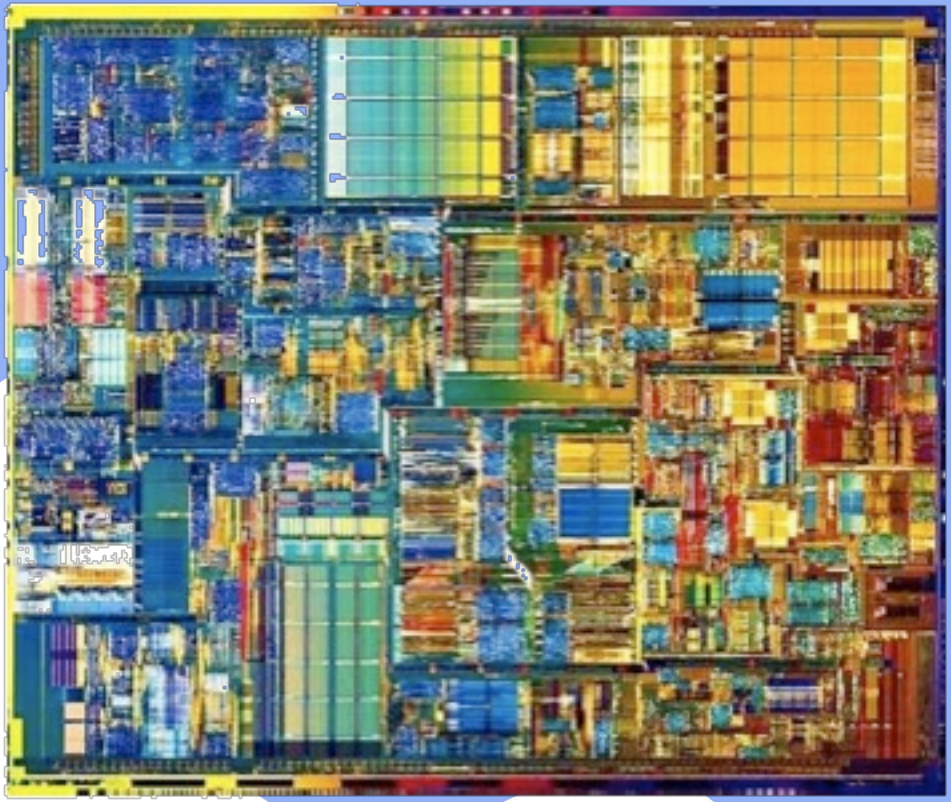
Modeling Instruction Placement on a Spatial Architecture

Martha Mercaldi
Steven Swanson, Andrew Petersen, Andrew Putnam, Andrew Schwerin
Mark Oskin and Susan Eggers

University of Washington

Why Spatial Architectures?

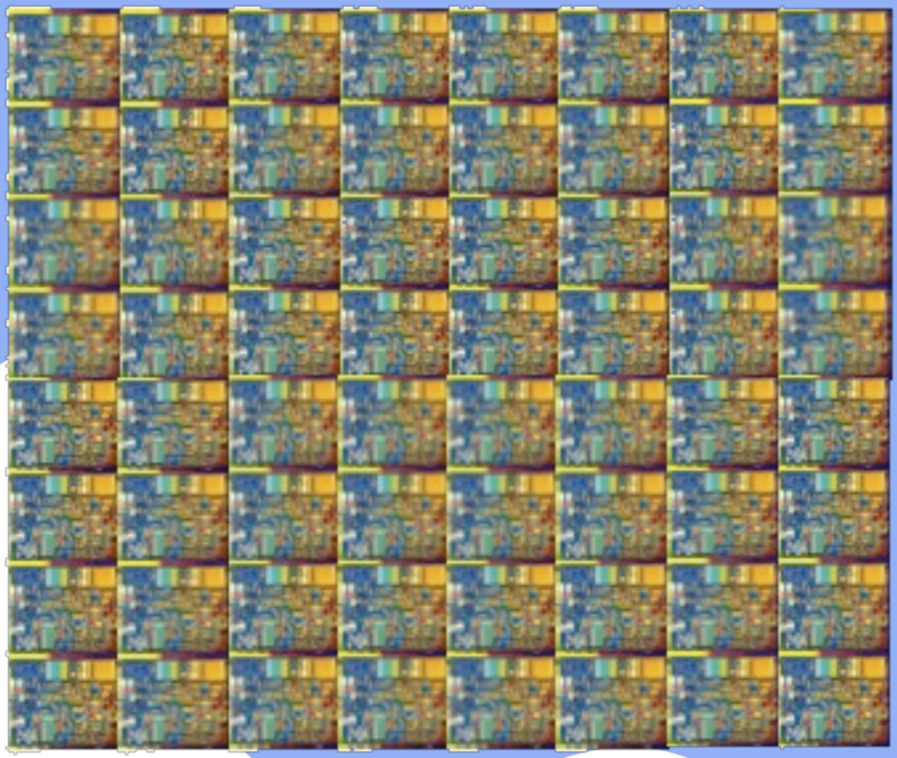


Scalability?

Complexity?

Power?

Why Spatial Architectures?

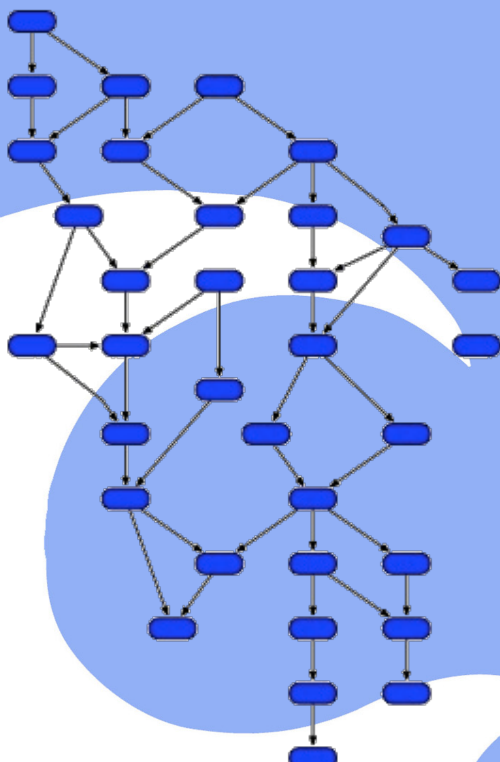


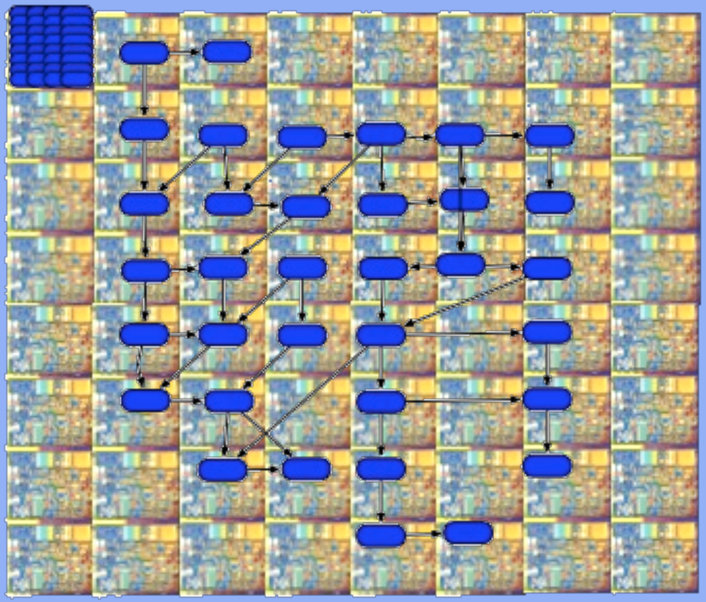
- Scalability
  - Short wires
- Complexity
  - Simple, replicated unit
- Power
  - Turn off unneeded tiles

What should execute where?

Instruction Placement

On a spatial architecture, where should execution occur?





Why model placement?

Enable exploration -

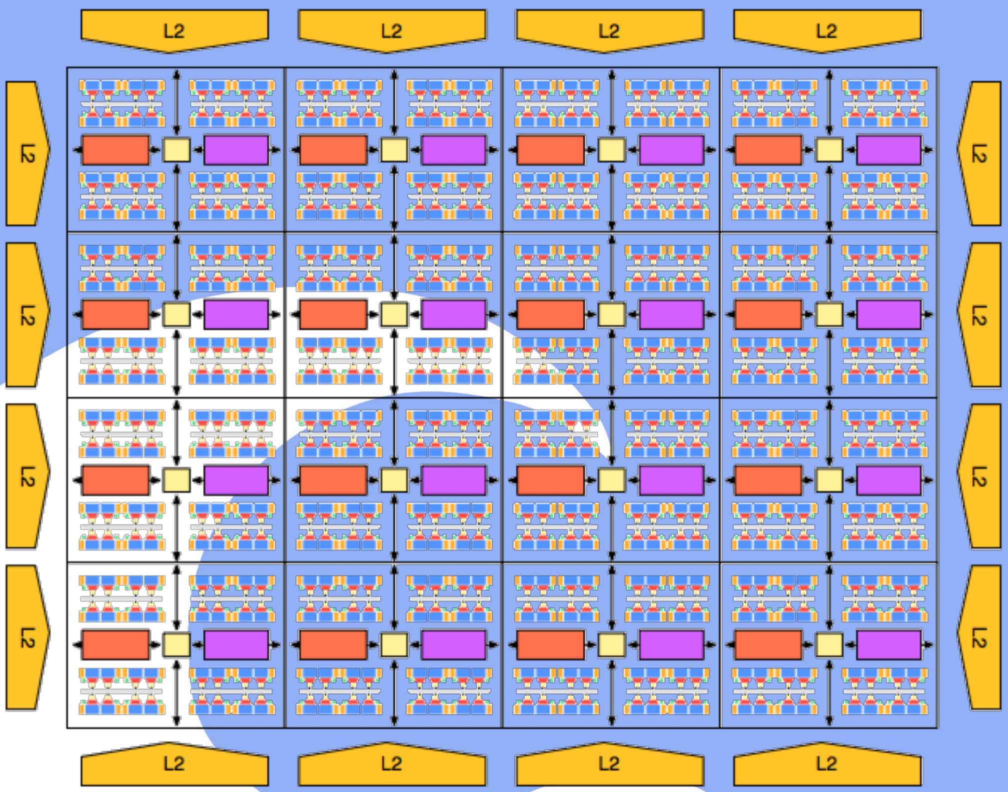
- *of placements*
- *of microarchitecture*

Guide for development of placement algorithms [ASPLOS 06]

Talk Outline

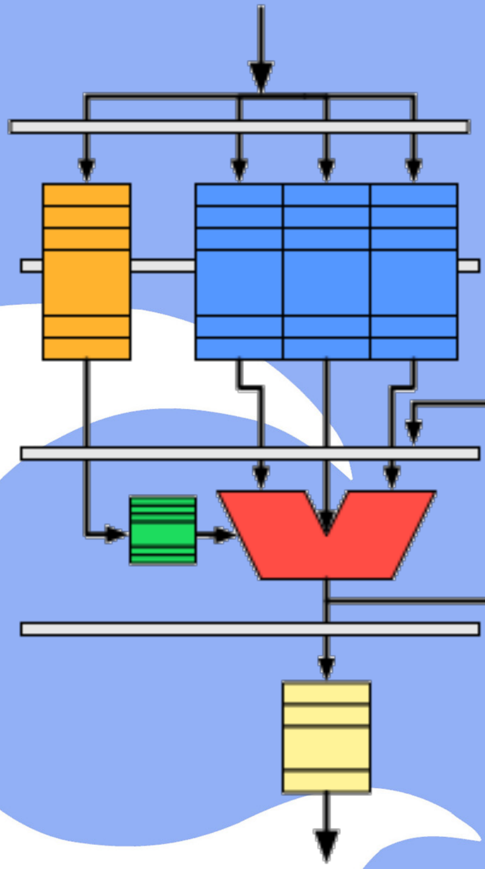
- Motivation
- WaveScalar Background
- Sub-model Construction & Evaluation
- Unified Model Construction & Evaluation

WaveScalar Processor



- Dataflow execution model
- Tiled microarchitecture

Processing Element



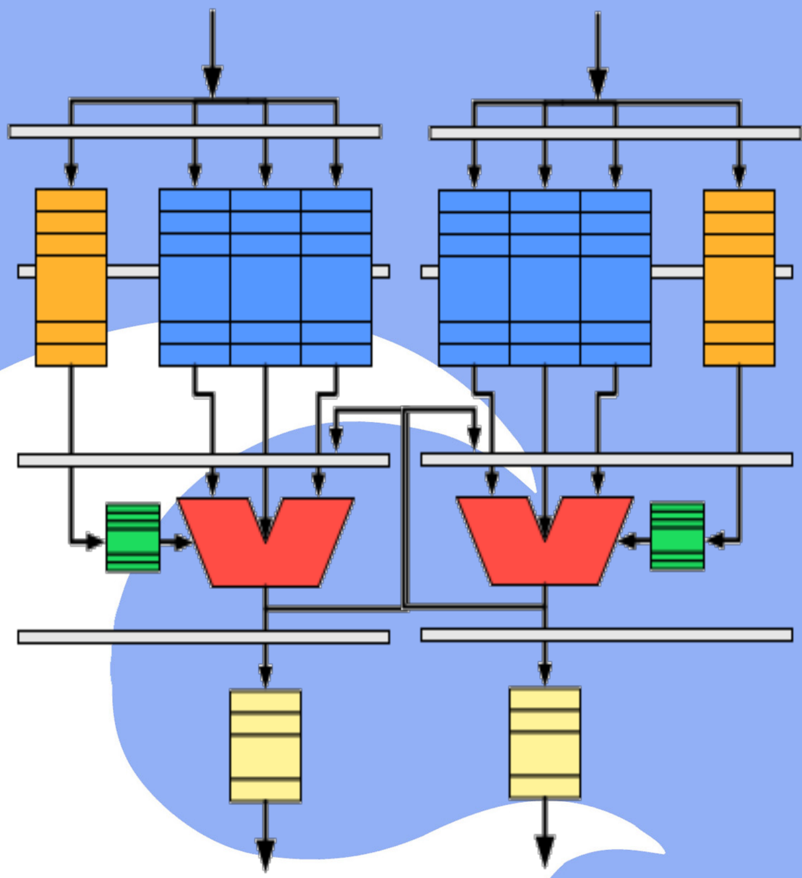
5-stage pipeline

Holds 64 instructions

1 execution unit

1 cycle operand latency

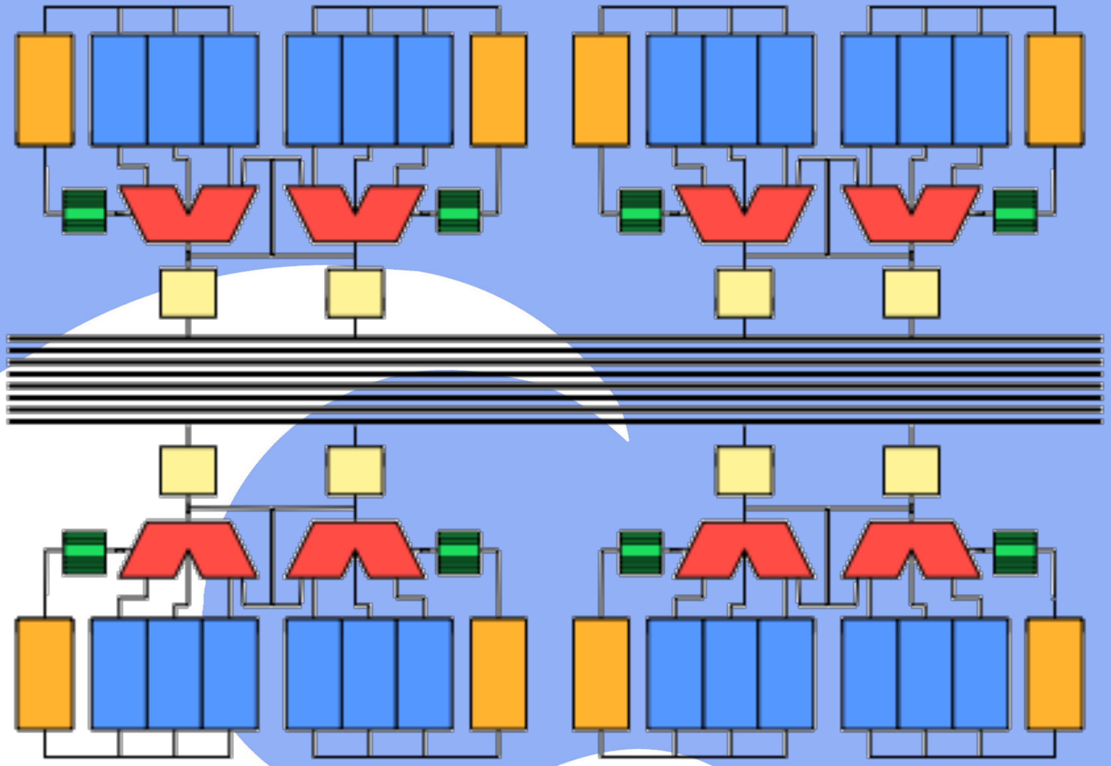
PEs in a Pod



2 Processing Elements

Execution stages linked

Domain

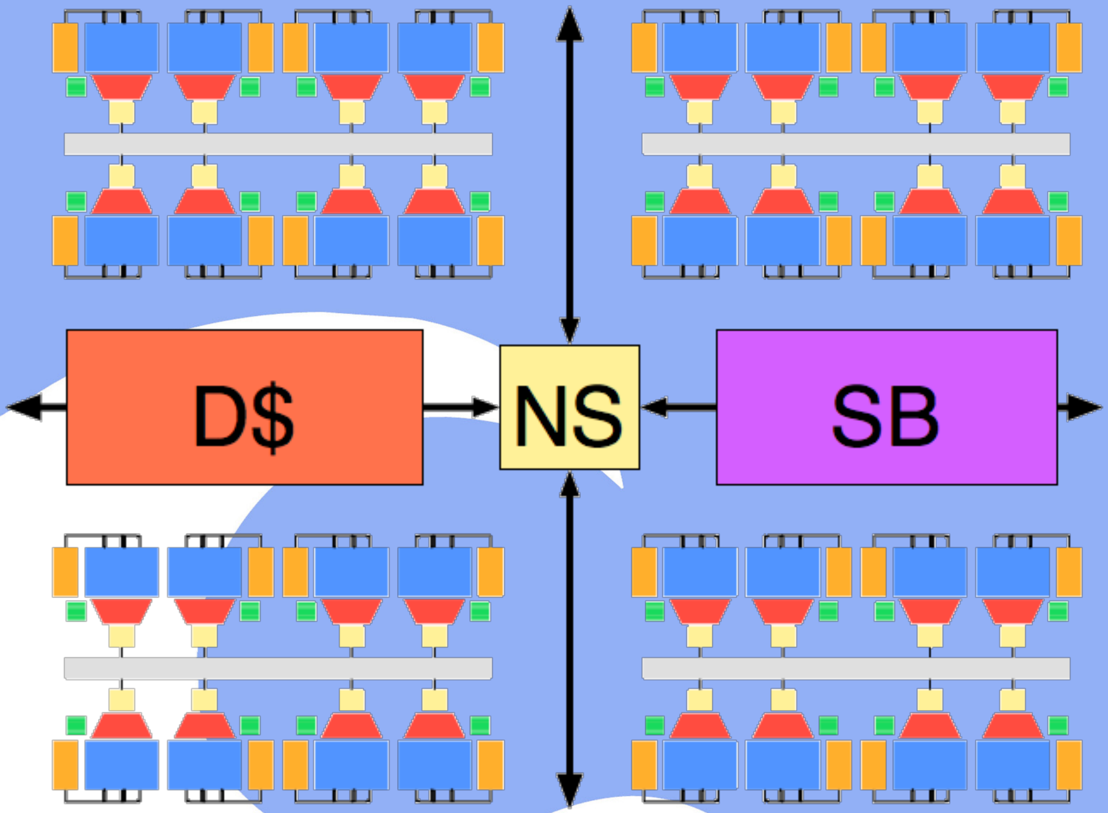


4 Pods

Crossbar interconnect

EXE to EXE: 4 cycles

Cluster



4 Domains

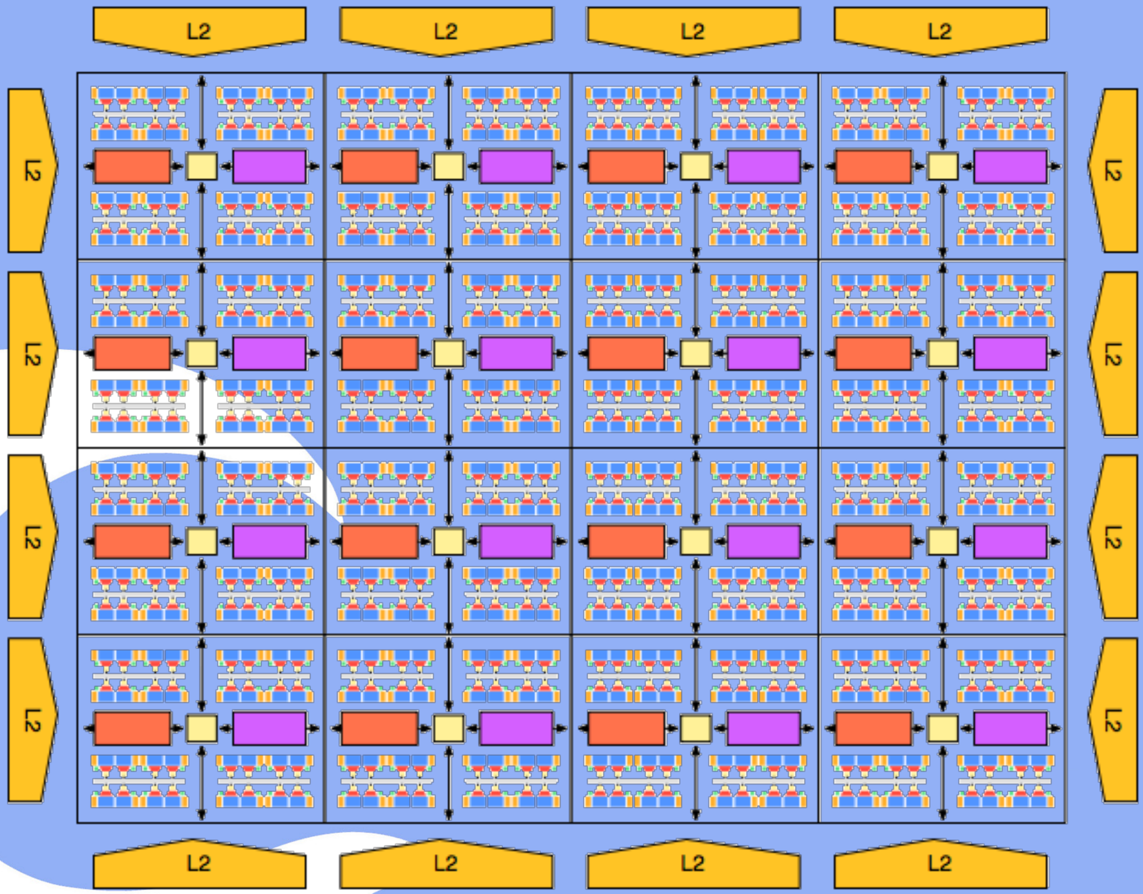
Network switch

Local L1 Data Cache

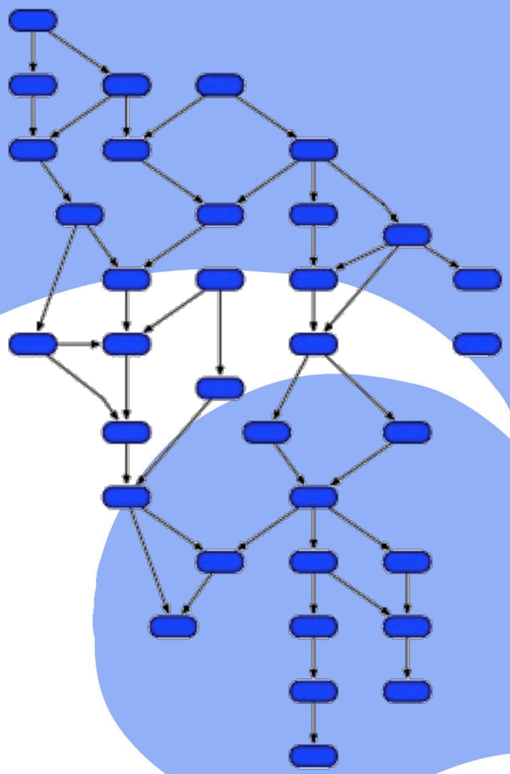
Store Buffer

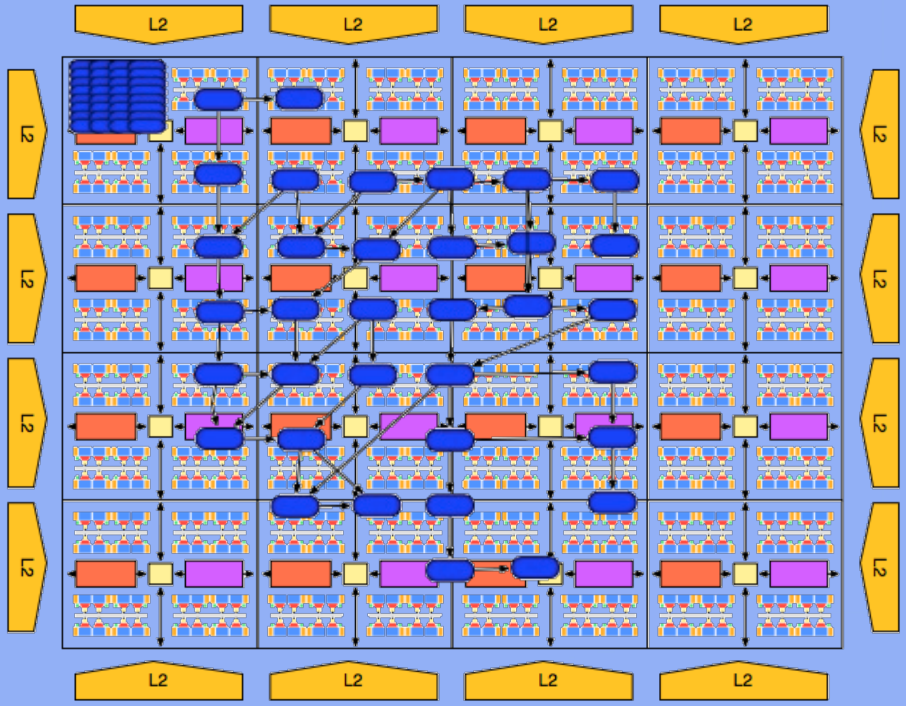
EXE to EXE: 7 cycles

WaveScalar Processor



Application Execution

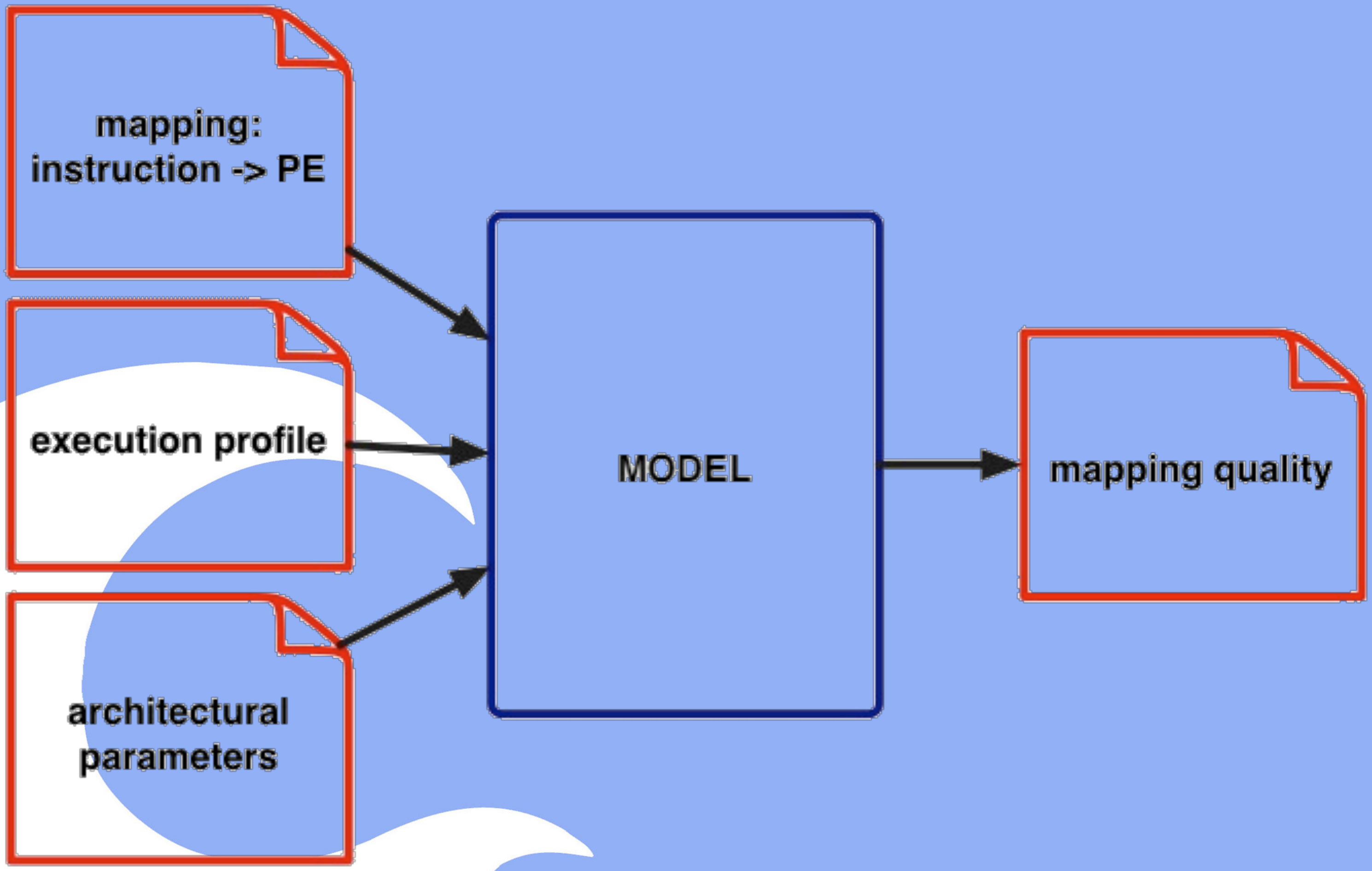




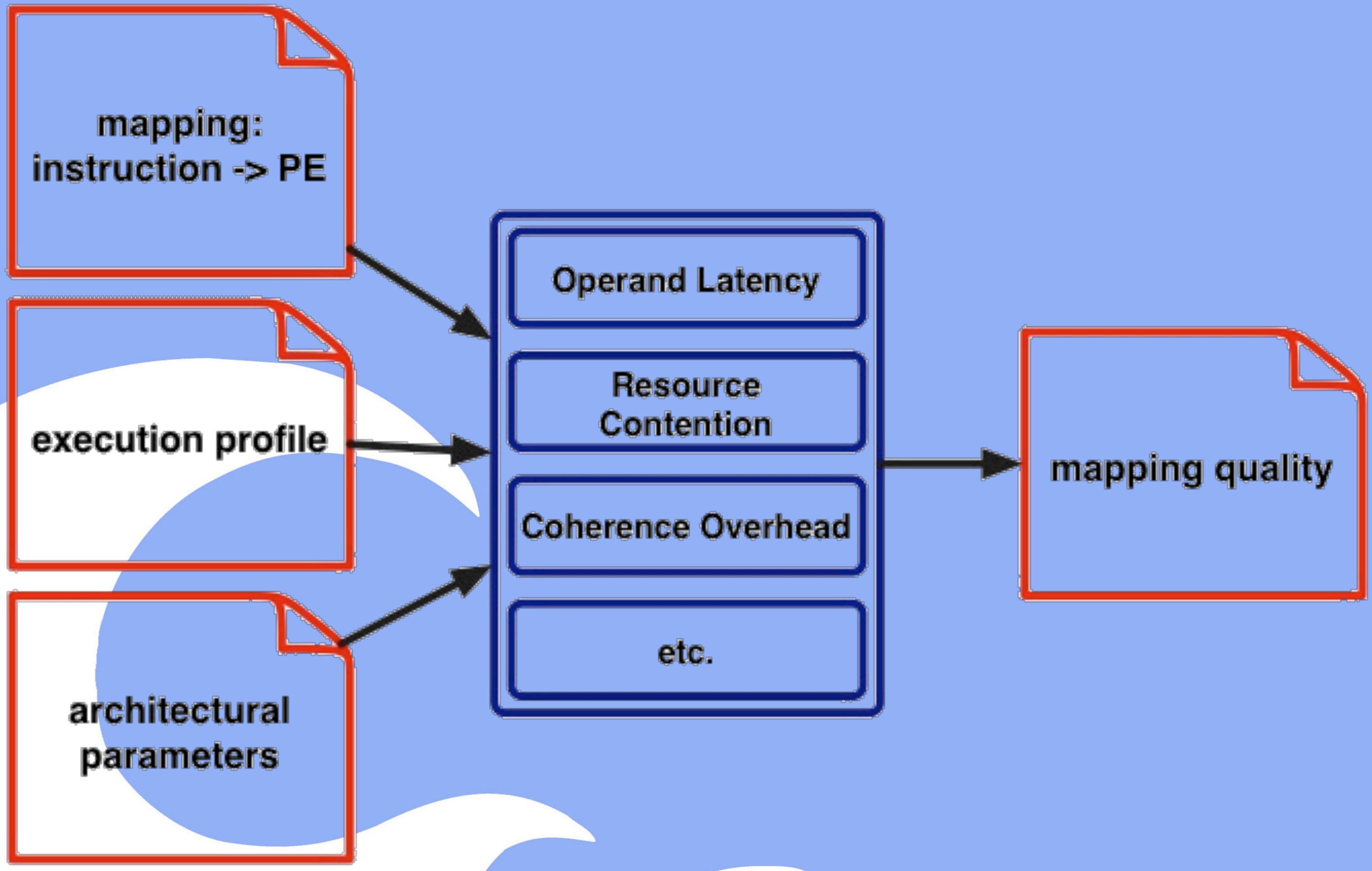
Talk Outline

- Motivation
- WaveScalar Background
- Sub-model Construction & Evaluation
  - Methodology
  - Example
- Unified Model Construction & Evaluation

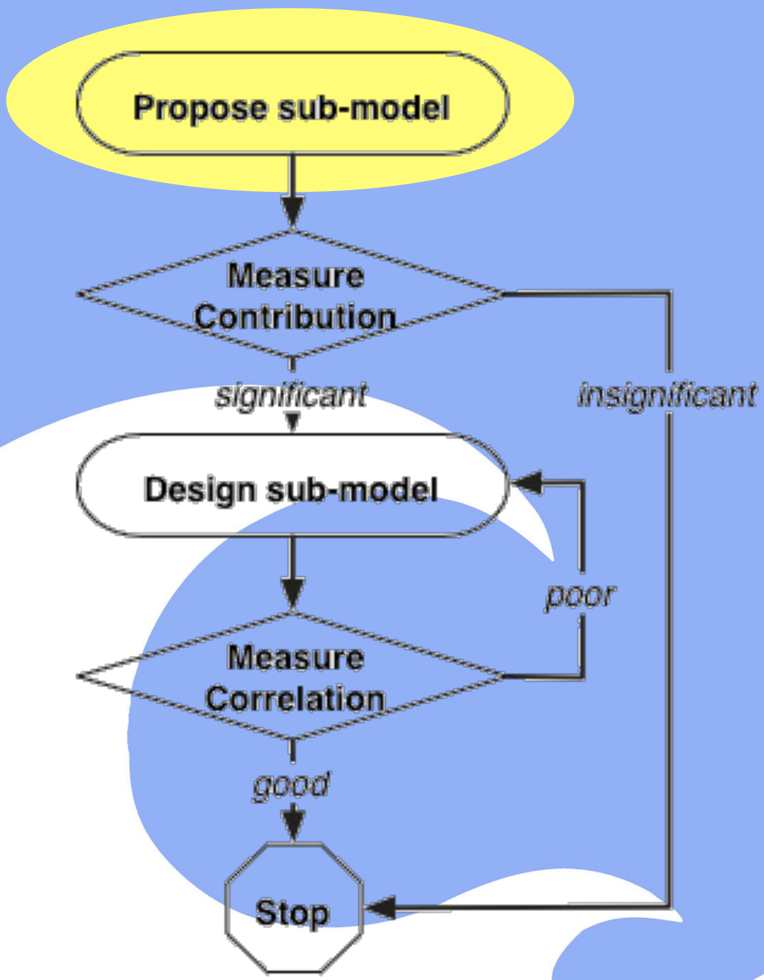
Model Inputs & Output



Internal Model Structure



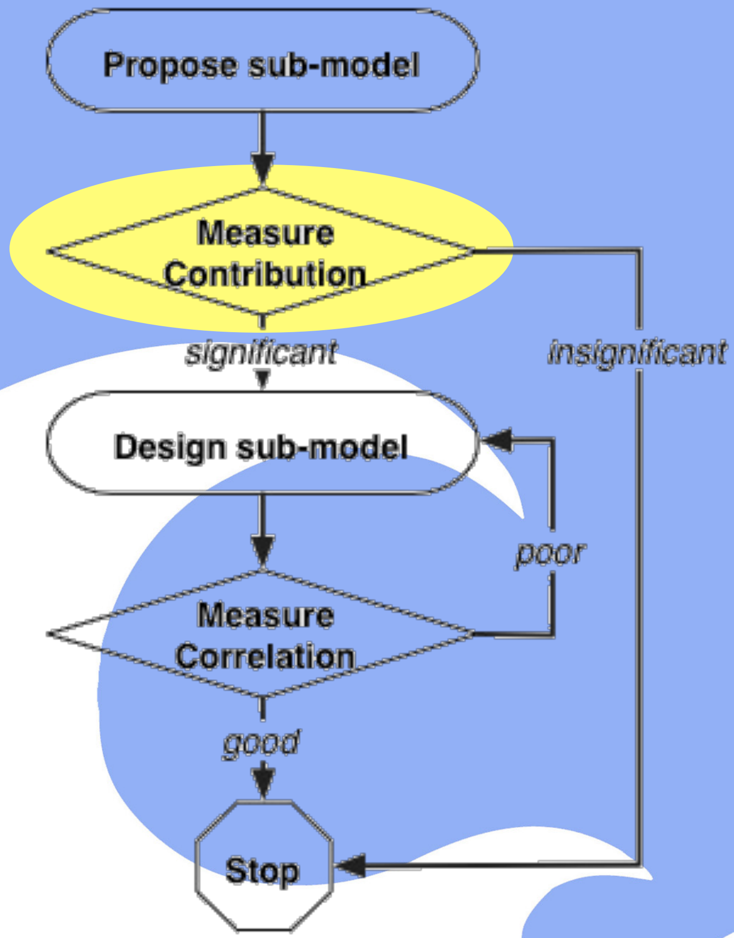
Sub-model Methodology



How might placement effect performance?

- –Operand Latency
- –Resource Contention
- –Network Bandwidth
- –Coherence overhead

Sub-model Methodology



How much does X effect performance?

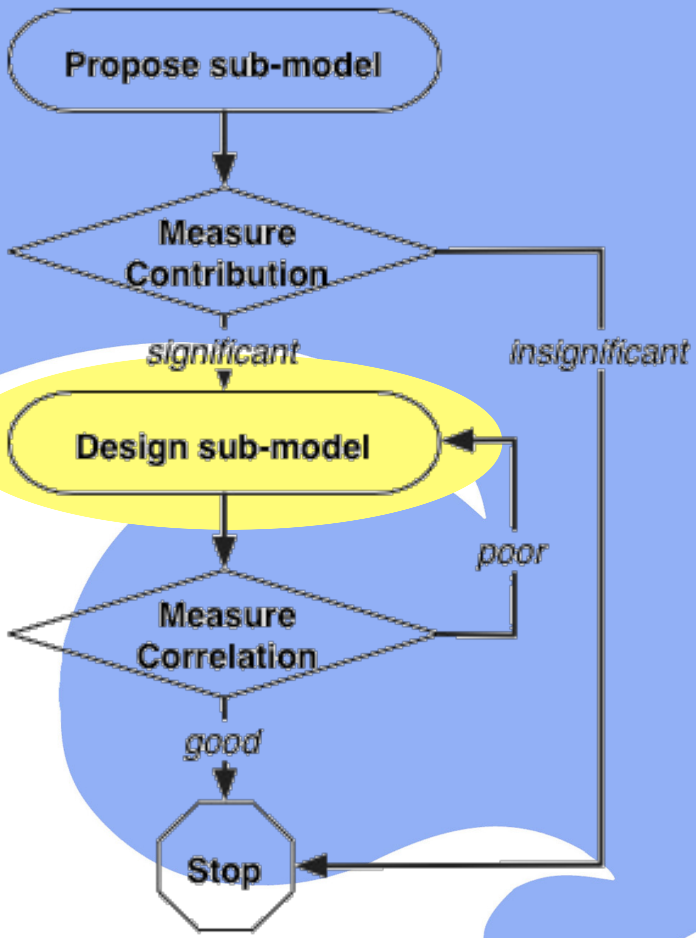
1. Generate a sampling of placements

2. Run idealized simulation

   (To measure contribution of X, idealize everything except X)

3. Contribution = Variance in IPC / Average IPC

Sub-model Methodology



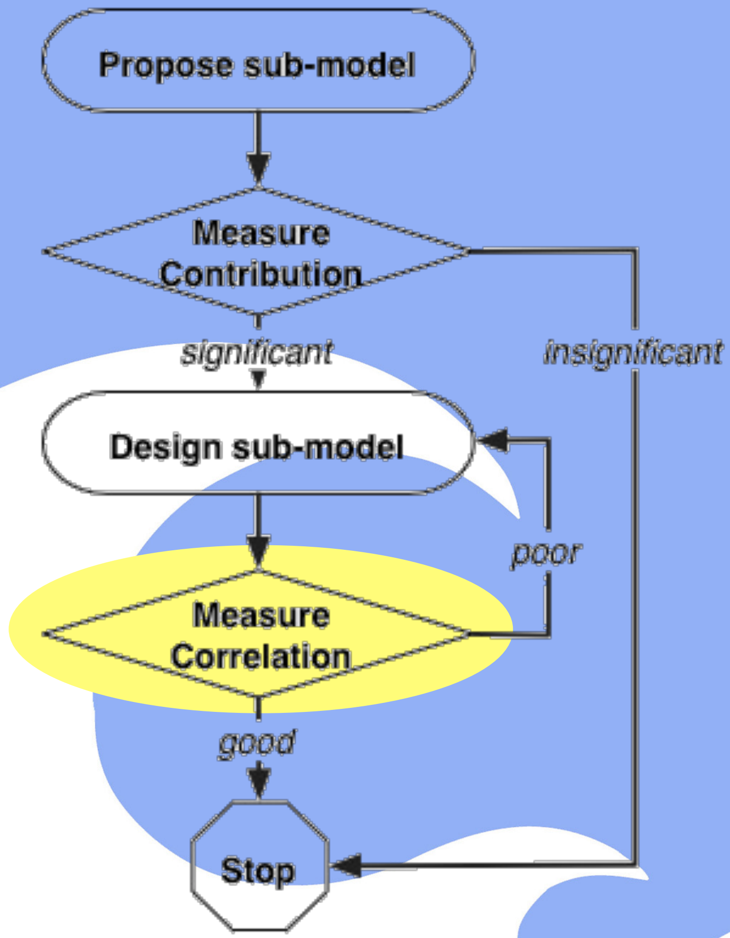
For a placement, what is the cost wrt. X?

Takes three inputs

- placement
- profile
- microarchitectural parameters

Produces cost for X

Sub-model Methodology



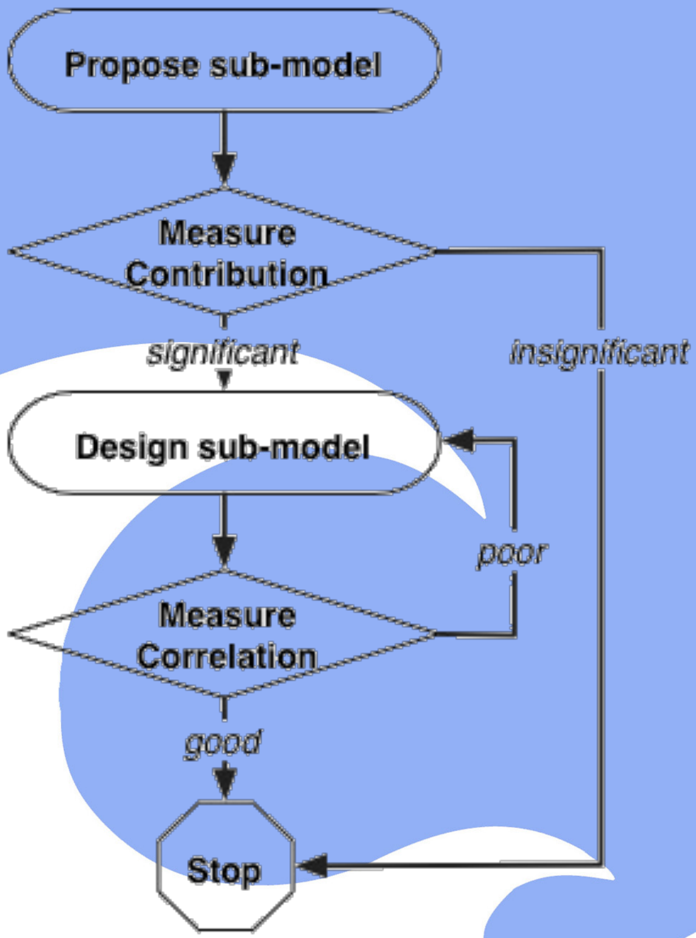
How good is the submodel?

Measure correlation between sub-model output to simulated IPC

(Still using idealized simulator)

Perfect correlation: -1.0

Sub-model Example: Operand Latency



Producer-consumer distance determines operand latency

In simulator, idealized:

- Interconnect bandwidth
- Execution resources
- Data & instruction caches

Contribution

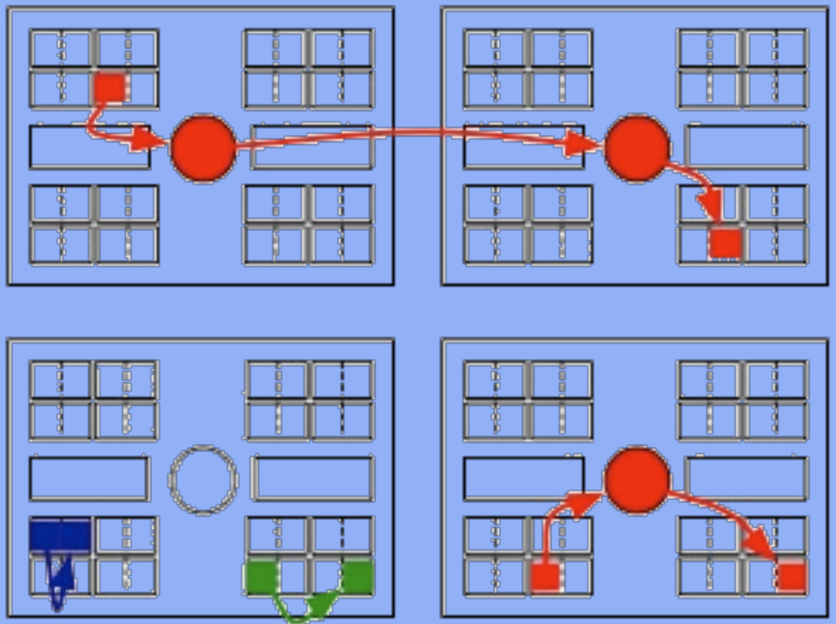
= Variance(IPC) / Average(IPC)

= 0.84

Sub-model Example: Operand Latency

Cost depends on type of communication

- Intra-pod
  - Latency = 0
- Intra-domain
  - $Latency_{i,j} = 4$
- Inter-domain
  - $Latency_{i,j} = 7 + ||C_i - C_j||$

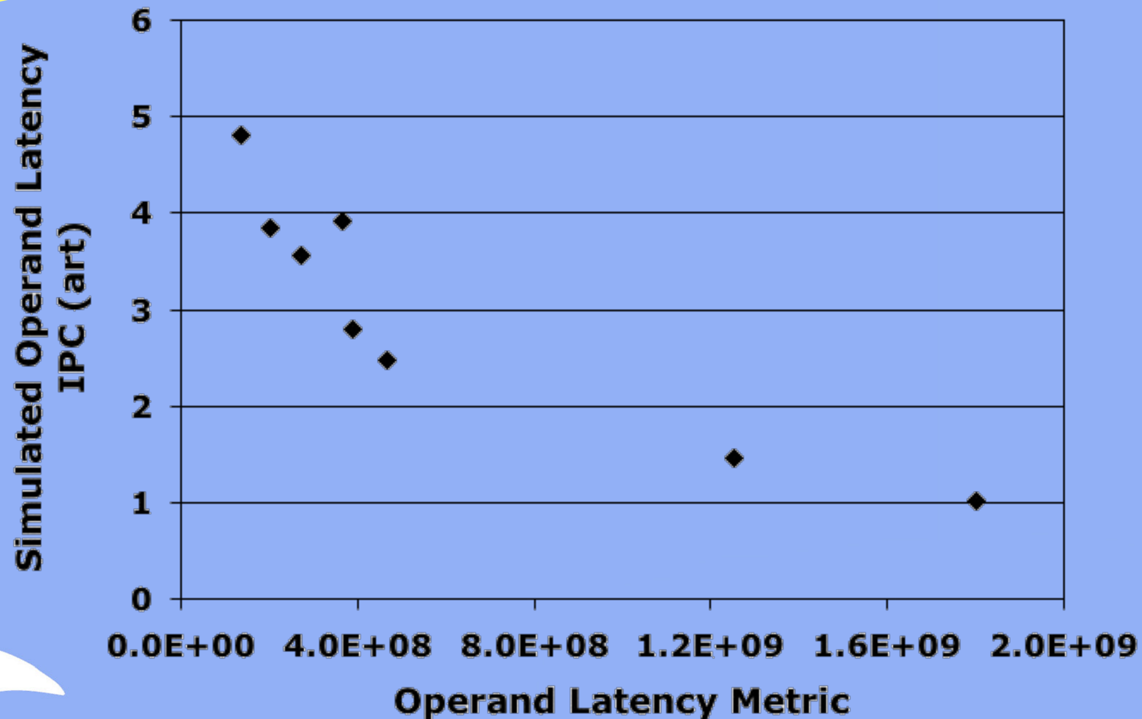


$T_{i,j}$ = dynamic number of operand tokens

$Latency = \sum_{i,j} (T_{i,j} * Latency_{i,j})$

Sub-model Example: Operand Latency

	Correlation
art	-0.90
equake	-0.92
fft	-0.88
gzip	-0.93
lu	-0.86
mcf	-0.80
twolf	-0.89
vpr	-0.84
Average	-0.88



Sub-model Summary

	Contribution (sub-model importance)	Correlation (sub-model quality)
Operand latency	0.84	-0.88
Interconnect bandwidth	0.01	--
PE contention	1.21	-0.76
Cache coherence overhead	0.34	-0.84

Talk Outline

Motivation

WaveScalar Background

Sub-model Construction & Evaluation

Unified Model Construction & Evaluation

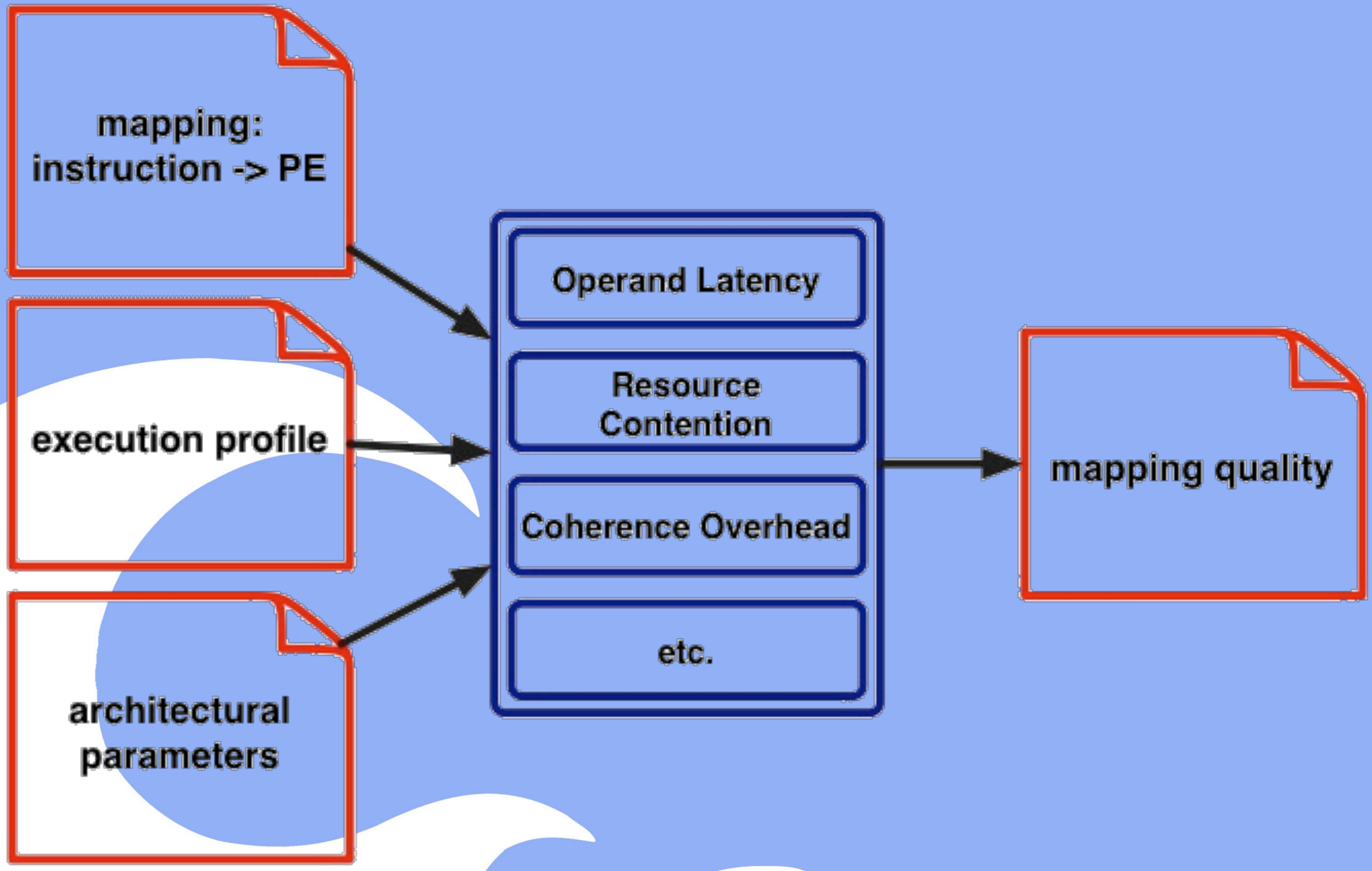
Sub-model Unification

	Contribution (sub-model importance)	**Correlation (sub-model quality)**
Operand Latency	~~0.84~~ 0.35	-0.88
Interconnect Bandwidth	~~0.01~~ 0.00	--
PE Contention	~~1.21~~ 0.51	-0.76
Cache coherence overhead	~~0.34~~ 0.14	-0.84

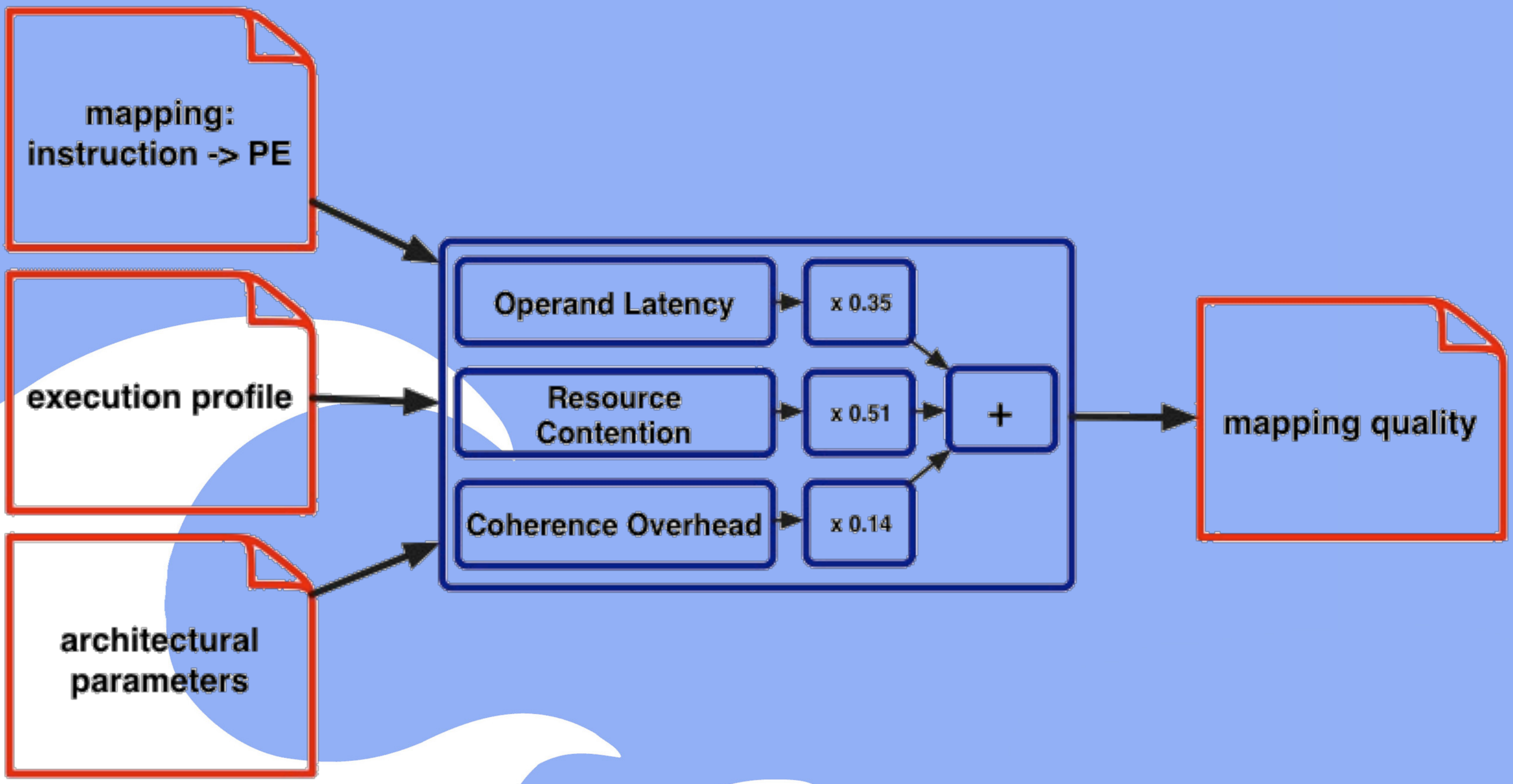
TotalScore =

0.35 x OperandLatencyScore +
0.51 x PeContentionScore +
0.14 x CoherenceOverheadScore

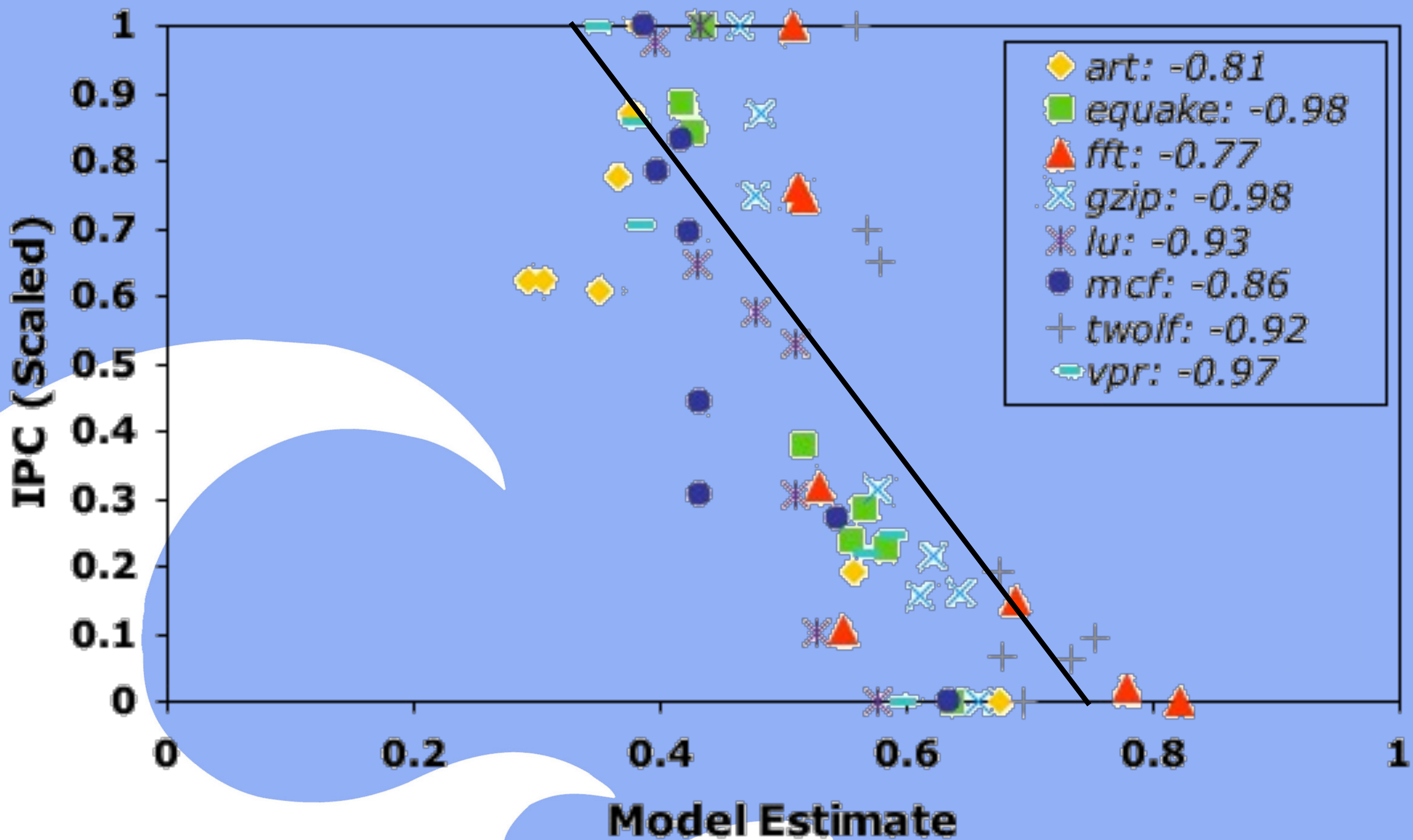
Internal Model Structure



Internal Model Structure



Unified Model: Evaluation



Unified Model: Evaluation

How does model predict performance of **new** application?

- Use cross-validation
- Split data into training and test sets
  - Example:
    - Training: all benchmarks except fft
    - Test: fft
- Derive model from ***training*** data
- Measure correlation on ***test*** data

Combined Model: Evaluation

Training Set	Test Set	Correlation Coeff. (on test set)
all except art	art	-0.76
all except equake	equake	-0.89
all except fft	fft	-0.74
all except gzip	gzip	-0.83
all except lu	lu	-0.77
all except mcf	mcf	-0.95
all except twolf	twolf	-0.76
all except vpr	vpr	-0.89
Average		-0.82

Conclusion

Application placement demands analytical model

Model that predicts application placement performance based on multiple factors

Predictions shows -0.82 correlation with simulated performance

For more information:

http://wavescalar.cs.washington.edu

Supporting Material

Sub-model Example: PE Contention

1. Proposed sub-model

Oversubscription of PE instruction cache hurts performance.

2. Measure Contribution

In simulator, idealized:

- Interconnect bandwidth
- Interconnect latency
- Data & instruction caches

Contribution

= Variance(IPC) / Average(IPC)

= 1.21

3. Construct sub-model

PeCapacity = 64

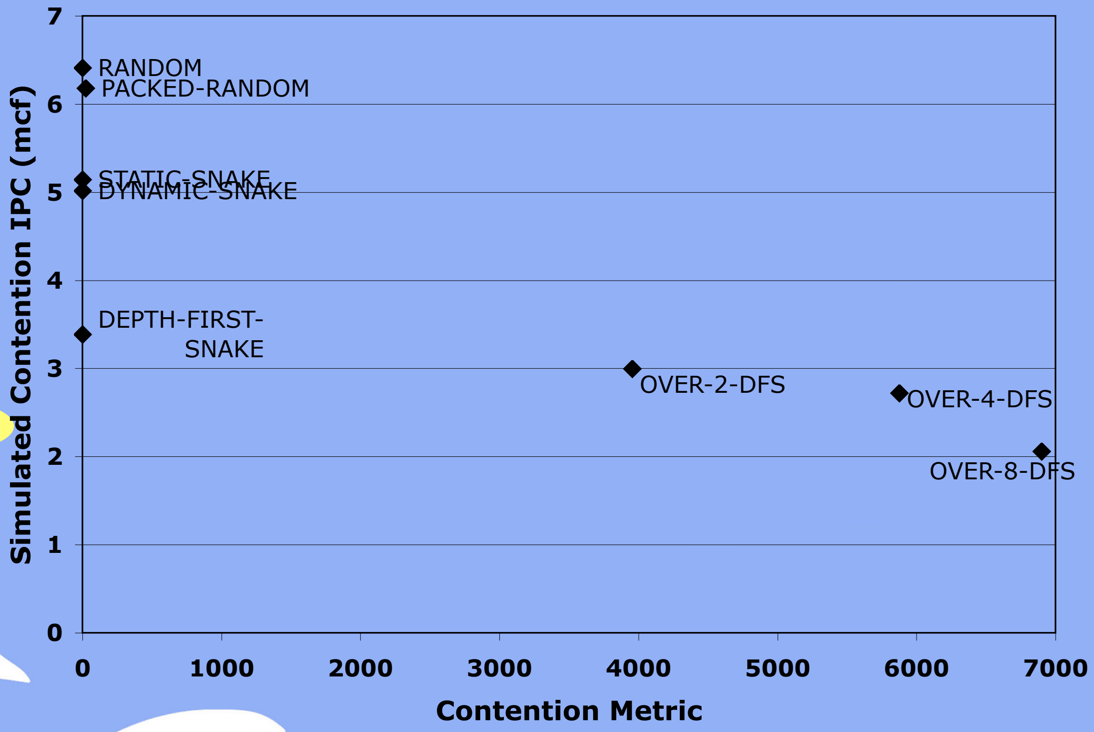
I_p *= number of instructions mapped to PE p*

$\text{Contention}_p = \max(0, I_p - \text{PeCapacity})$

$\text{PeContention} = \sum_p(\text{Contention}_p)$

Sub-model Example: PE Contention

	Correlation
art	-0.69
equake	-0.84
fft	-0.74
gzip	-0.83
lu	-0.65
mcf	-0.83
twolf	-0.79
vpr	-0.67
Average	-0.76



Sub-model Example: Cache Coherence Overhead

1. Proposed sub-model

Instruction placement determines location of cache line requests for distributed L1 data cache.

2. Measure Contribution

In simulator, idealized:

- Interconnect bandwidth
- Interconnect latency
- PE resourced

Contribution

= Variance(IPC) / Average(IPC)

= 0.34

3. Construct sub-model

C_a = *number of clusters accessing line a*

N_a = *total number of accesses to line a*

$$\text{misses}_a = \begin{cases} 1 & \text{if } C_a == 1 \\ C_a & \text{if } C_a > 1 \end{cases}$$

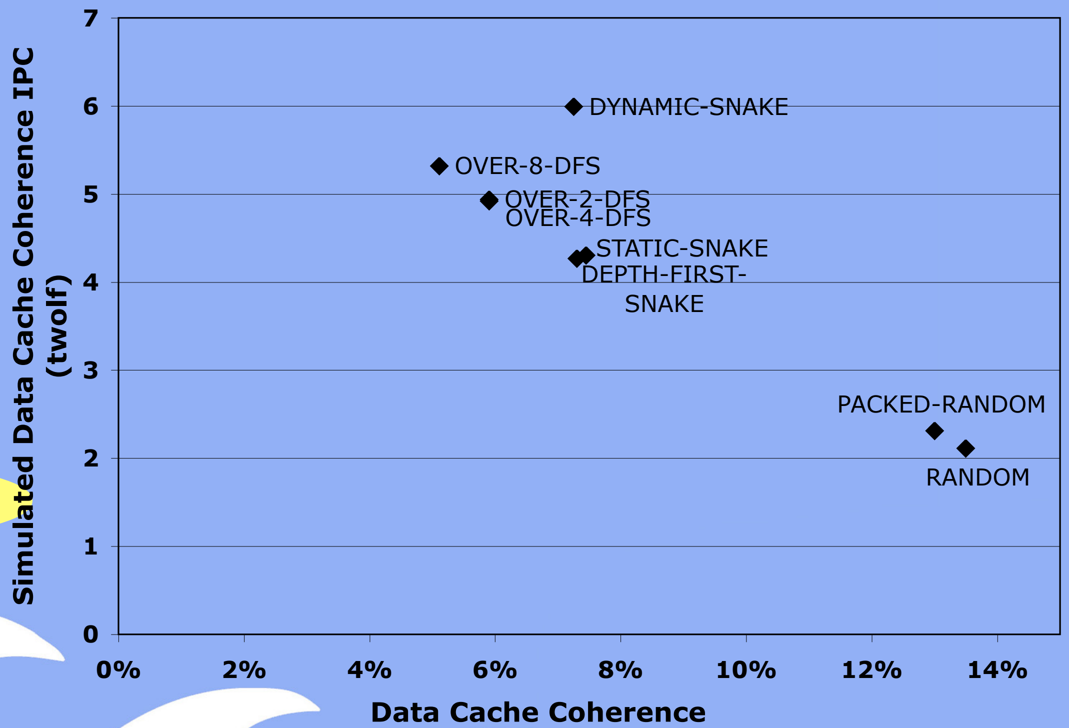
$$\text{hits}_a = \begin{cases} N_a - 1 & \text{if } C_a == 1 \\ N_a - C_a & \text{if } C_a > 1 \end{cases}$$

CoherenceOverhead =

Average miss rate for all a

Sub-model Example: Cache Coherence Overhead

	Correlation
art	-0.92
equake	-0.99
fft	-0.33
gzip	-0.95
lu	-0.64
mcf	-0.97
twolf	-0.92
vpr	-1.0
Average	-0.84



Dataflow Execution Model

- Not a new idea [Dennis 1975]
- Code is a graph
  - Vertices = instructions
  - Edges = operands
- Execution governed by "dataflow firing rule"