



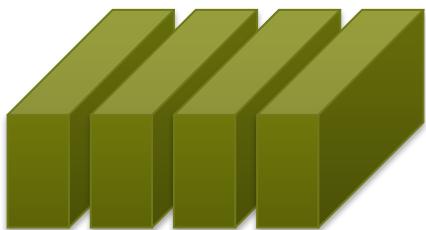
A. Calotoiu<sup>1</sup> (GRS), T. Hoefler (ETH), M. Poke (GRS), F. Wolf (GRS)

# Using Automated Performance Modeling to Find Scalability Bugs in Complex Codes

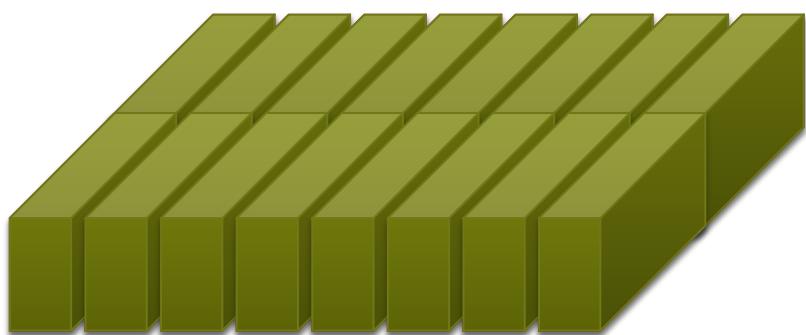


# Latent Scalability Bugs

System size

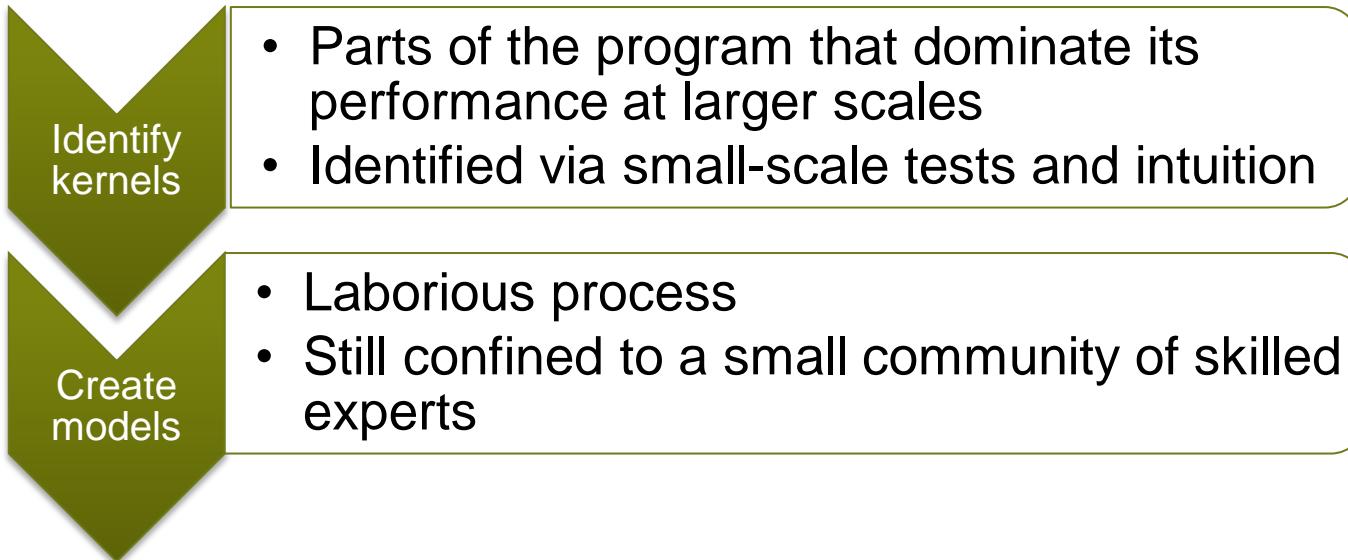


Execution time





# Analytical performance modeling

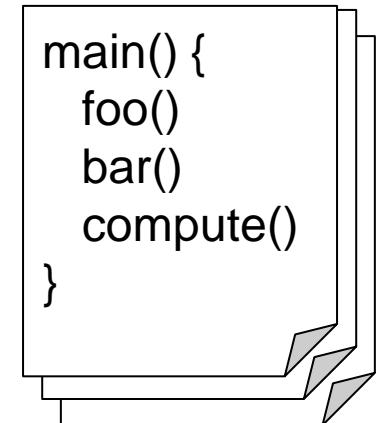


## Disadvantages

- Time consuming
- Danger of overlooking unscalable code



# Scalability bug detector



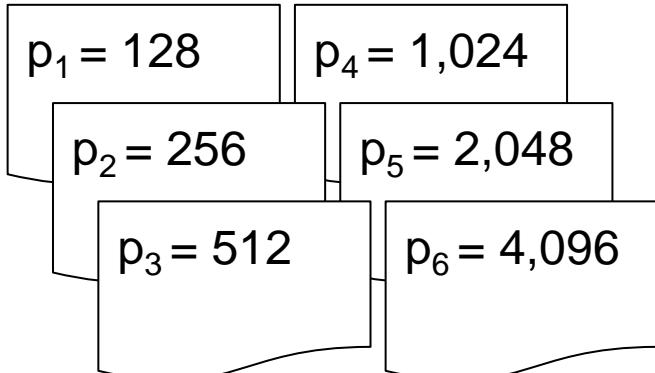
Input

Output

## Instrumentation

- All functions

Performance measurements (profiles)



## Automated modeling

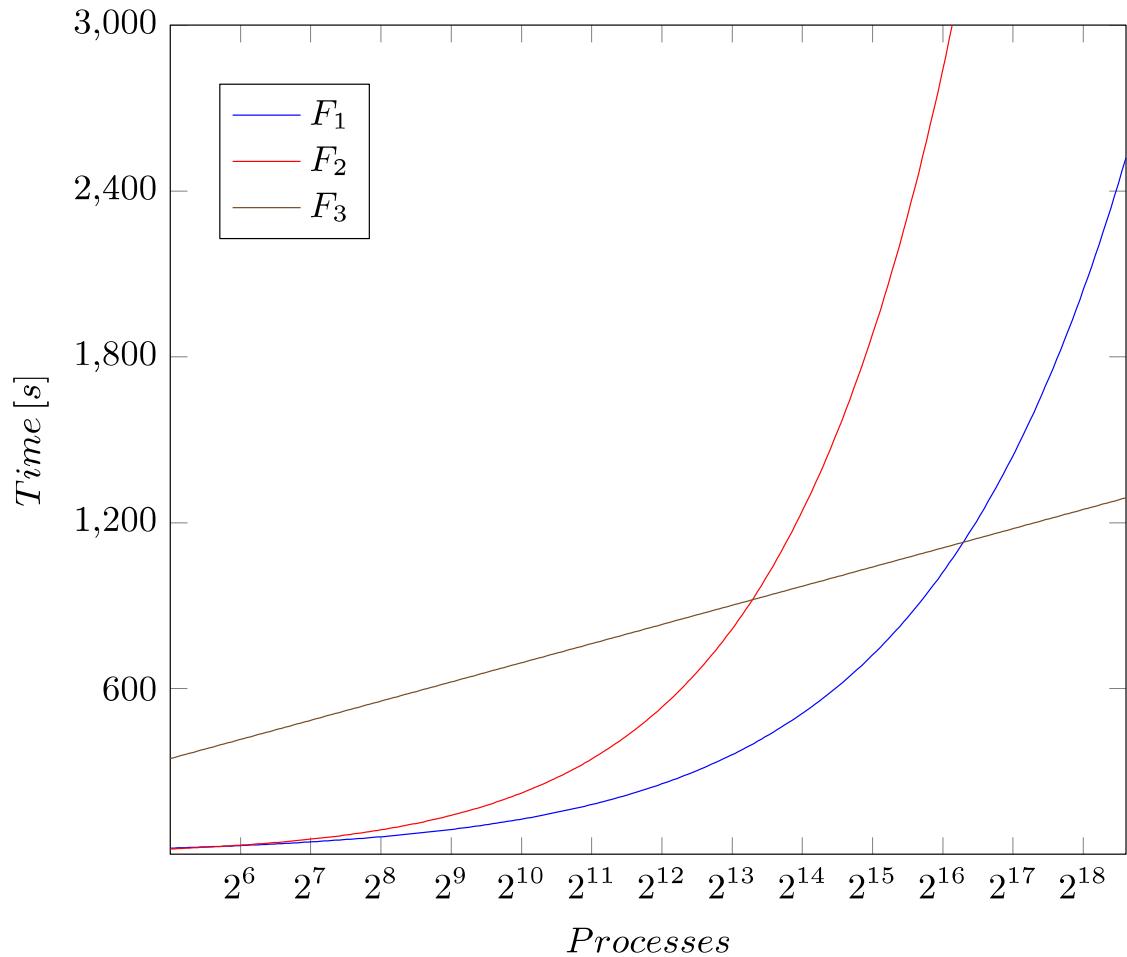
### Ranking:

- Asymptotic
- Target scale  $p_t$

1. foo
2. compute
3. main
4. bar
- [...]



# Primary focusing on scaling trend

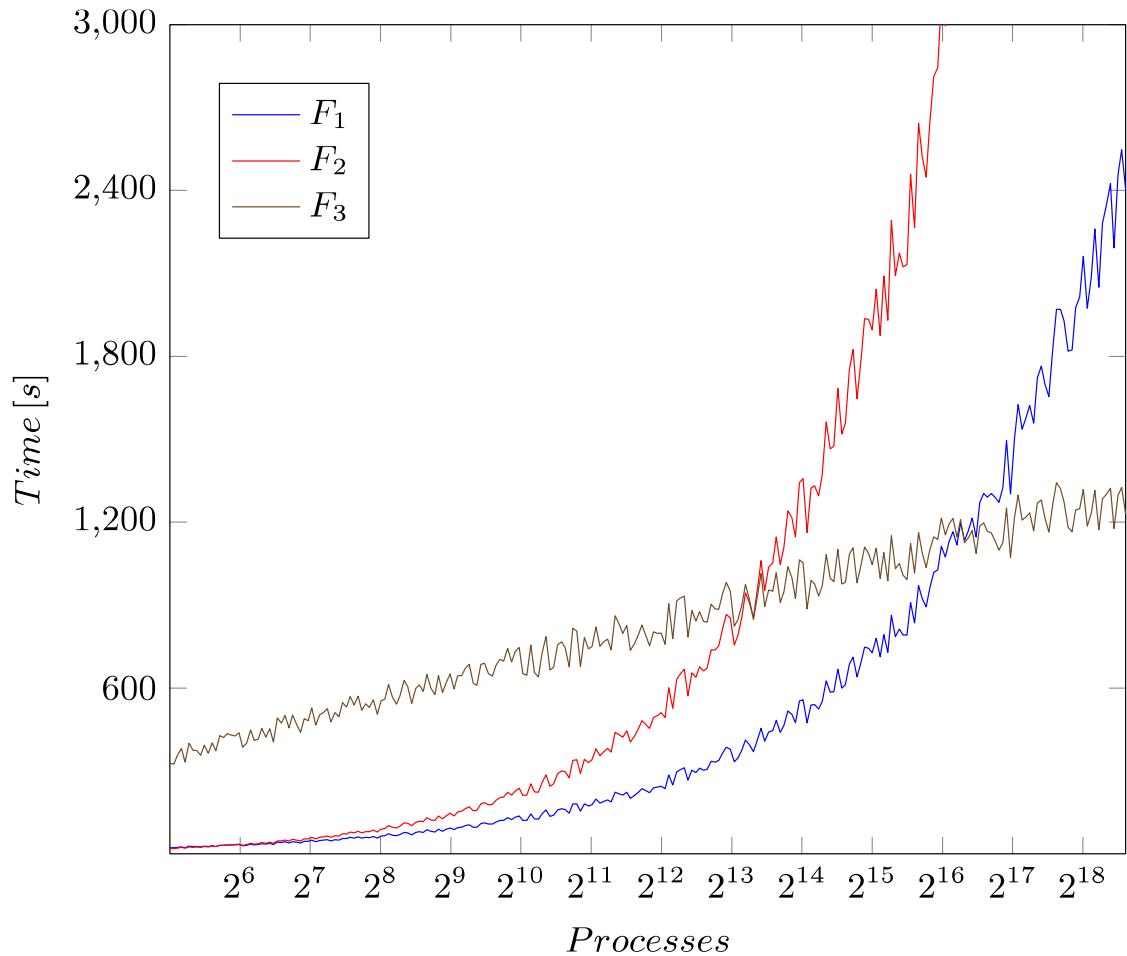


Ranking:

1.  $F_2$
2.  $F_1$
3.  $F_3$



# Primary focusing on scaling trend

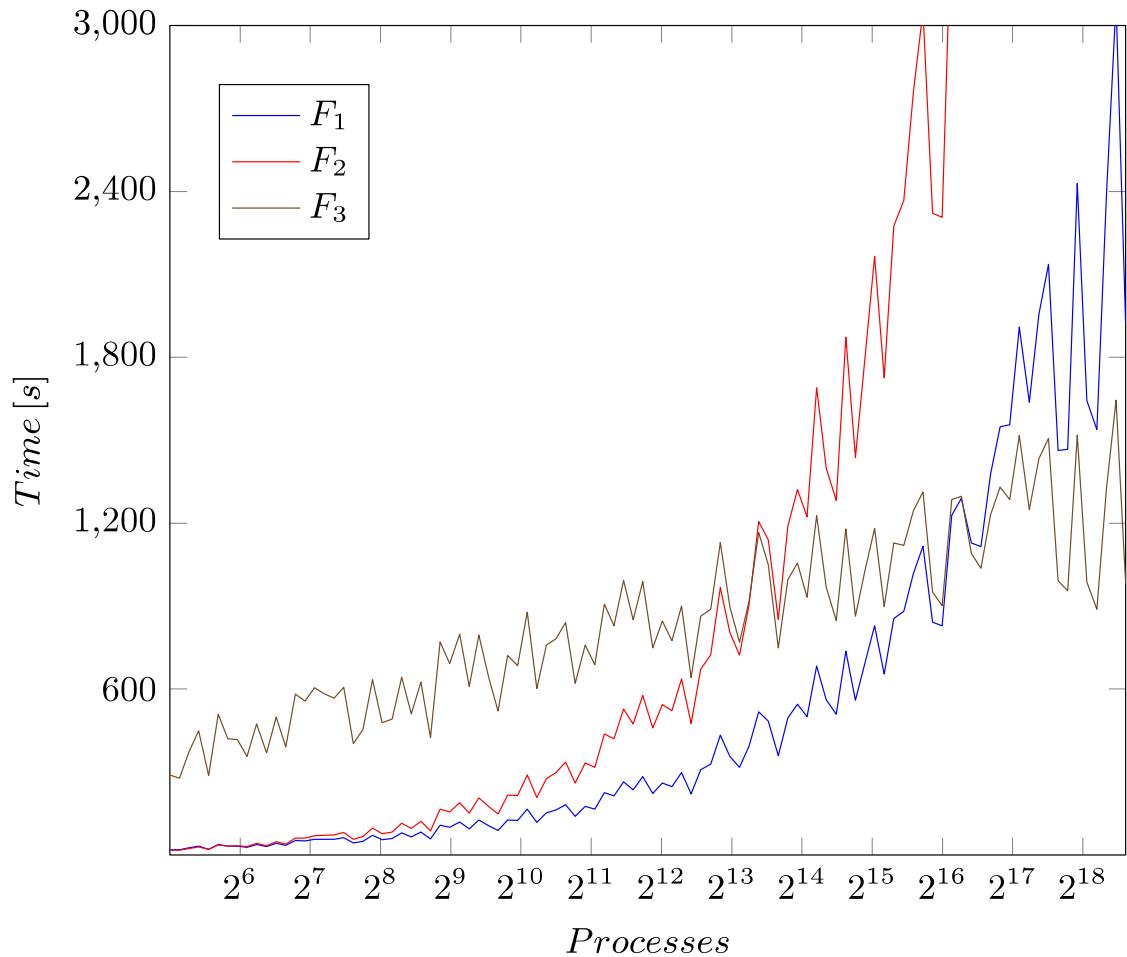


Ranking:

1.  $F_2$
2.  $F_1$
3.  $F_3$



# Primary focusing on scaling trend



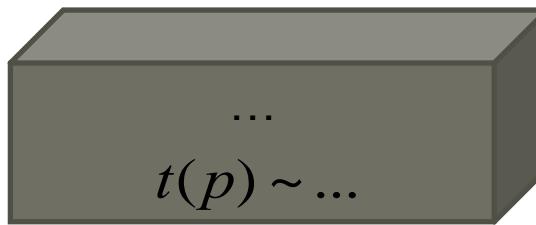
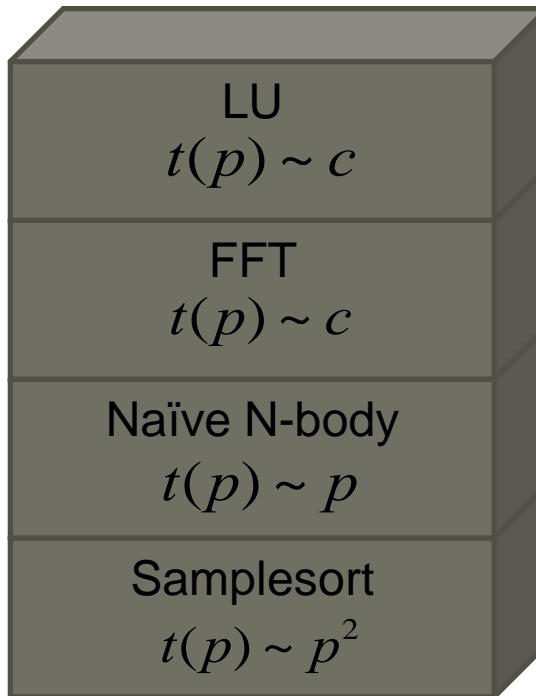
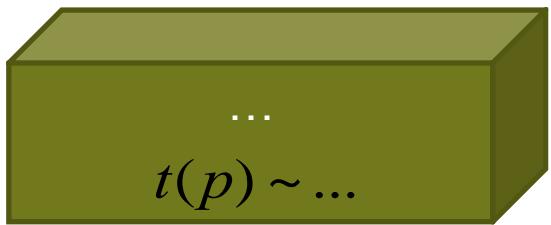
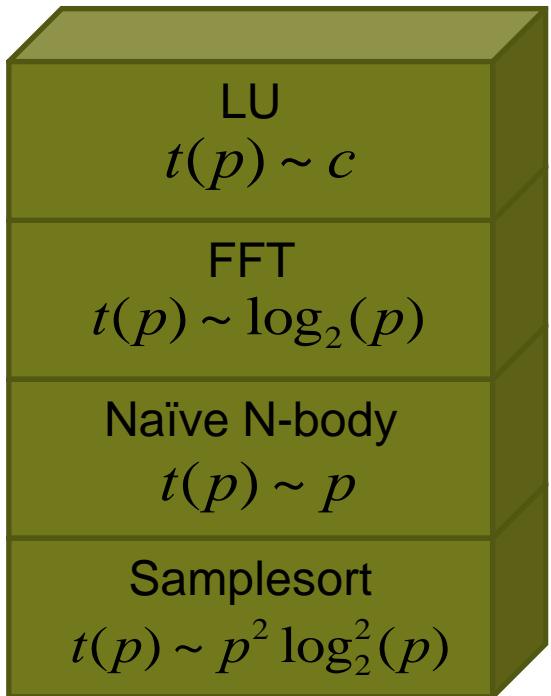
Ranking:

1.  $F_2$
2.  $F_1$
3.  $F_3$



# Model building blocks

Computation



Communication



# Performance model normal form

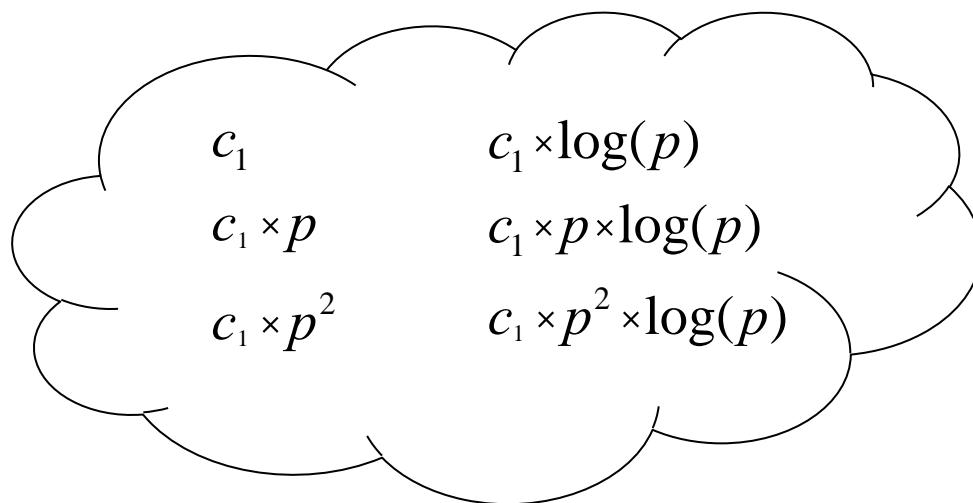
$$f(p) = \bigodot_{k=1}^n c_k \times p^{i_k} \times \log_2^{j_k}(p)$$

$n$	$\uparrow$	$\mathbb{N}$
$i_k$	$\uparrow$	$I$
$j_k$	$\uparrow$	$J$
$I, J$	$\uparrow$	$\mathbb{Q}$

$$n = 1$$

$$I = \{0, 1, 2\}$$

$$J = \{0, 1\}$$





# Performance model normal form

$$f(p) = \bigodot_{k=1}^n c_k \times p^{i_k} \times \log_2^{j_k}(p)$$

$n = 2$

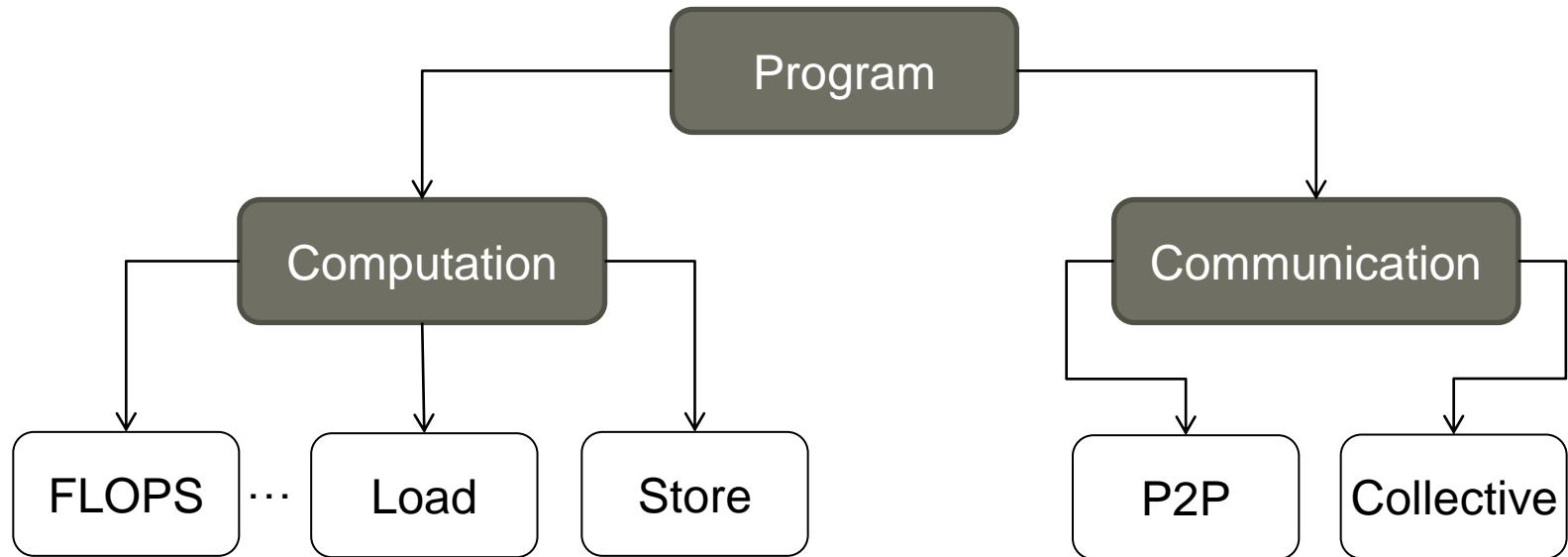
$I = \{0, 1, 2\}$

$J = \{0, 1\}$

$\boxed{\begin{matrix} n \uparrow & N \\ i_k \uparrow & I \\ \cdot & \hat{\cdot} \end{matrix}}$

$c_1 + c_2 \times p$	$c_1 \times \log(p) + c_2 \times p$
$c_1 + c_2 \times p^2$	$c_1 \times \log(p) + c_2 \times p^2$
$c_1 + c_2 \times \log(p)$	$c_1 \times \log(p) + c_2 \times p^2 \times \log(p)$
$c_1 + c_2 \times p \times \log(p)$	$c_1 \times p + c_2 \times p \times \log(p)$
$c_1 + c_2 \times p^2 \times \log(p)$	$c_1 \times p + c_2 \times p^2 \times \log(p)$
	$c_1 \times p \times \log(p) + c_2 \times p^2$
	$c_1 \times p \times \log(p) + c_2 \times p^2 \times \log(p)$
	$c_1 \times p^2 + c_2 \times p^2 \times \log(p)$

# Requirements modeling



---

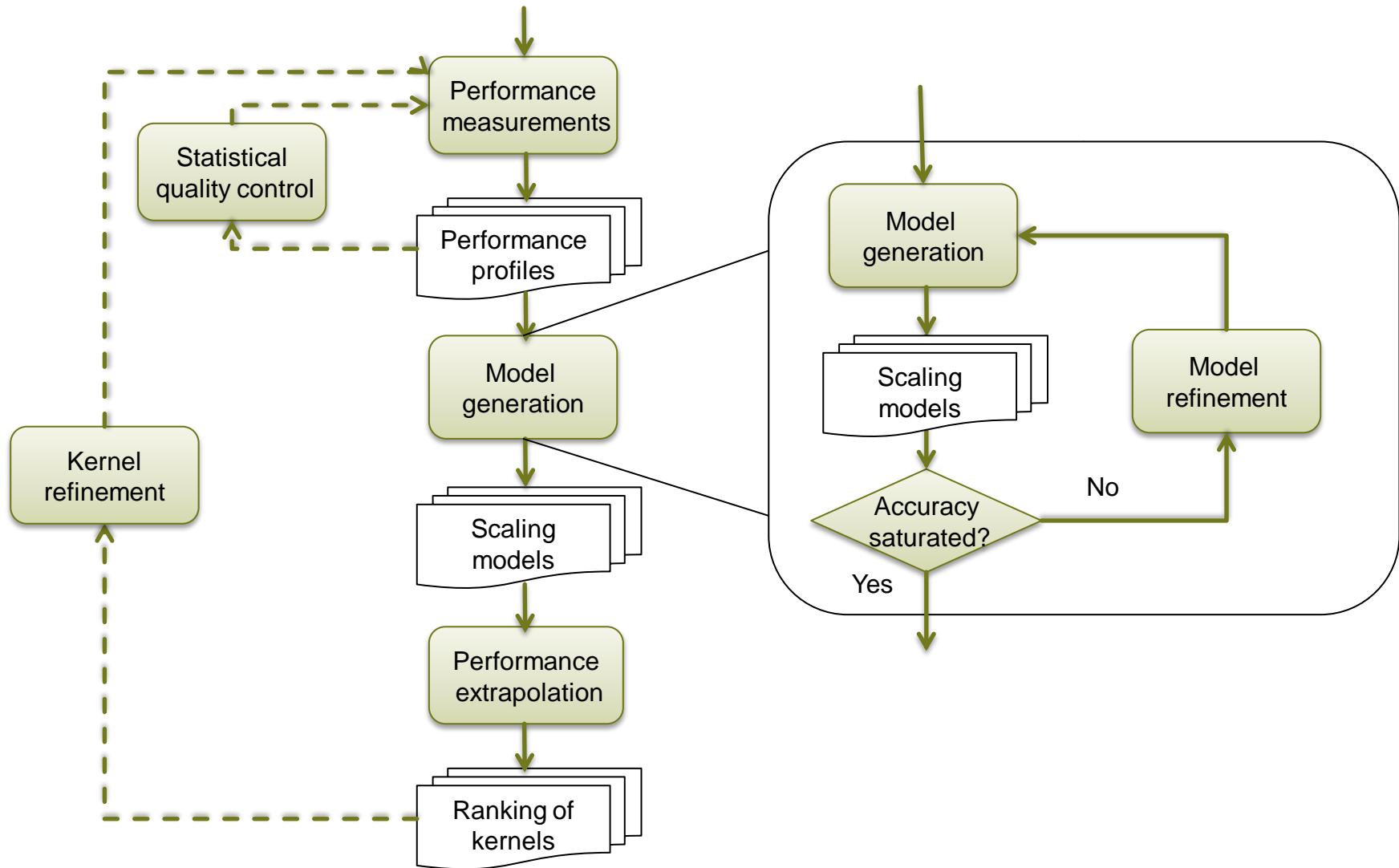
Disagreement may be indicative of wait states

---

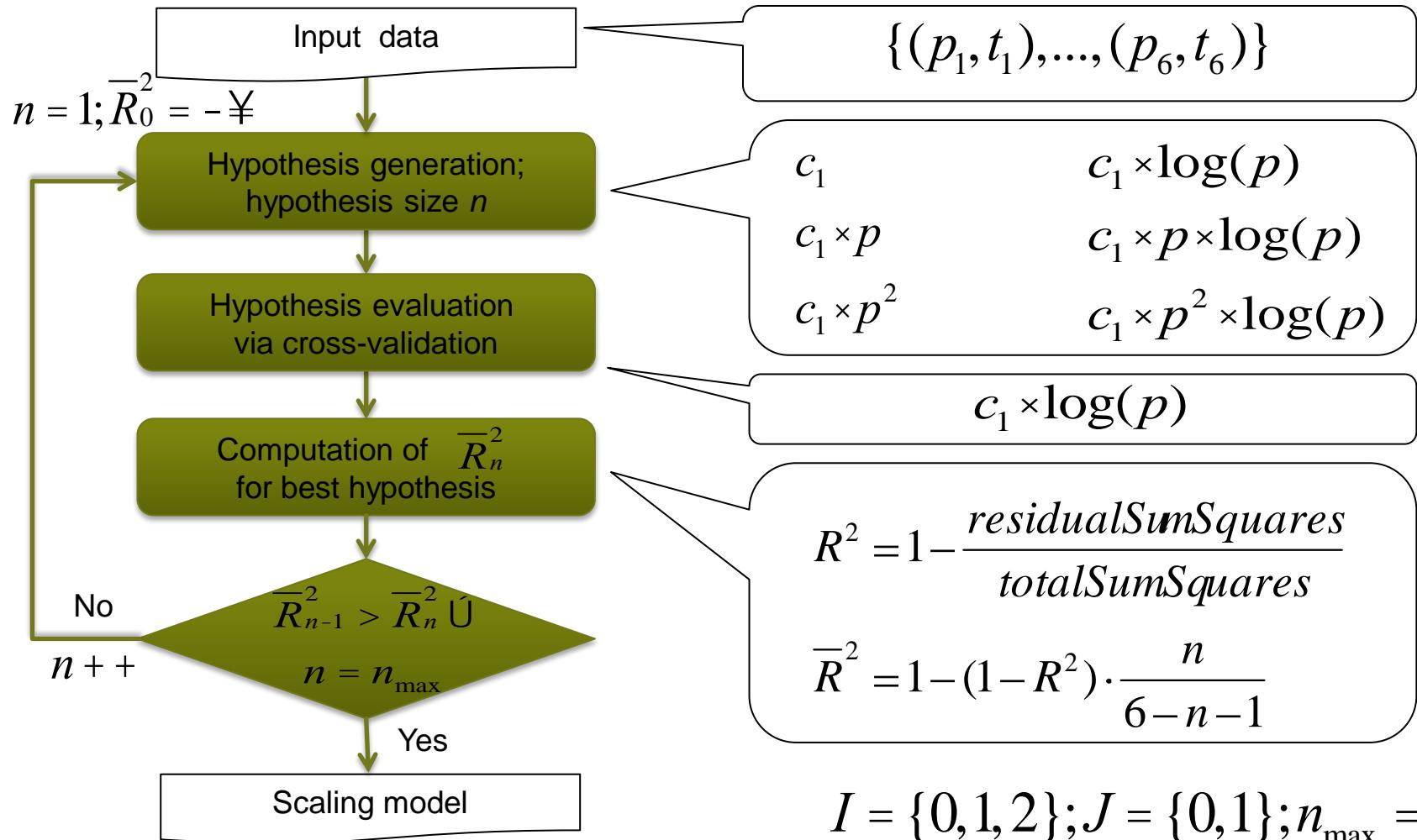


Time

# Workflow



# Model refinement





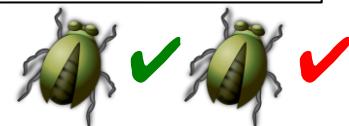
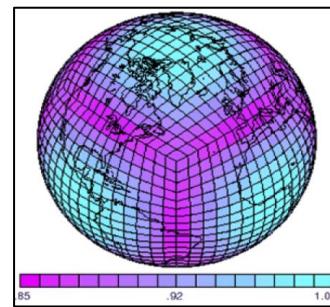
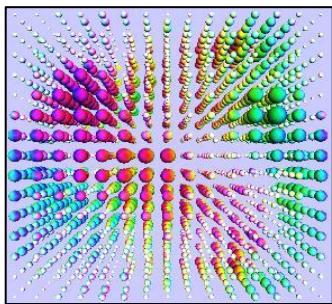
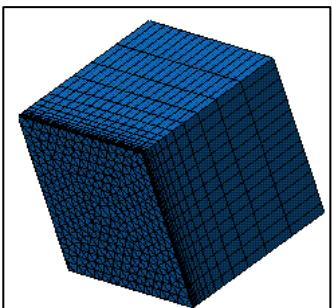
# Evaluation



$$I = \left\{ \frac{0}{2}, \frac{1}{2}, \frac{2}{2}, \frac{3}{2}, \frac{4}{2}, \frac{5}{2}, \frac{6}{2} \right\}$$

$$J = \{0, 1, 2\}$$

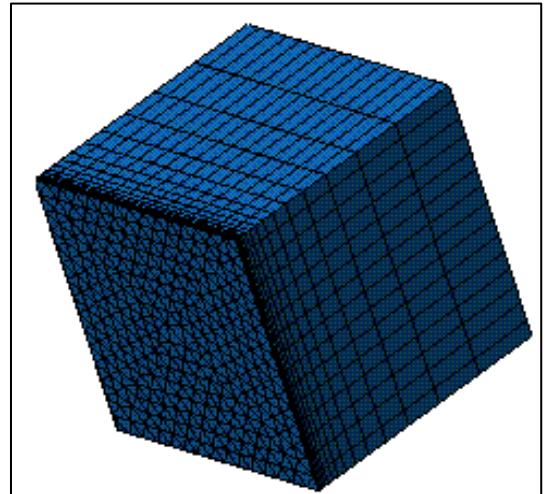
$$n = 5$$



# Sweep3D

Solves neutron transport problem

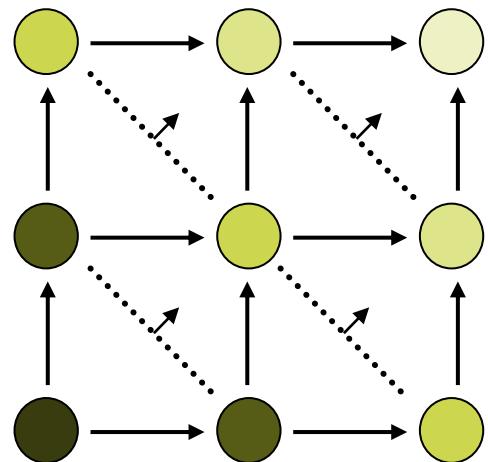
- 3D domain mapped onto 2D process grid
- Parallelism achieved through pipelined wave-front process



LogGP model for communication developed by [1].

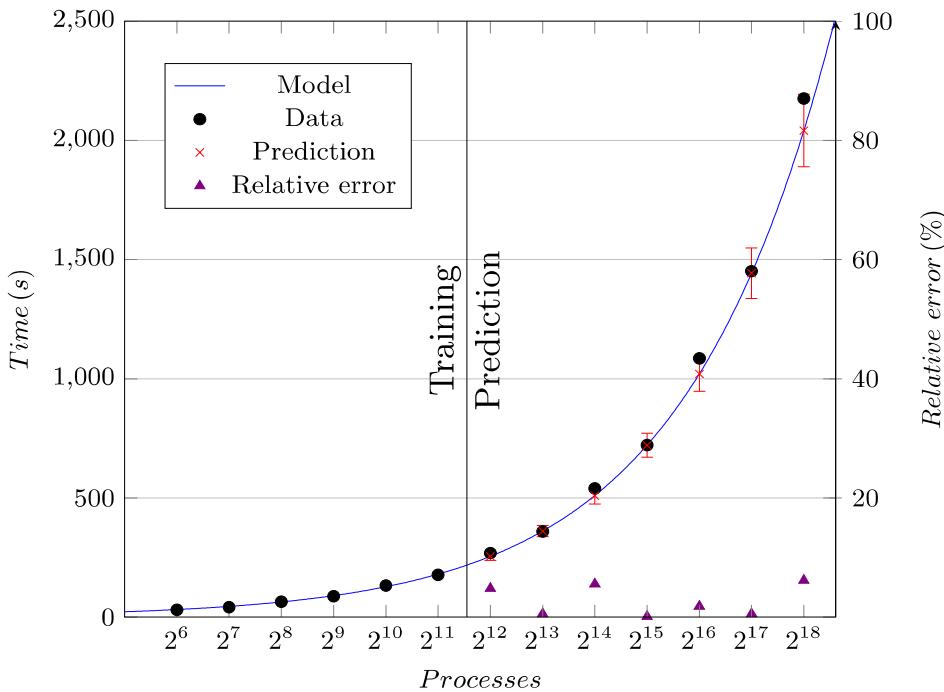
$$t^{comm} = [2(p_x + p_y - 2) + 4(n_{sweep} - 1)] \times t_{msg}$$

$$t^{comm} = c \times \sqrt{p}$$





## Sweep3D (2)

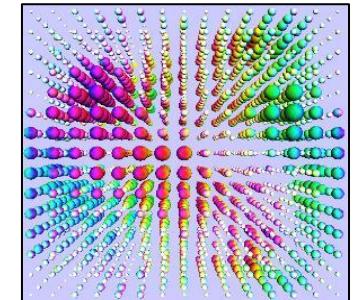


Kernel [2 of 40]	Runtime[%] $p_t=262k$	Model [s] $t = f(p)$	Predictive error [%] $p_t=262k$
sweep->MPI_Recv	65.35	$4.03\sqrt{p}$	5.10
sweep	20.87	582.19	0.01



# MILC

MILC/su3\_rmd – code from MILC suite of QCD codes with performance model manually created by [2].



- Time per process should remain constant except for a rather small logarithmic term caused by global convergence checks

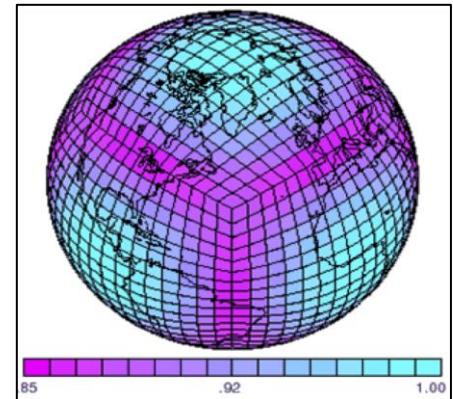
Kernel [3 of 479]	Model [s] $t=f(p)$	Predictive Error [%] $p_t=64k$
compute_gen_staple_field	$2.40 \times 10^{-2}$	0.43
g_vecdoublesum>MPI_Allreduce	$6.30 \times 10^{-6} \times \log_2^2(p)$	0.01
mult_adj_su3_fieldlink_lathwec	$3.80 \times 10^{-3}$	0.04

$$P_i \in 16k$$

# HOMME

## Core of the Community Atmospheric Model (CAM)

- Spectral element dynamical core  
on a cubed sphere grid



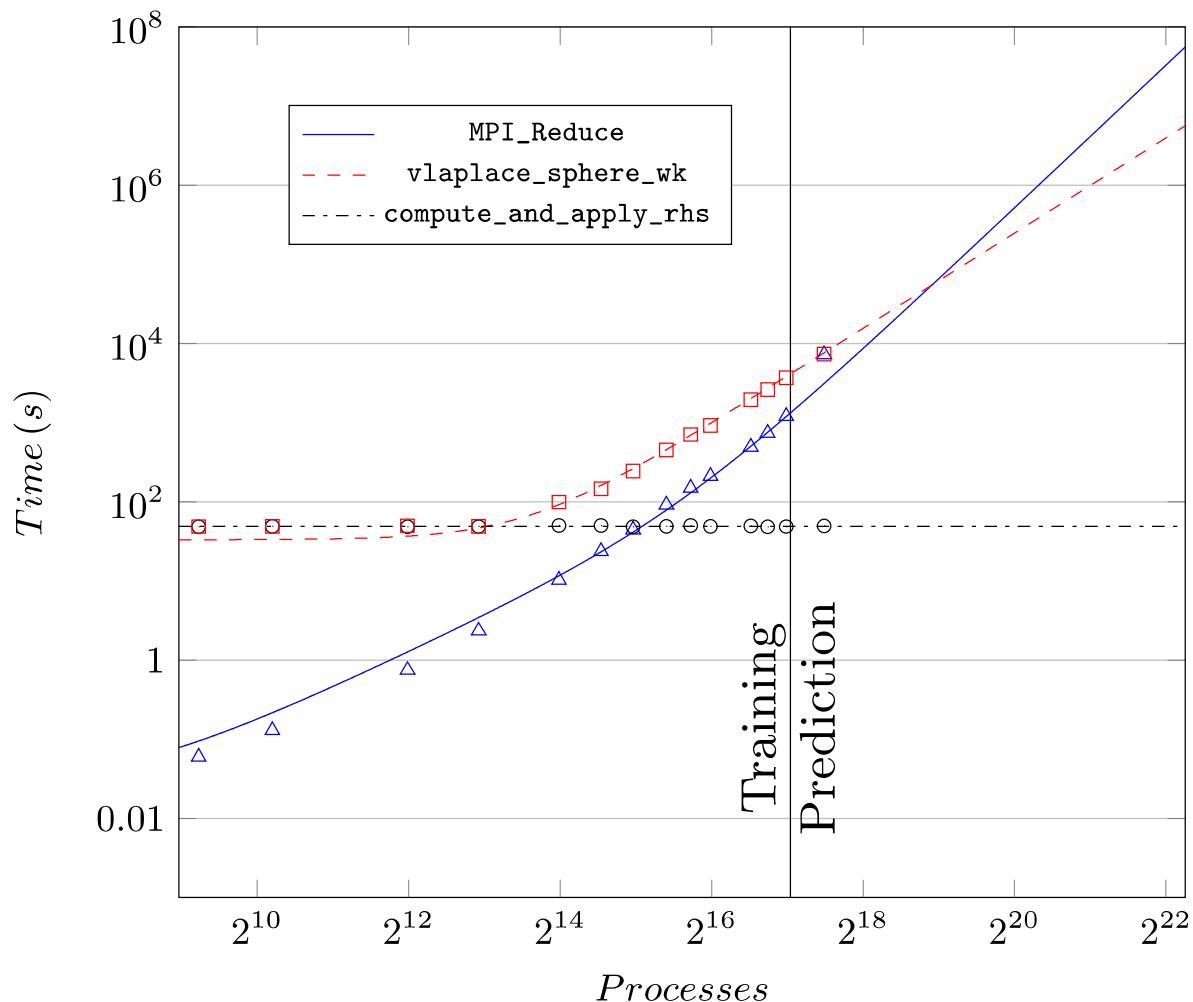
Kernel [3 of 194]	Model [s] $t = f(p)$	Predictive error [%] $p_t = 130k$
Box_rearrange->MPI_Reduce	$3.63 \times 10^{-6} p \times \sqrt{p} + 7.21 \times 10^{-13} p^3$	30.34
Vlaplace_sphere_vk	$24.44 + 2.26 \times 10^{-7} p^2$	4.28
Compute_and_apply_rhs		0.83

$$P_i \in 43k$$

The G8 Research Councils Initiative on Multilateral Research Funding  
Interdisciplinary Program on Application Software towards Exascale Computing for Global Scale Issues



# HOMME (2)



# Mass-producing performance models

- Is feasible
- Offers insight
- Requires low effort
- Improves code coverage

## Future work

- Integration into Scalasca
- Strong scaling
- Asymptotic requirements characterization



# Acknowledgements

- John Dennis and Rich Loft  
National Center For Atmospheric Research
- Marc-André Hermanns  
German Research School for Simulation Sciences





# Cost of first prediction

## Assumptions

- Input experiments at scales  $\{2^0, 2^1, 2^2, \dots, 2^m\}$
- Target scale at  $2^{m+k}$
- Application scales perfectly



k	Full scale [%]	Input [%]
1	100	<100
2	100	<50
3	100	<25
4	100	<12.5

Jitter may require more experiments per input scale, but to be conclusive experiments at the target scale would have to be repeated as well