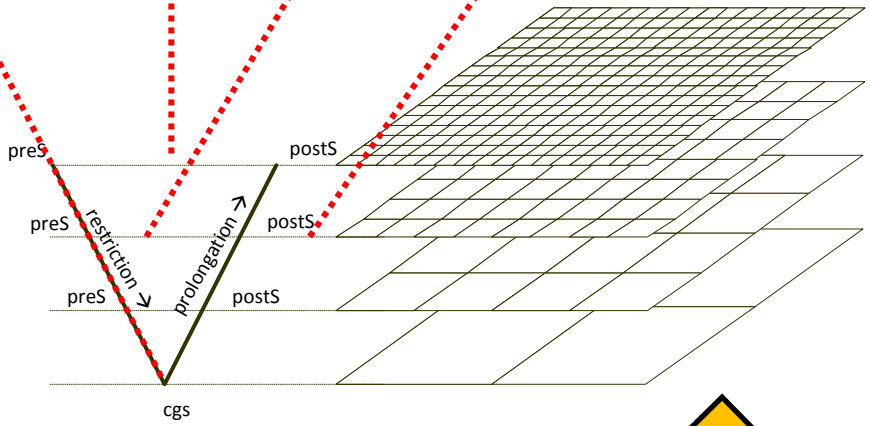




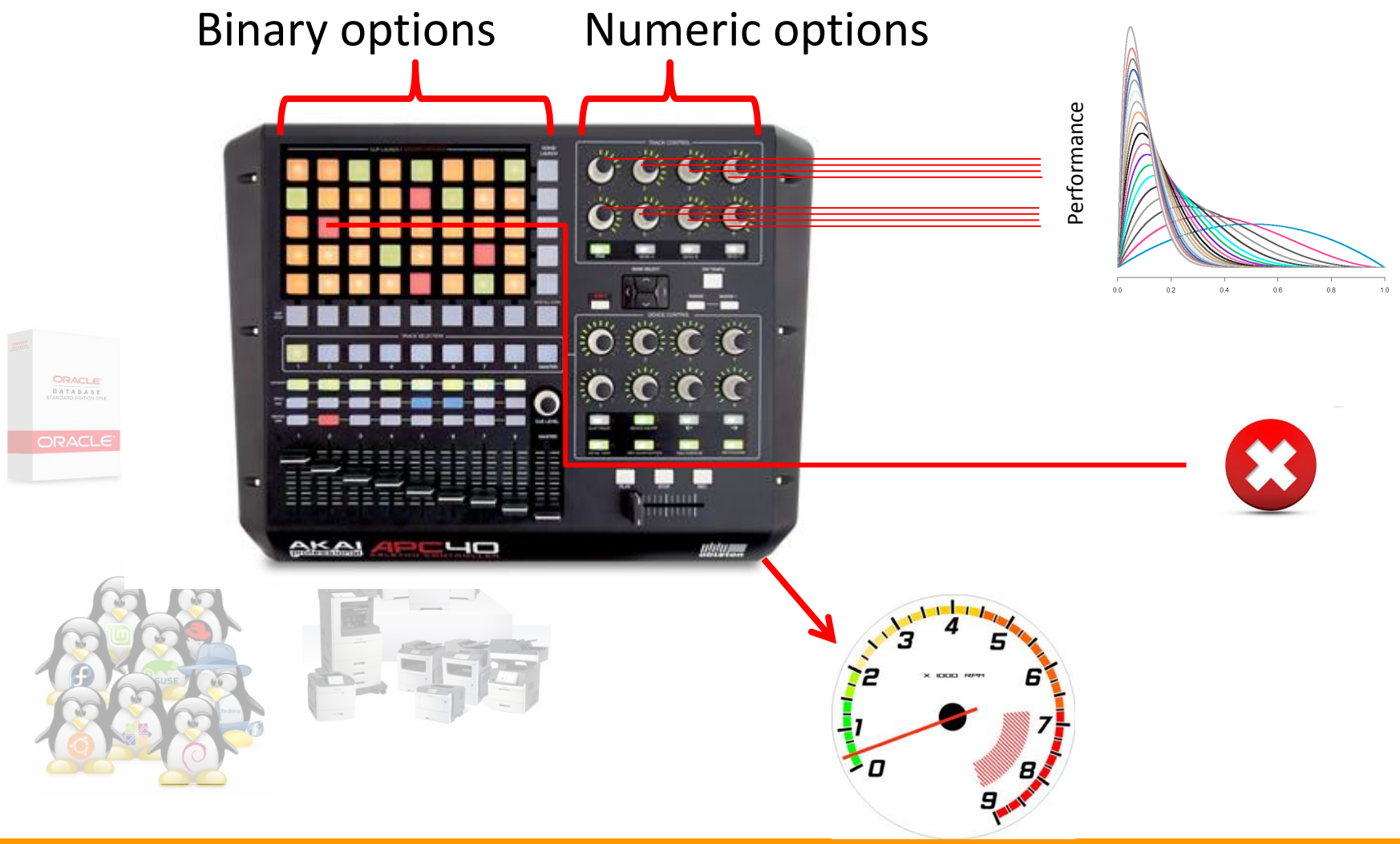
Their Problem: Finding the Optimal Configuration for a Given Hardware Platform?



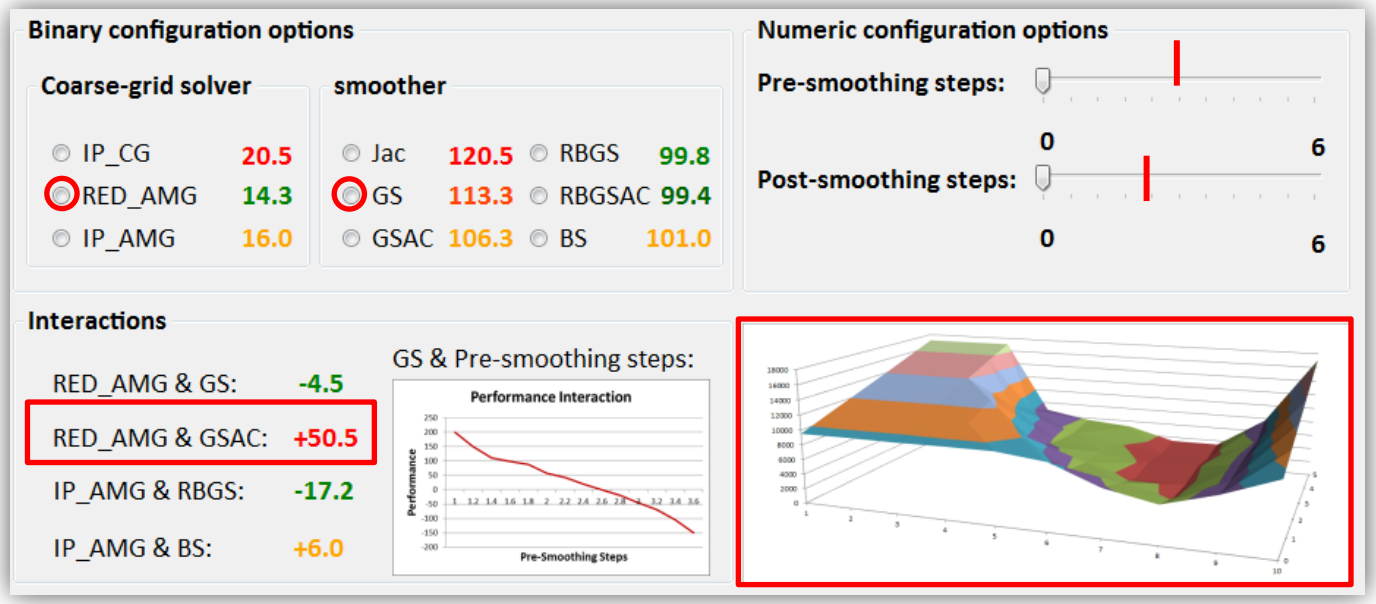
Stencil code



What is the Influence of Configuration Options on Performance?



Vision: Performance-Influence Models

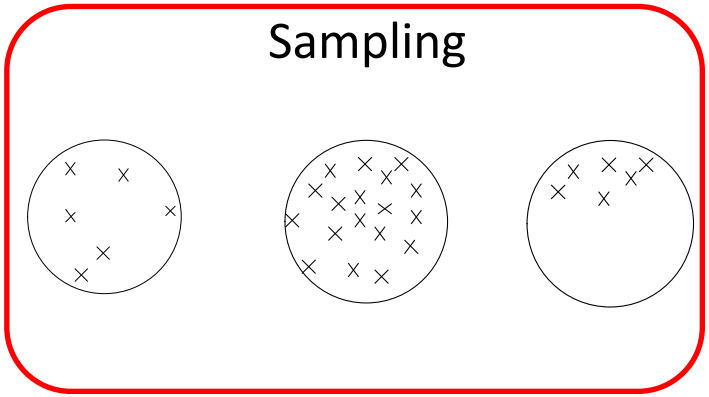


Determine the influence of configuration options and their interactions and use it for:

- Understanding
- Debugging
- Prediction and optimization

14.3 + 113.3 - 4.5 - 73.9 + 171.7 = 219.9

Learning Procedure

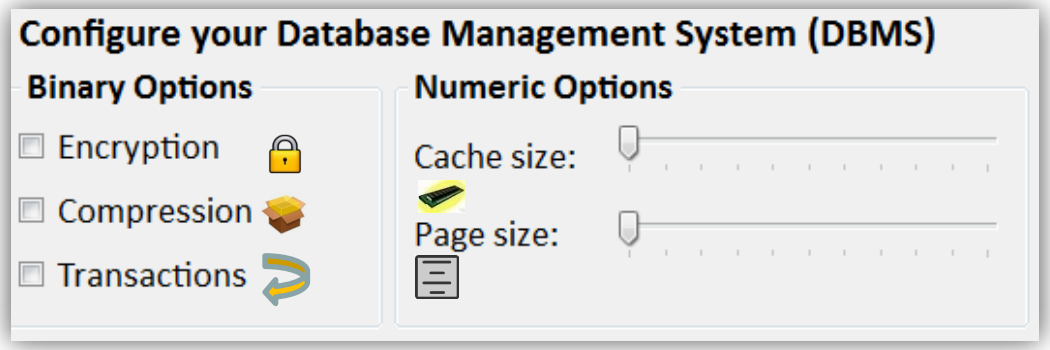


Learning

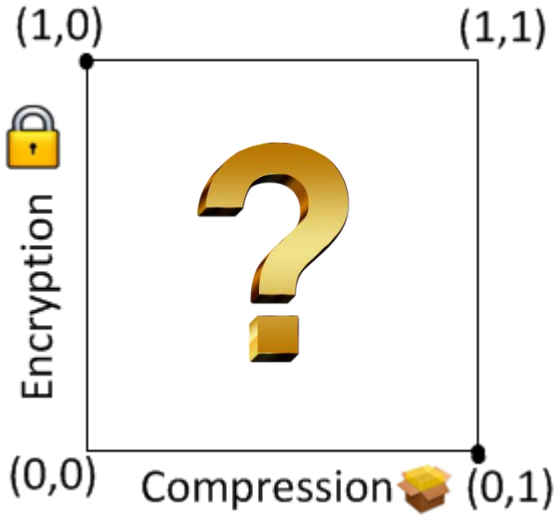
Performance-Influence Model

189392.754871324 * root + -5341.1497460973 * cells + -
1097.39668559875 * post + 6852.5884693721 * GradientSolver +
752.542387648181 * BiCGSTABSolver + 1.88645994097499 * cells * post *
post + 547.236997276589 * GradientSolver * pre * pre + -
113.547148735501 * cells * GradientSolver + 0.706383967601375 * cells *
cells * cells + 385.840181761216 * GradientSolver * post * post +
2.727073632198 99 * cells * pre + -39.9822314861407 * BiCGSTABSolver *
pre * pre + -11.9701281275527 * cells * GradientSolver * pre * pre + -
8.2288112361045 * cells * GradientSolver * post * post +
112.152829956782 * post * SeqSOR

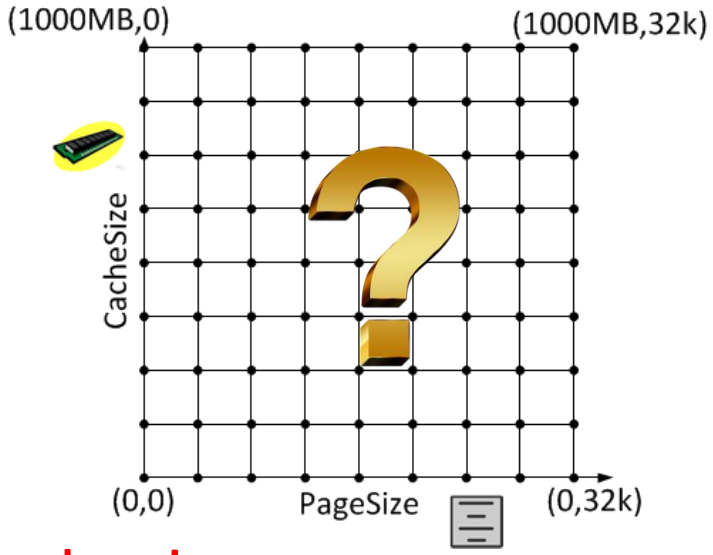
Sampling Binary and Numeric Options



Binary Options



Numeric Options



Exponential number!

Structured sampling approaches for the different kinds of options

Heuristics for Binary-Option Sampling


Predicting Performance via Automated Feature-Interaction Detection

Siegmund et al., ICSE'12

- Random?
  - Unlikely to select a valid configuration
  - Locally clustered solutions using SAT

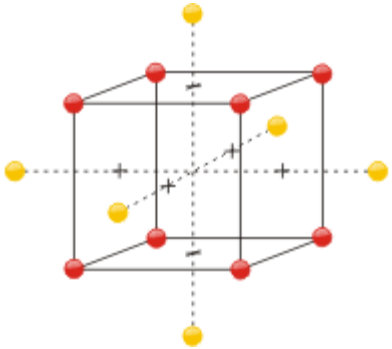
- Heuristics
  - Option-Wise (OW): {🔒}, {📦}, {🔄}, {🛢}, {}
  - Negative Option-Wise (nOW):

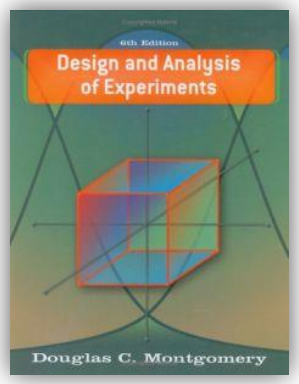
    {🔒📦🔄🛢}, {🔒📦🔄}, {🔒📦🛢}, {🔒🔄🛢}, {📦🔄🛢}
  - Pair-Wise (PW) : {🔒📦}, {📦🔄}, {🔄🛢}, {🔒🔄}, {🔒🛢},

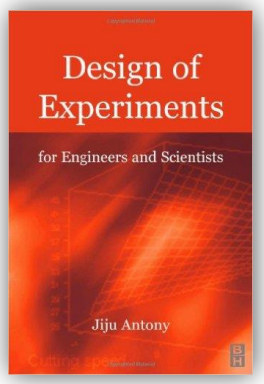
    {📦🛢}

Numeric-Option Sampling (Experimental Designs)

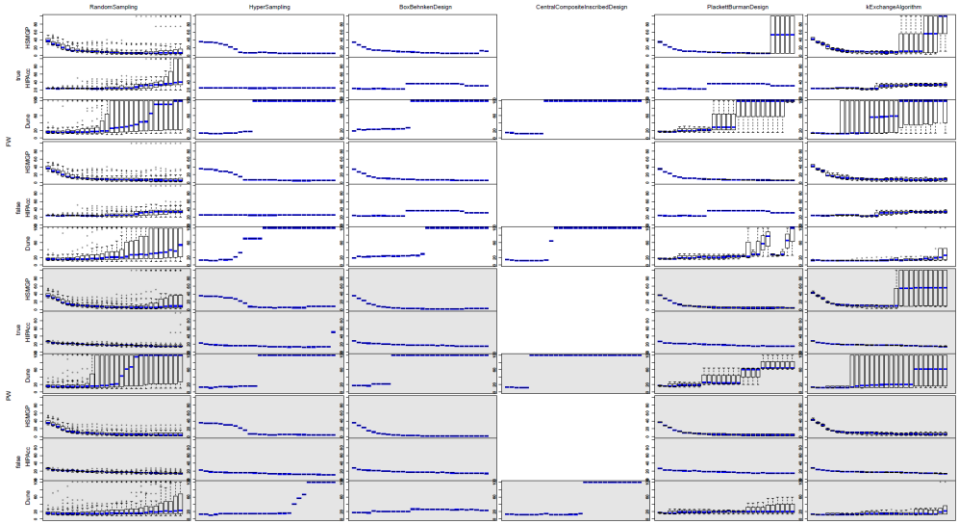
- Fractional factorial designs
- Optimal designs



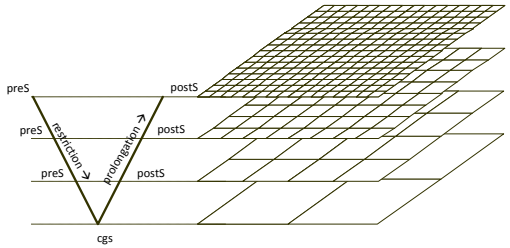


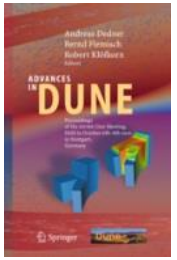


- Pre-study:



Multi-grid solver as subject systems



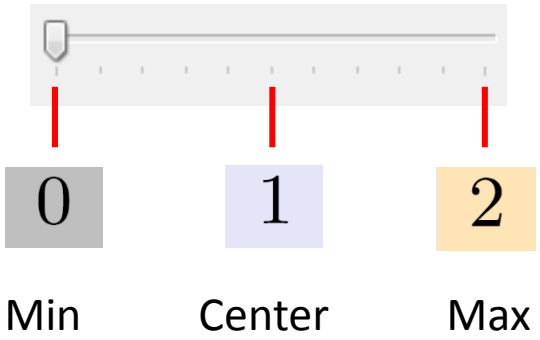


Plackett-Burman Design as best design

Plackett-Burman Design (PBD)

- Minimizes the variance of the estimates of the independent variables (numeric options)
- …while using a limited number of measurements
- Design specifies *seeds* depending on the number of experiments to be conducted (i.e., configurations to be measured)

Value range of a numeric option



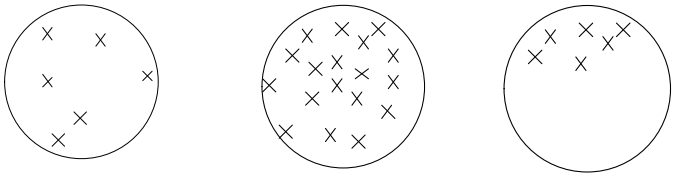
Numeric options

Configurations

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8
c_1	0	1	1	2	0	2	2	1
c_2	1	0	1	1	2	0	2	2
c_3	2	1	0	1	1	2	0	2
c_4	2	2	1	0	1	1	2	0
c_5	0	2	2	1	0	1	1	2
c_6	2	0	2	2	1	0	1	1
c_7	1	2	0	2	2	1	0	1
c_8	1	1	2	0	2	2	1	0
c_9	0	0	0	0	0	0	0	0

Learning Procedure

Sampling



Learning

Performance-Influence Model

189392.754871324 * root + -5341.1497460973 * cells + -
1097.39668559875 * post + 6852.58846993721 * GradientSolver +
752.542387648181 * BiCGSTABSolver + 1.88645994097499 * cells * post *
post + 547.236997276589 * GradientSolver * pre * pre + -
113.547148735501 * cells * GradientSolver + 0.706383967601375 * cells *
cells * cells + 385.840181761216 * GradientSolver * post * post +
2.727073632198 99 * cells * pre + -39.9822314861407 * BiCGSTABSolver *
pre * pre + -11.9701281275527 * cells * GradientSolver * pre * pre + -
8.2288112361045 * cells * GradientSolver * post * post +
112.152829956782 * post * SeqSOR

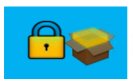
Regression Learning

Exponential number!

Unlimited candidates!

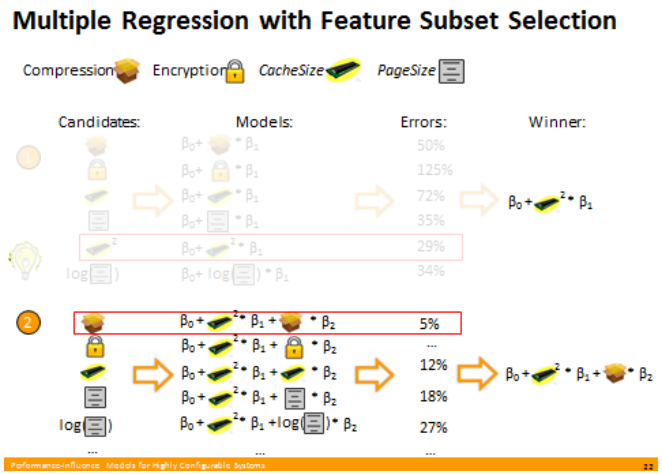
Individual Options | Interactions | Functions

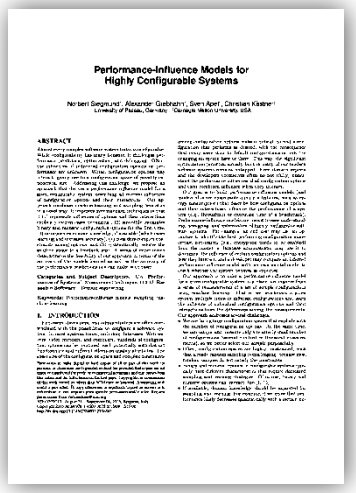
Configurations

											Perf.
1	0	0	20	0	0	0	0	400	N/A		833
0	1	1	50	16	0	0	800	2500	1.2		411
1	0	1	100	32	0	0	0	10000	0.9		290
1	1	0	50	32	1	0	1600	2500	1.5	=	799
1	1	1	20	32	1	1	640	400	1.5		753
0	0	0	100	0	0	0	0	10000	N/A		514
1	1	1	100	16	1	1	1600	10000	1.2		416
*	*	*	*	*	*	*	*	*	*		
β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}		Error:
102.4	84.3	54.1	5.4	1.3							41%
132.3	81.3	56.6	3.5	1.9	-14.1	-5.4	2.4				25%
130.3	83.5	54.2	0.01	0	-14.1	-5.4	1.4	2.1	8.8		7.4%

Multiple Regression with Feature Subset Selection

- Extend model in a stepwise manner
- Probe different candidates
  - Individual influences, interactions, functions

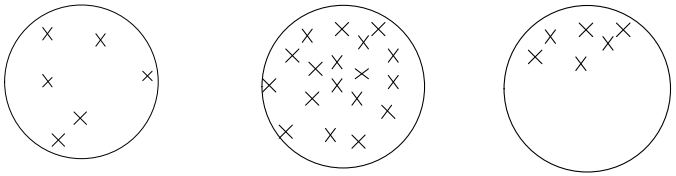




Siegmund et al., FSE'15

Learning Procedure

Sampling

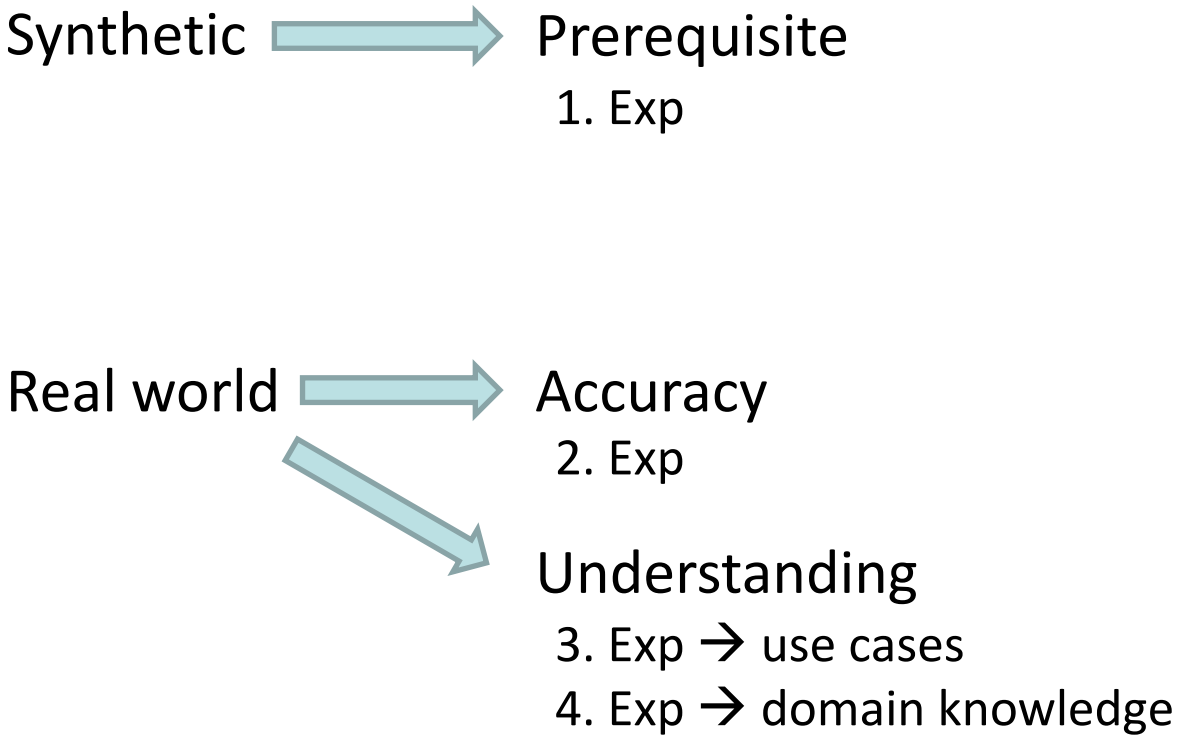


Learning

Performance-Influence Model

189392.754871324 * root + -5341.1497460973 * cells + -
1097.39668559875 * post + 6852.58846993721 * GradientSolver +
752.542387648181 * BiCGSTABSolver + 1.88645994097499 * cells * post *
post + 547.236997276589 * GradientSolver * pre * pre + -
113.547148735501 * cells * GradientSolver + 0.706383967601375 * cells *
cells * cells + 385.840181761216 * GradientSolver * post * post +
2.727073632198 99 * cells * pre + -39.9822314861407 * BiCGSTABSolver *
pre * pre + -11.9701281275527 * cells * GradientSolver * pre * pre + -
8.2288112361045 * cells * GradientSolver * post * post +
112.152829956782 * post * SeqSOR

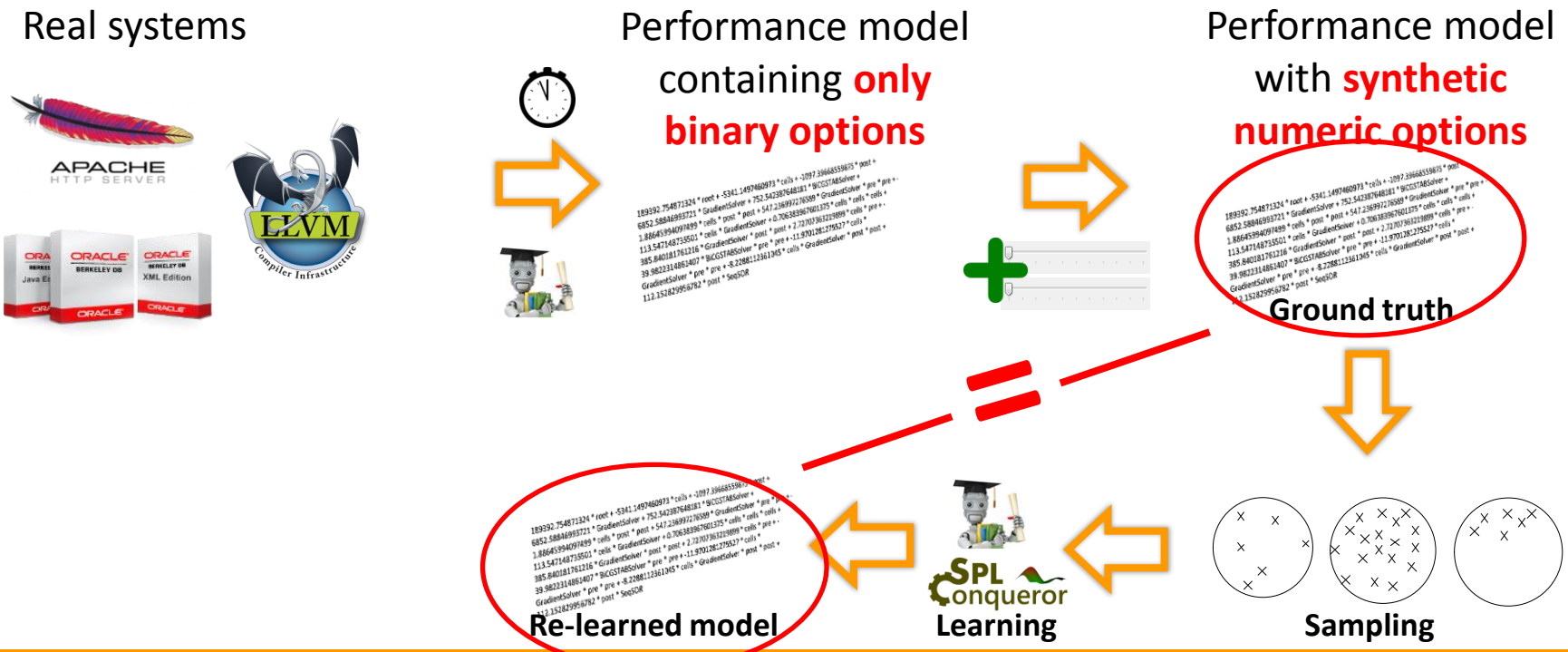
Experimental Evaluation



1. Experiment: Finding the Actual Influences

RQ: Do we find the actually existing influences and interactions?

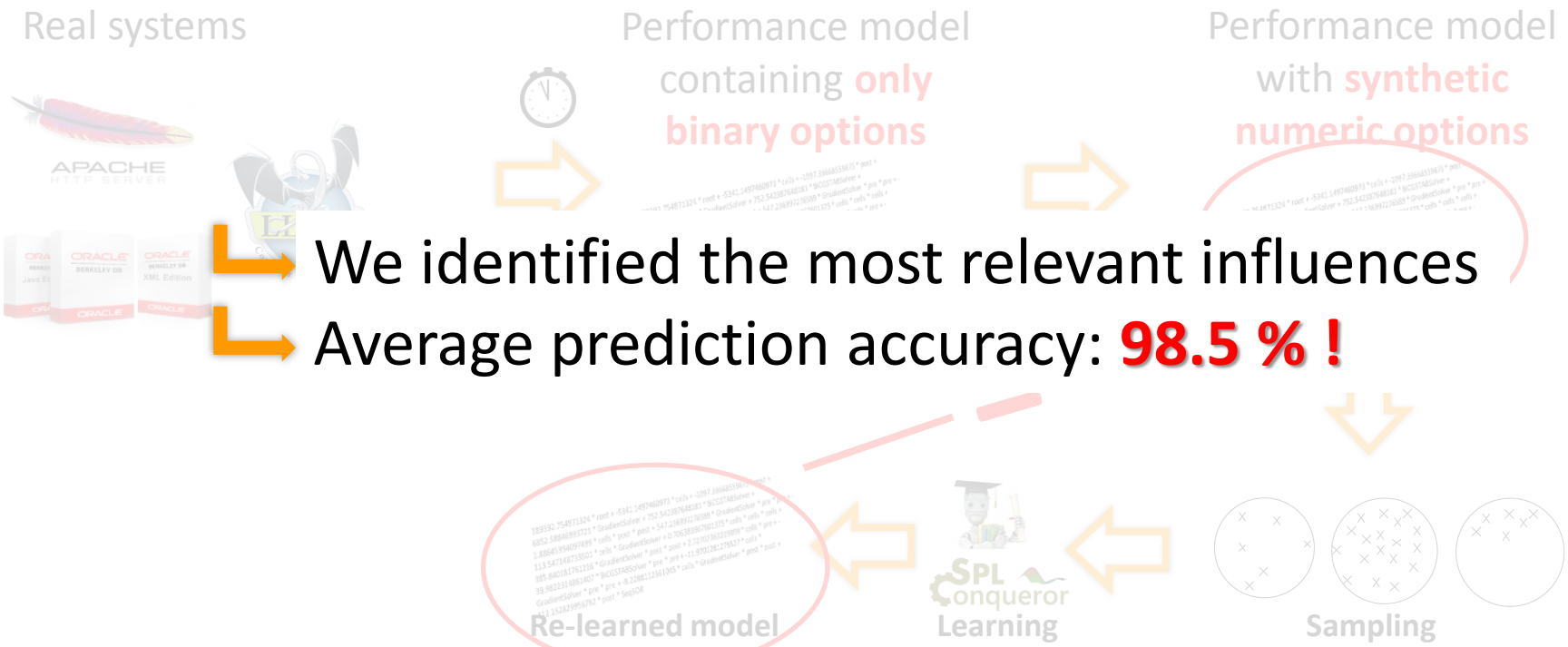
- Design:



1. Experiment: Finding the Actual Influences

RQ: Do we find the actually existing influences and interactions?

- Design:

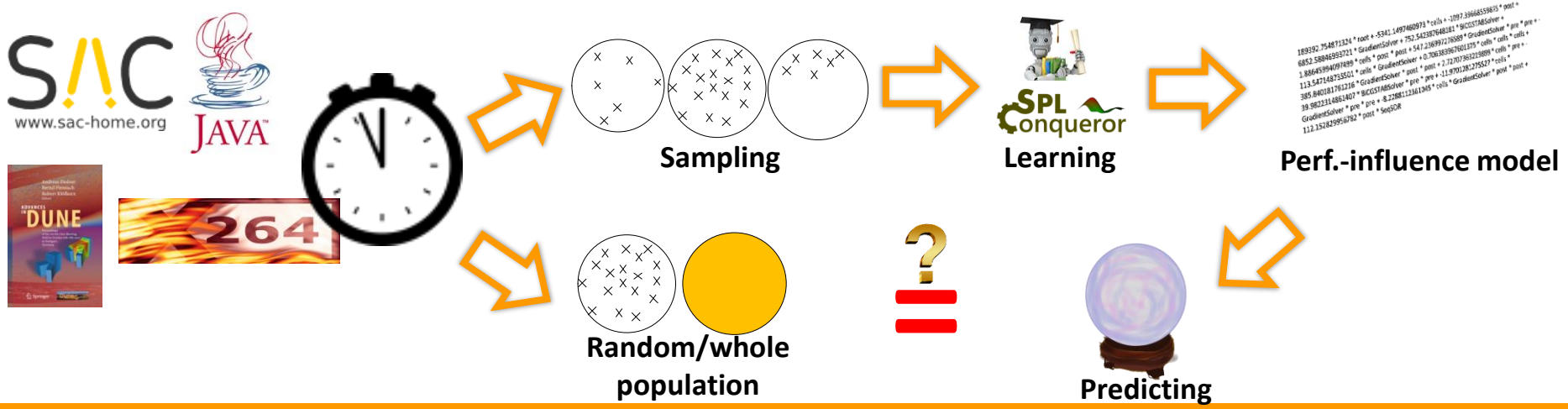


- We identified the most relevant influences
- Average prediction accuracy: **98.5 % !**

2. Experiment: Performance Prediction

RQ: Which combination of sampling approaches achieve the highest prediction accuracy?

System	Domain	# Binary Opt.	# Numeric Opt.	# Constraints	# Configs
Dune MGS	Multi-Grid Solver	8	3	20	2 304
HIPA^{cc}	Image Processing	31	2	416	13 485
HSMGP	Multi-Grid Solver	11	3	45	3 456
JavaGC	Runtime Env.	12	23	4	10^{31}
SaC	Compiler	53	7	10	10^{23}
x264	Video Encoder	8	13	0	10^{27}



2. Experiment: Performance Prediction

RQ: Which combination of sampling approaches achieve the highest prediction accuracy?

System	Domain	# Binary Opt.	# Numeric Opt.	# Constraints	# Configs
Dune MGS	Multi-Grid Solver	8	3	20	2 304
HIPA^{cc}	Image Processing	31	2	416	13 485
HSMGP	Multi-Grid Solver	11	3	45	3 456
JavaGC	Runtime Env.	12	23	4	10^{31}
SaC					0^{23}
x264					0^{27}

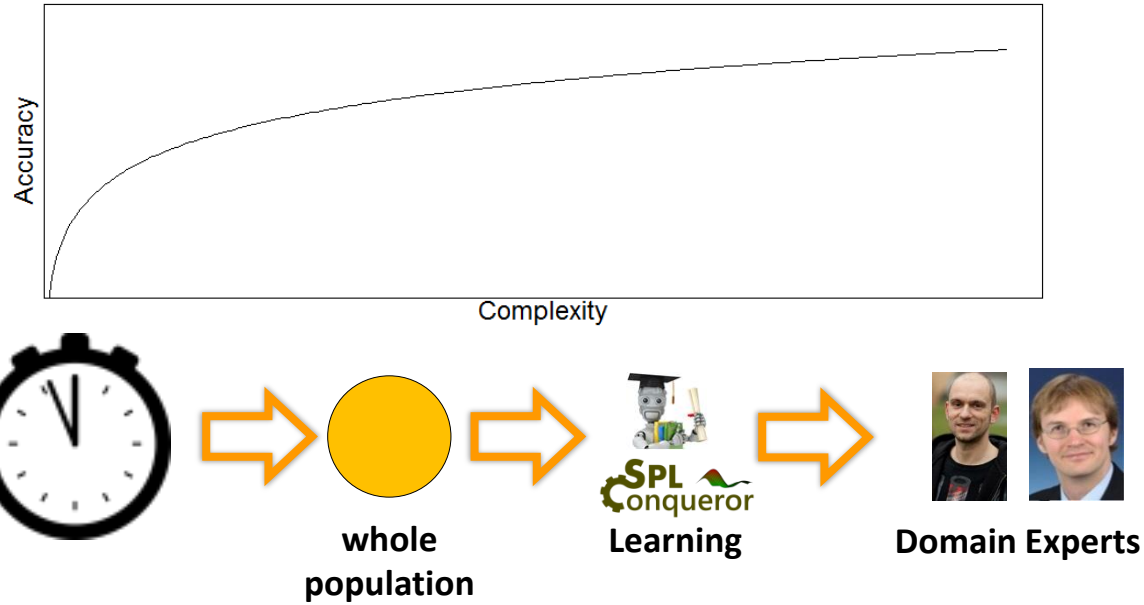
- Average prediction accuracy (PW+PBD): **86.8 % !**
- Experimental designs are better than random sampling

3. Experiment: Accuracy vs. Complexity

Collaboration with:

Sergiy Kolesnikov

- RQ: Is it necessary to learn accurate but complex models?



↳ It depends on the use case. Simple models help to identify the most relevant influences.

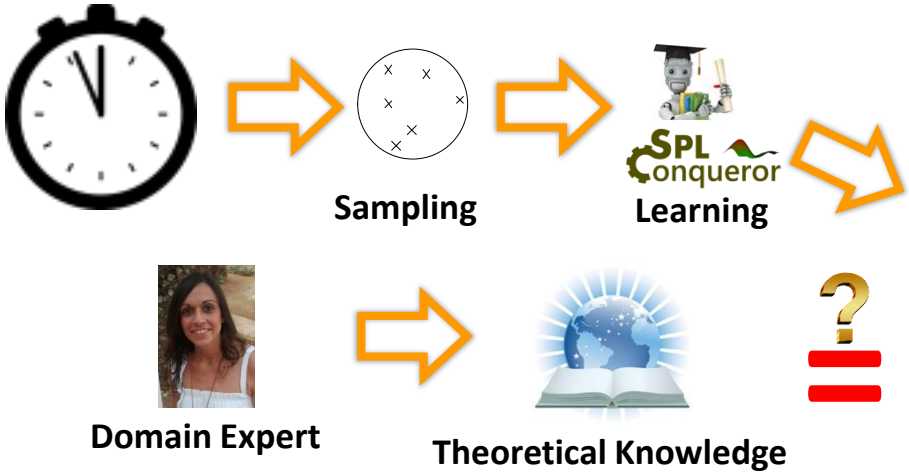
4. Experiment: Validation of Domain Knowledge

Collaboration with:

Carmen Rodrigo

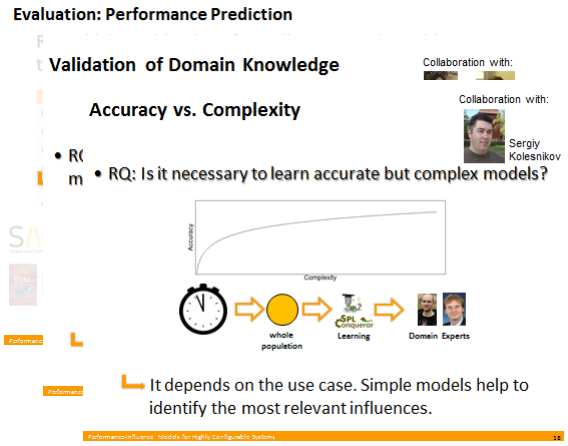
Francisco Gaspar

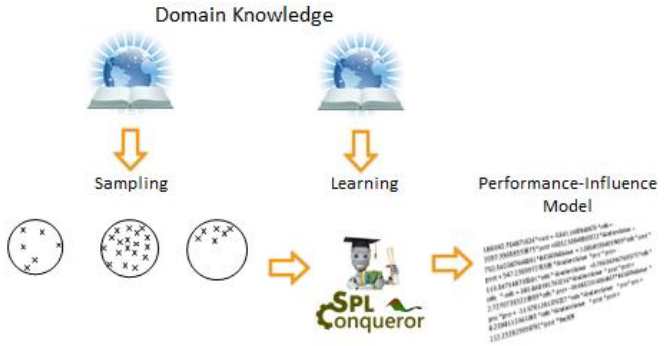
- RQ: Are we able to validate domain knowledge using a model?

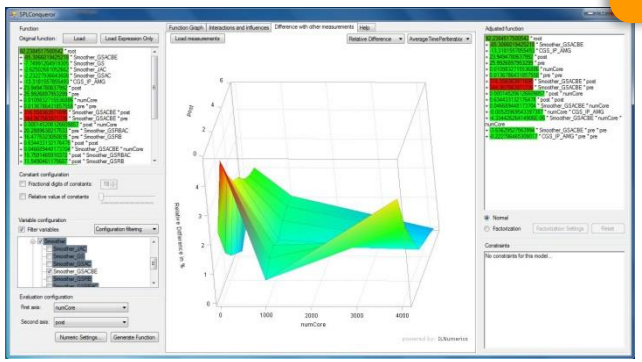


↳ Validate domain knowledge

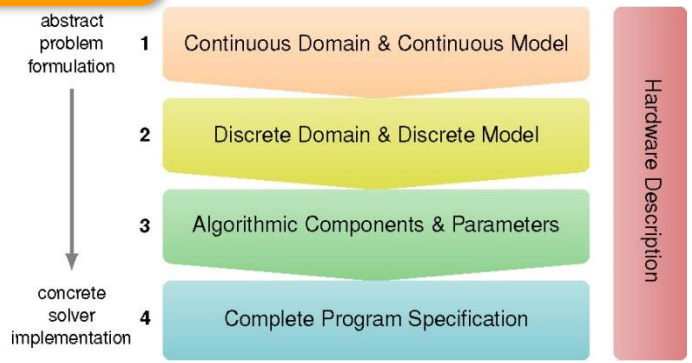
Findings and Future Work







Thank You!



SPL Conqueror

https://github.com/nsiegmun/SPLConqueror
http://fosd.de/SPLConqueror/