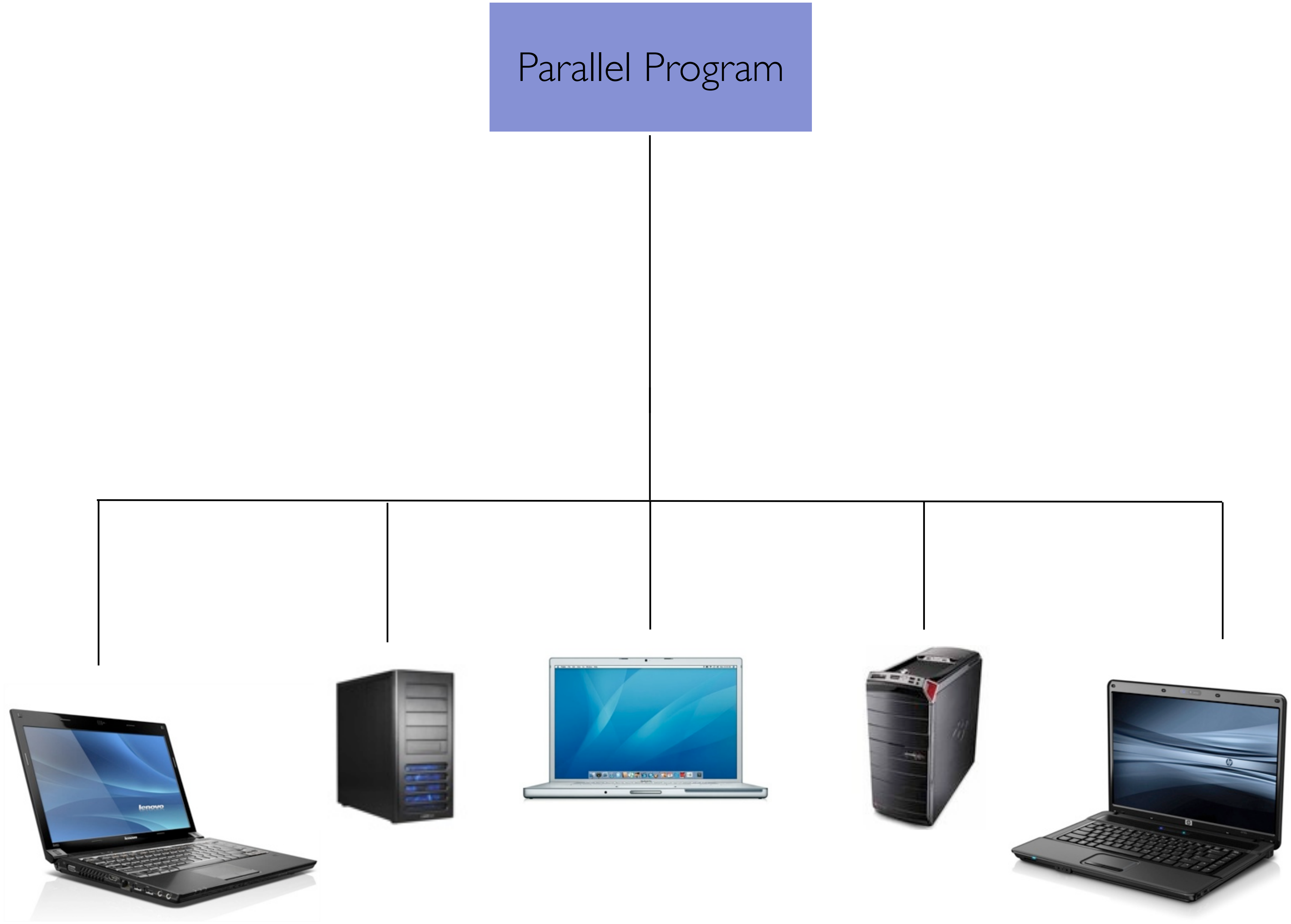
Software Engineering Seminar

Atune-IL: An Instrumentation Language for Auto-Tuning Parallel Applications

Christoph A. Schaefer, Victor Pankratius, Walter F. Tichy
Institue for Program Structures and Data Organization (IPD)
University of Karlsruhe
2009

Michael Berli, December 14th 2011

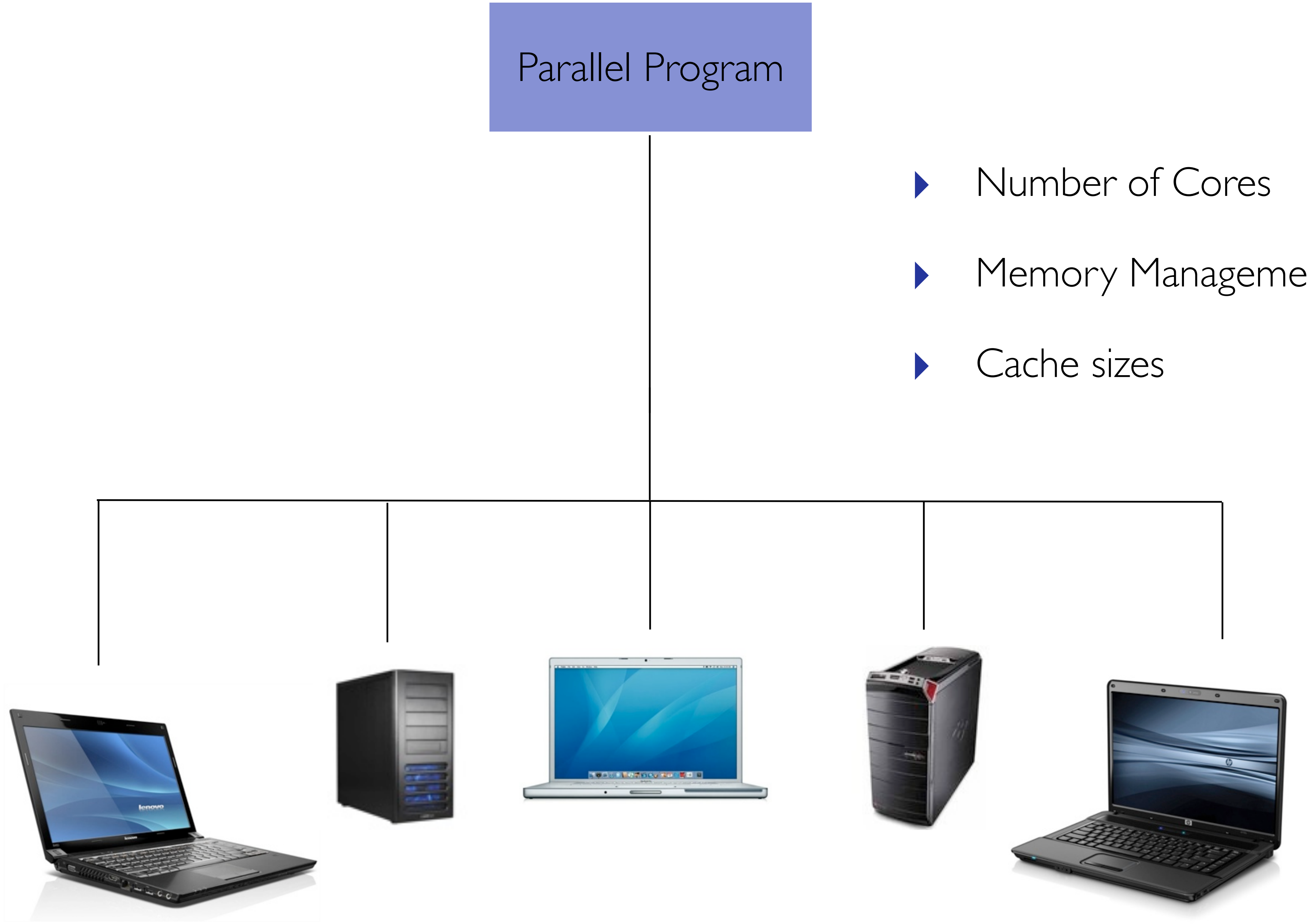
Motivation



Parallel Program

http://www.teknovadi.com/lenovo-laptop/lenovo-ideapad-b560/ http://lian-li.com/v2/tw/product/upload/image/pc-60fn/pc-60fn-26.jpg http://www.netzwelt.de/news/74776_3-apple-intel-produktuebersicht-erstes-fazit.html http://www.shoppydoo.co.uk/prices-desktop-packard_bell_ipower.html http://stodolatest.pl/produkt/HP_6735s_KU221EA/opinie/0/0/1

Motivation



- Number of Cores
- Memory Management
- Cache sizes

http://www.teknovadi.com/lenovo-laptop/lenovo-ideapad-b560/ http://lian-li.com/v2/tw/product/upload/image/pc-60fn/pc-60fn-26.jpg http://www.netzwelt.de/news/74776_3-apple-intel-produktuebersicht-erstes-fazit.html http://www.shoppydoo.co.uk/prices-desktop-packard_bell_ipower.html http://stodolatest.pl/produkt/HP_6735s_KU221EA/opinie/0/0/1

Motivation

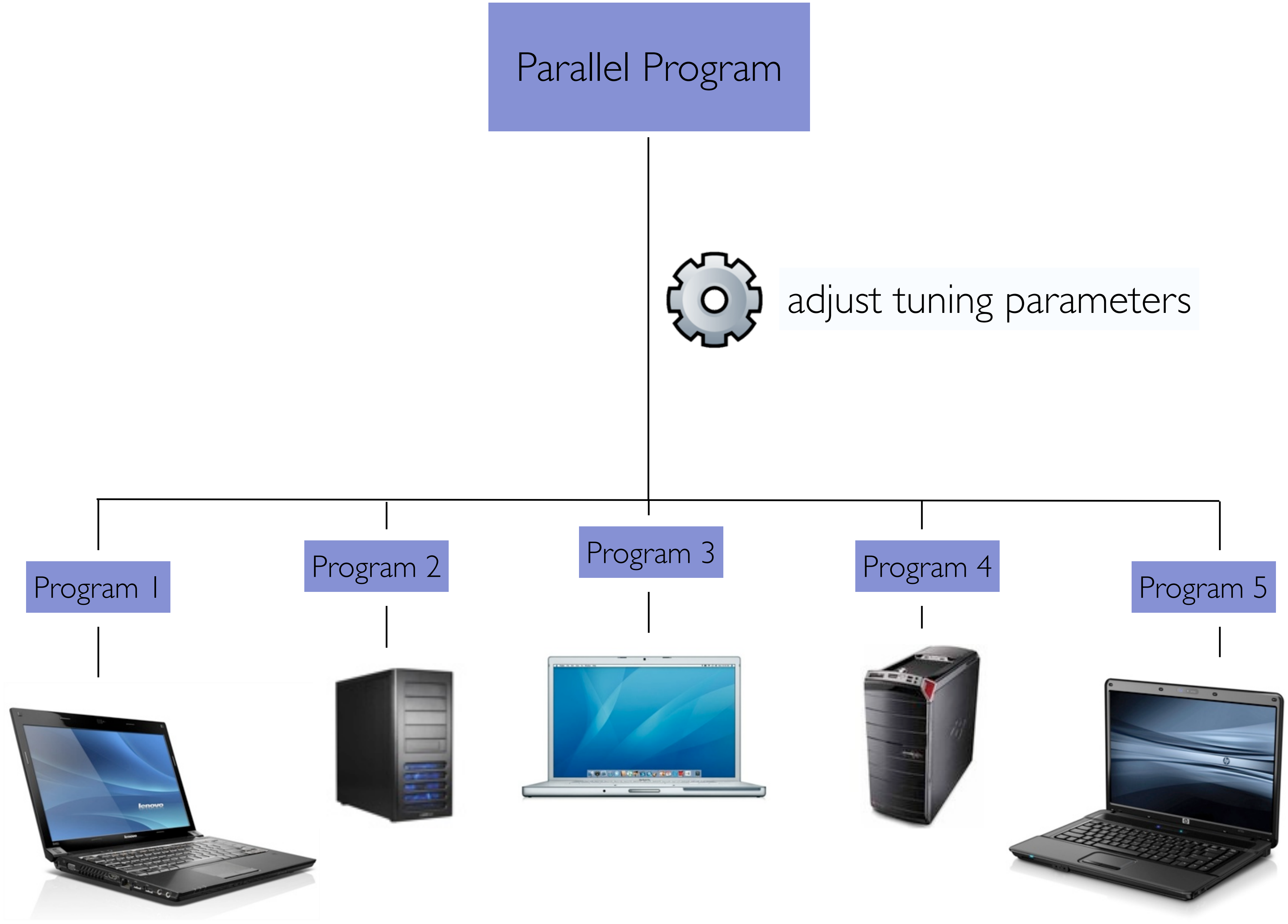
Parallel Program

- Number of Cores
- Memory Management
- Cache sizes

gain optimal performance

http://www.teknovadi.com/lenovo-laptop/lenovo-ideapad-b560/ http://lian-li.com/v2/tw/product/upload/image/pc-60fn/pc-60fn-26.jpg http://www.netzwelt.de/news/74776_3-apple-intel-produktuebersicht-erstes-fazit.html http://www.shoppydoo.co.uk/prices-desktop-packard_bell_ipower.html http://stodolatest.pl/produkt/HP_6735s_KU221EA/opinie/0/0/1

Motivation



Parallel Program

adjust tuning parameters

Program 1

Program 2

Program 3

Program 4

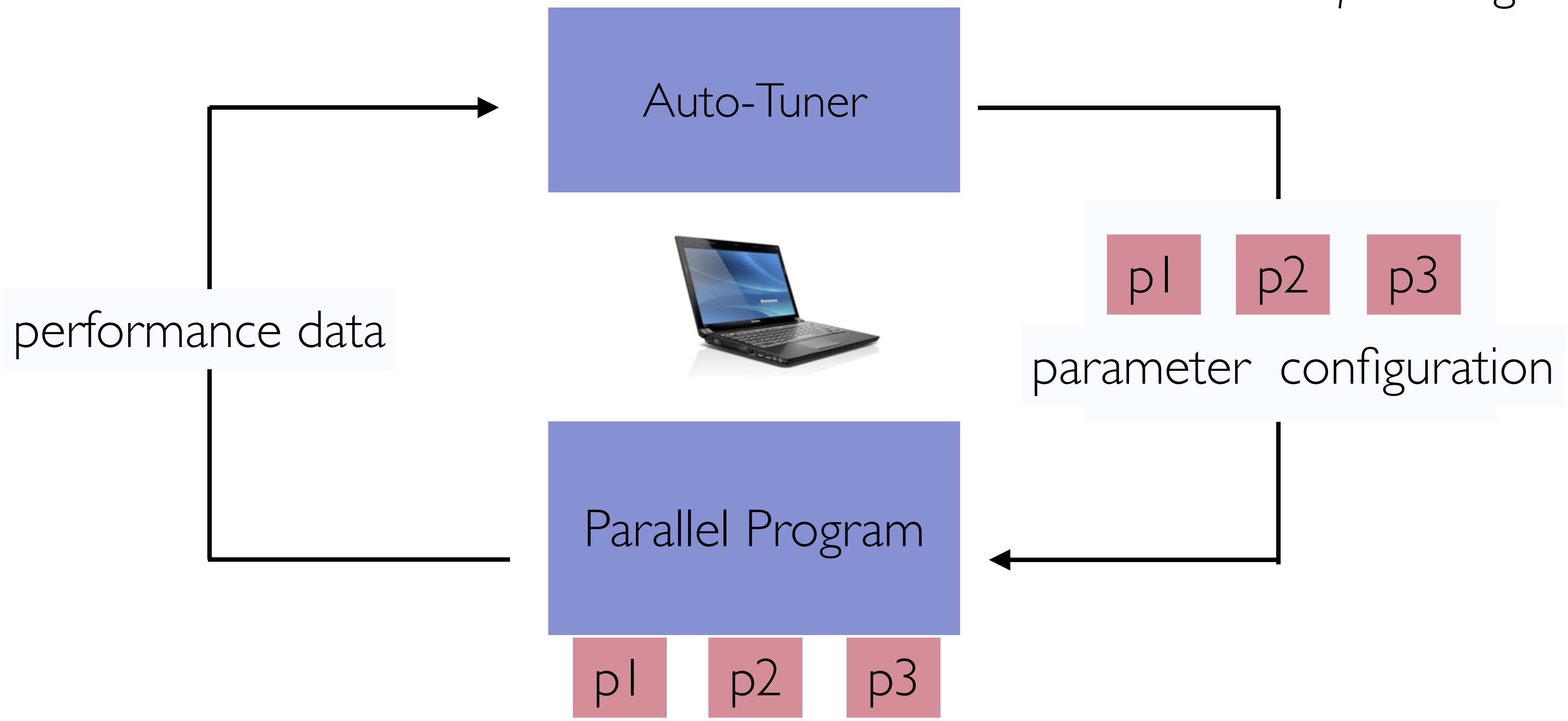
Program 5

http://www.teknovadi.com/lenovo-laptop/lenovo-ideapad-b560/ http://lian-li.com/v2/tw/product/upload/image/pc-60fn/pc-60fn-26.jpg http://www.netzwelt.de/news/74776_3-apple-intel-produktuebersicht-erstes-fazit.html http://www.shoppydoo.co.uk/prices-desktop-packard_bell_ipower.html http://stodolatest.pl/produkt/HP_6735s_KU221EA/opinie/0/0/1 http://www.iconarchive.com/show/soft-scraps-icons-by-deleket/Gear-icon.html

Automatic Performance Tuning

- Auto-Tuner: Generate several program variants automatically
  - on a specific architecture
  - find an optimal tuning parameter configuration

p1:	2, 4, 6, 8
p2:	„static“, „dynamic“
p3:	algo1, algo2

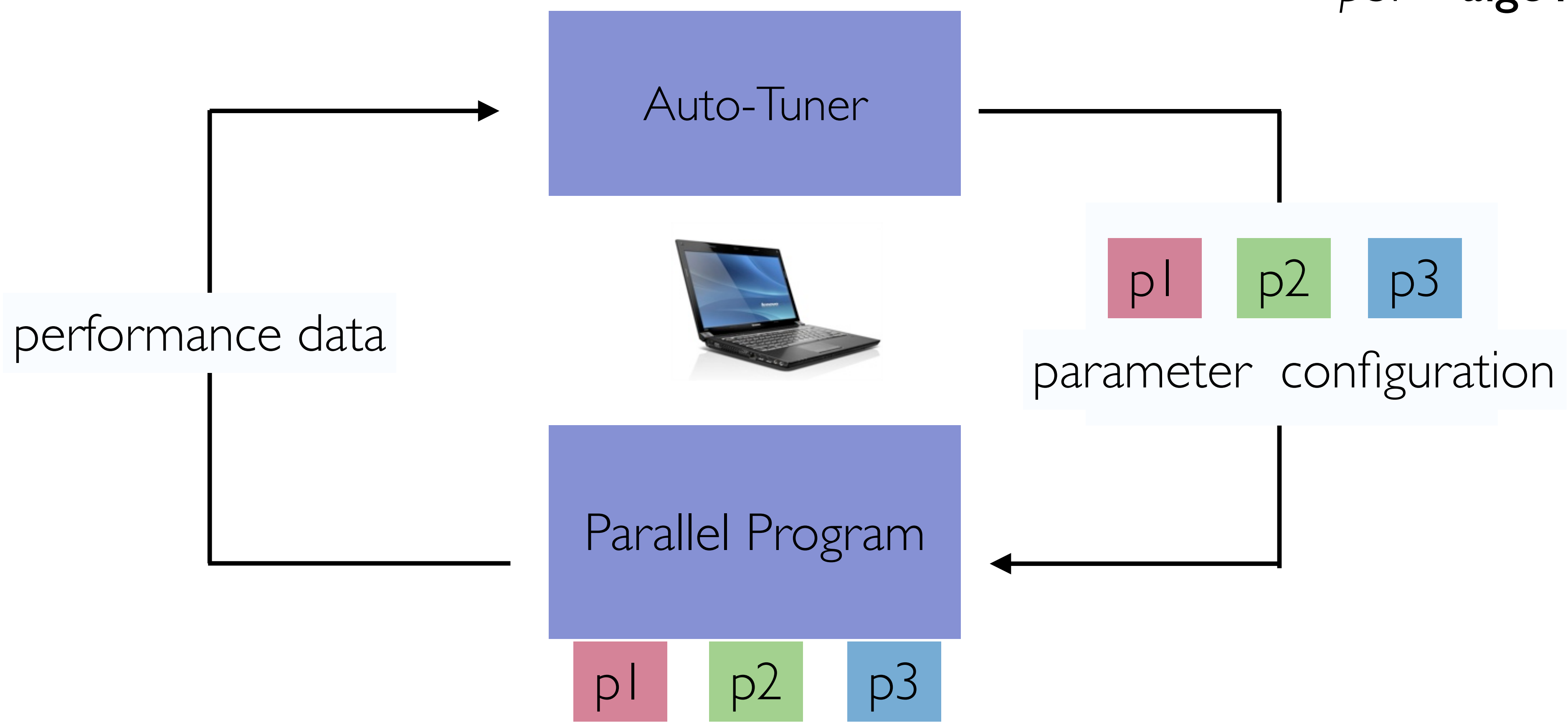


http://www.teknovadi.com/lenovo-laptop/lenovo-ideapad-b560/

Automatic Performance Tuning

- Auto-Tuner: Generate several program variants automatically
  - on a specific architecture
  - find an optimal tuning parameter configuration

p1: **2**, 4, 6, 8
p2: **static**, dynamic
p3: **algo1**, algo2

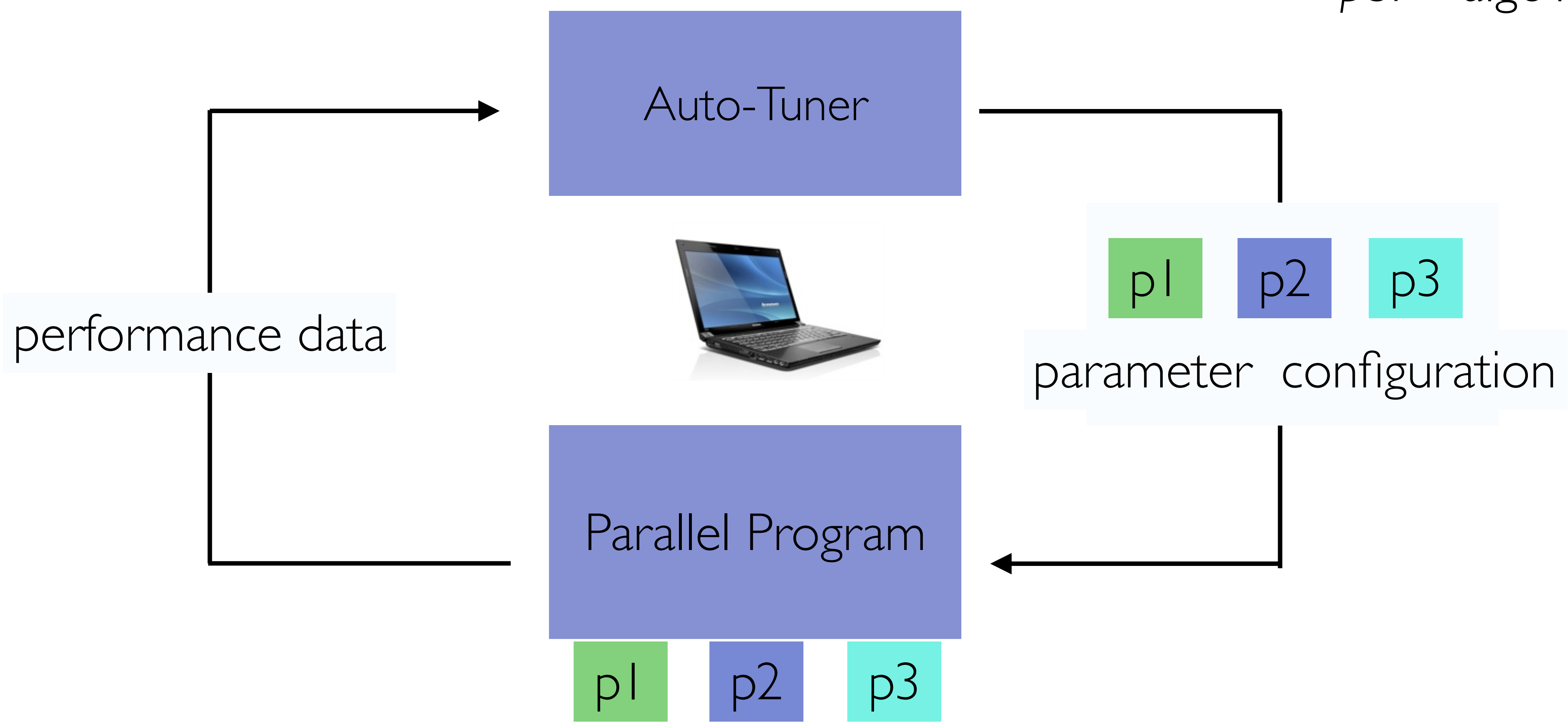


http://www.teknovadi.com/lenovo-laptop/lenovo-ideapad-b560/

Automatic Performance Tuning

- Auto-Tuner: Generate several program variants automatically
  - on a specific architecture
  - find an optimal tuning parameter configuration

p1:	2, **4**, 6, 8
p2:	**static**, dynamic
p3:	algo1, **algo2**

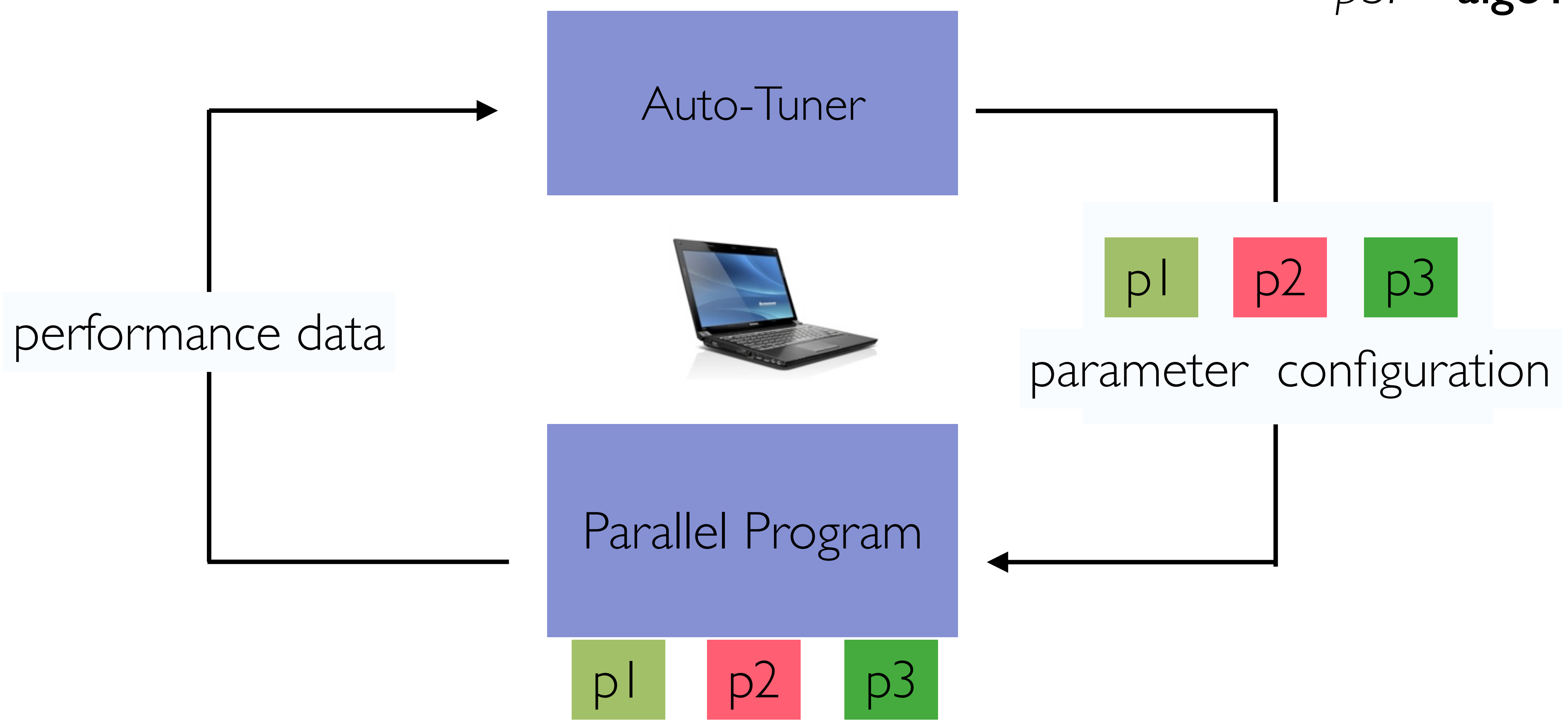


http://www.teknovadi.com/lenovo-laptop/lenovo-ideapad-b560/

Automatic Performance Tuning

- Auto-Tuner: Generate several program variants automatically
  - on a specific architecture
  - find an optimal tuning parameter configuration

p1: 2, 4, **6**, 8
p2: static, **dynamic**
p3: **algo1**, algo2



http://www.teknovadi.com/lenovo-laptop/lenovo-ideapad-b560/

Automatic Performance Tuning

- Huge search space
  - cross product of parameter domains

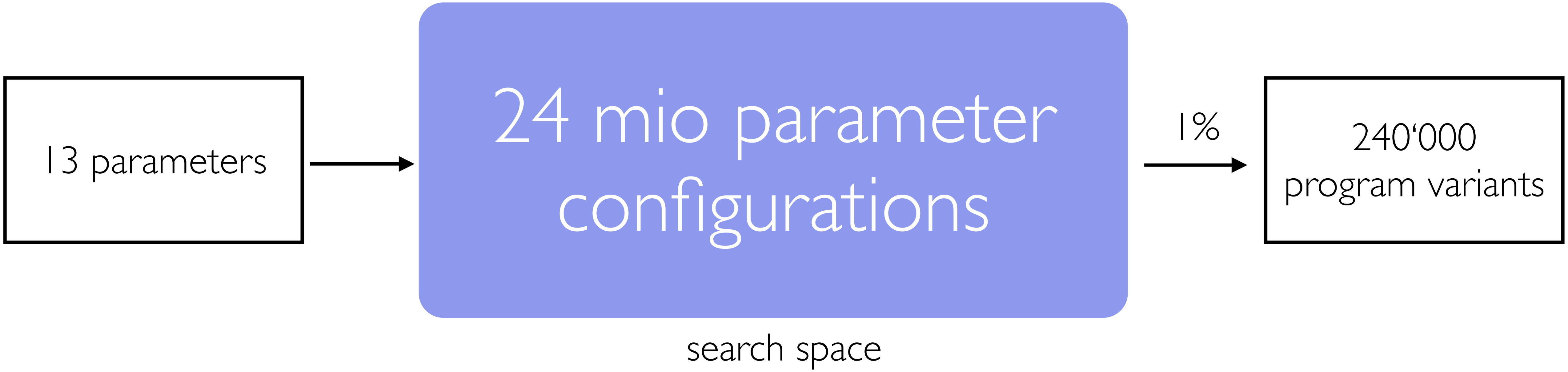
$$S = dom(p_1) \times ... \times dom(p_n)$$

p1:	2, 4, 6, 8	dom(p1) = 4
p2:	static, dynamic	dom(p2) = 2
p3:	algo1, algo2	dom(p3) = 2

Automatic Performance Tuning

- **Huge search space**
  - cross product of parameter domains

$$S = dom(p_1) \times ... \times dom(p_n)$$

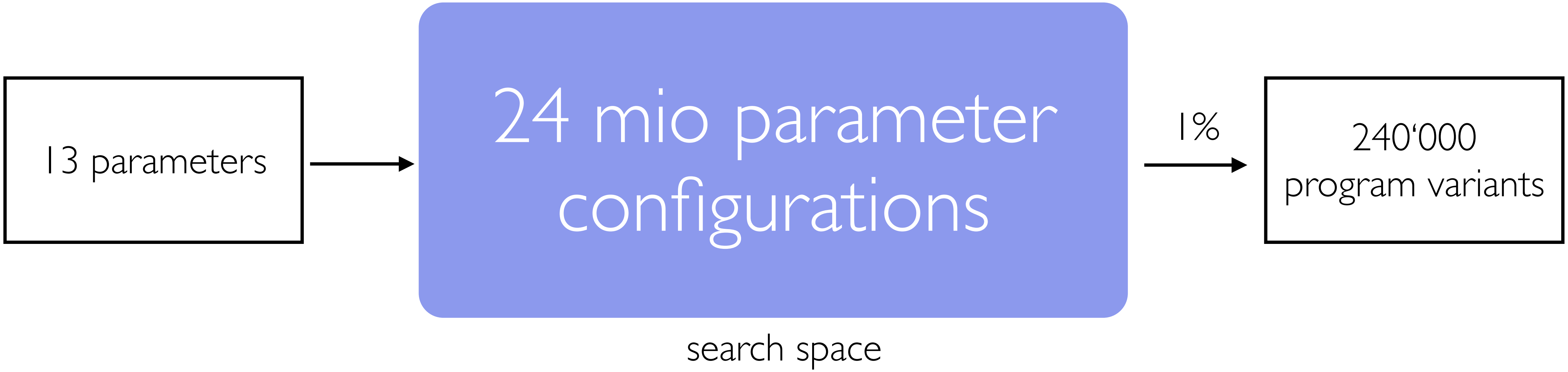


Automatic Performance Tuning

- **Huge search space**
  - cross product of parameter domains

$$S = dom(p_1) \times \ldots \times dom(p_n)$$

need to prune the search space !



13 parameters

24 mio parameter configurations

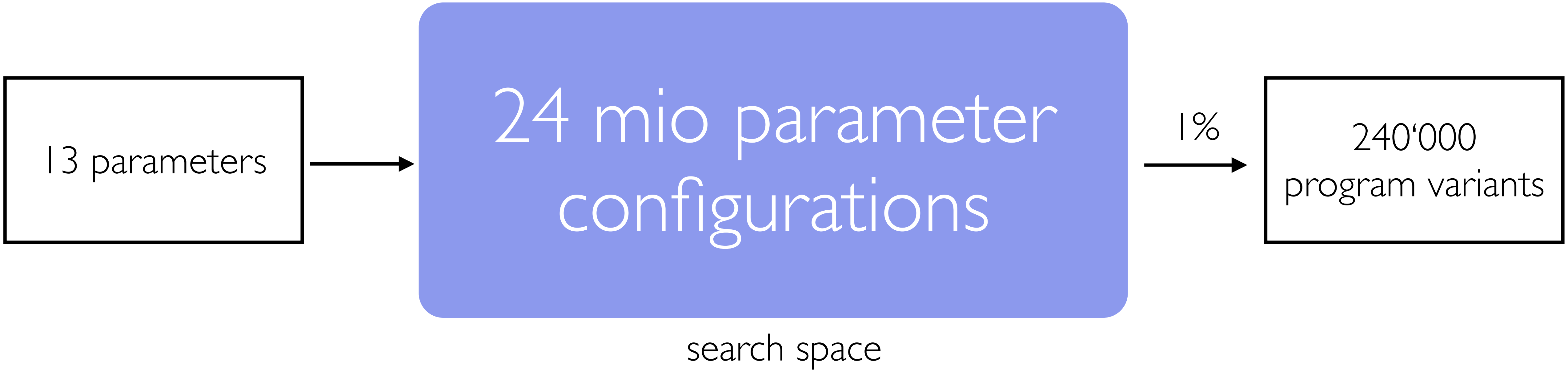
search space

1%

240‘000 program variants

Automatic Performance Tuning

- Three ways to prune the search space
  - try & fail
  - make use of heuristics / previous tuning iterations
  - use the developers knowledge



13 parameters

24 mio parameter configurations

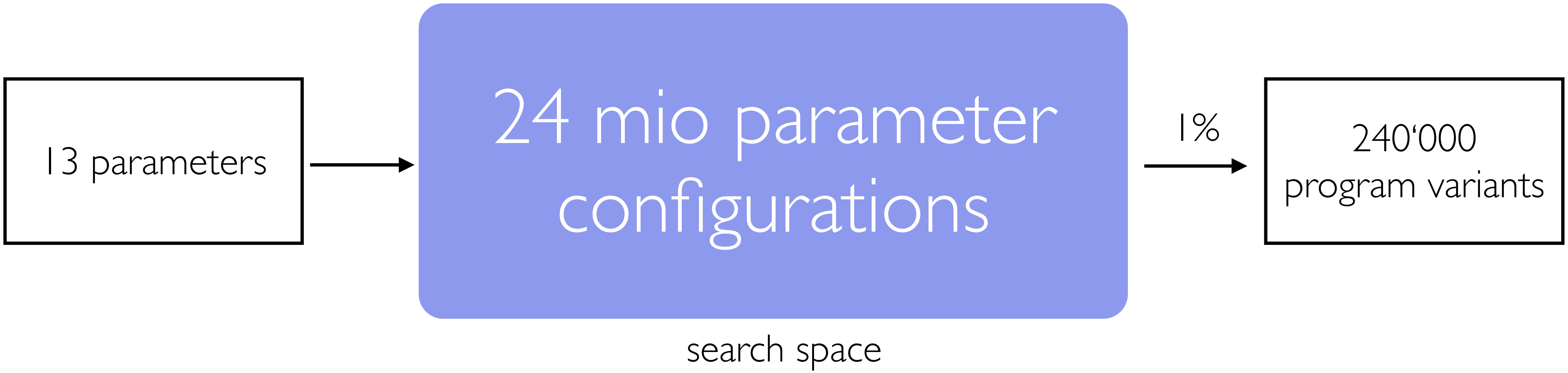
search space

1%

240‘000 program variants

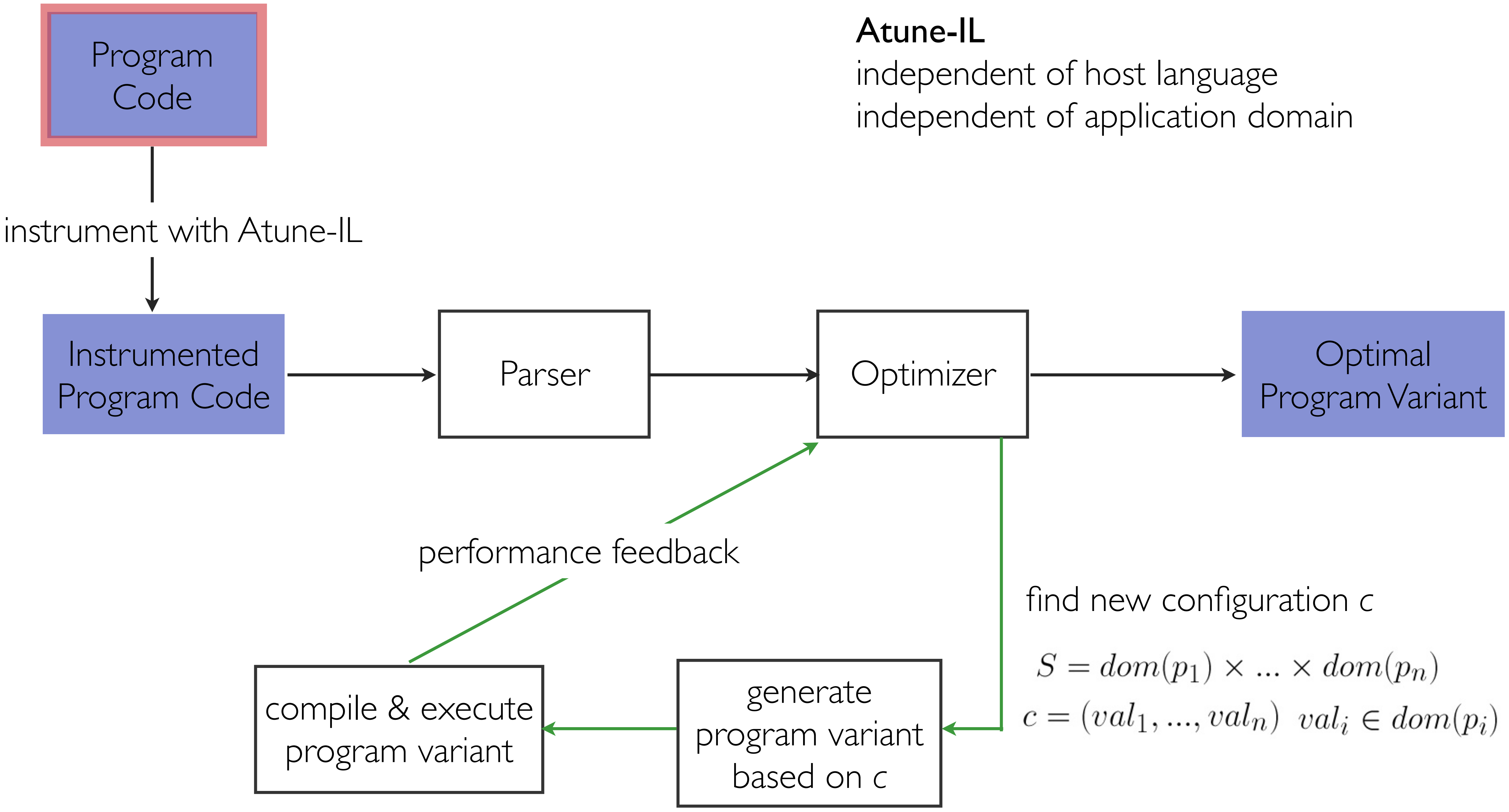
Automatic Performance Tuning

- Three ways to prune the search space
  - try & fail
  - make use of heuristics / previous tuning iterations
  - ✓ use the developers knowledge
    - Atune-IL: annotate tuning parameters, independent sections, monitoring probes...



Atune‘s tuning cycle

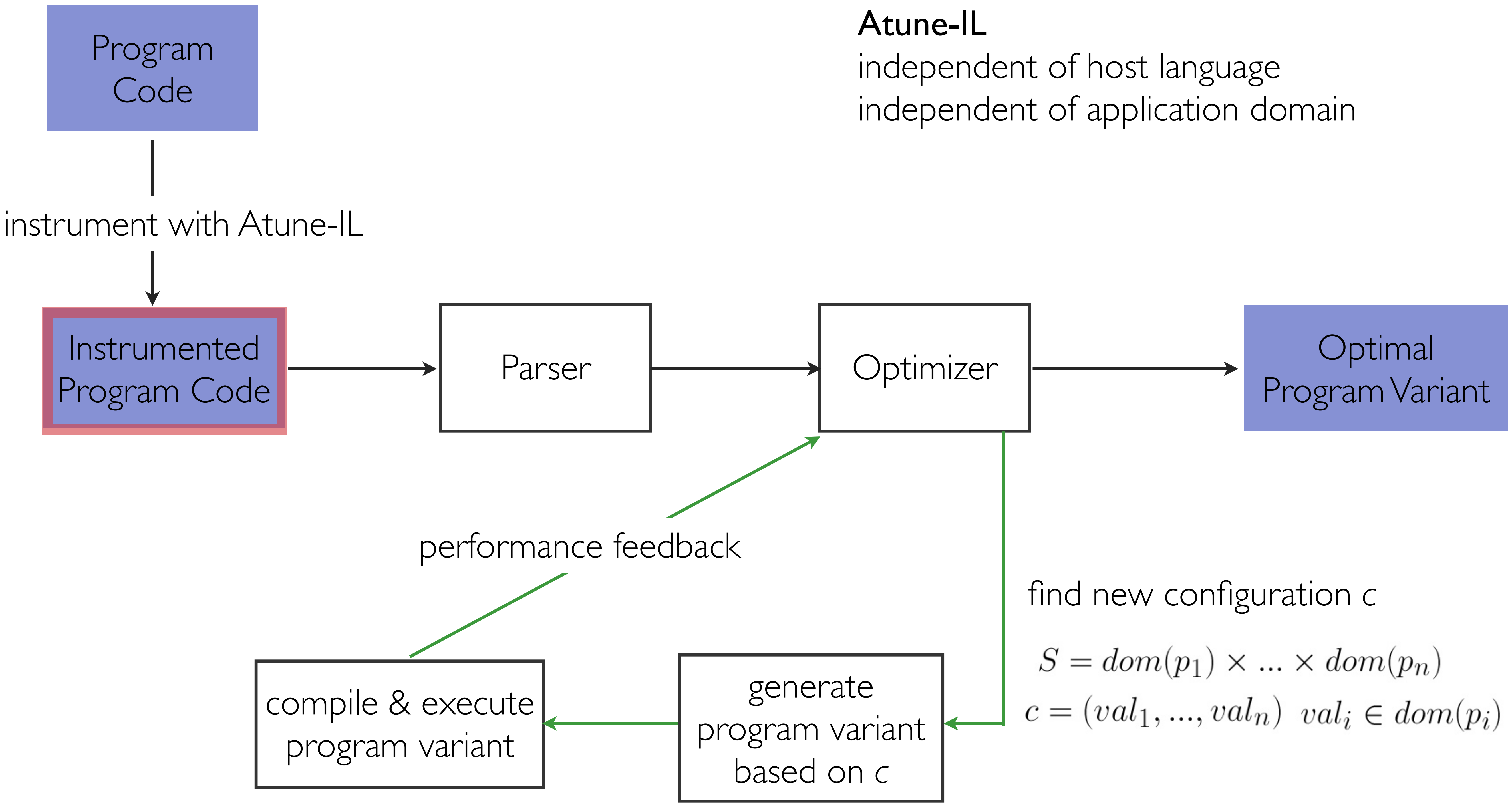
Atune-IL
independent of host language
independent of application domain



$$S = dom(p_1) \times ... \times dom(p_n)$$

$$c = (val_1, ..., val_n) \;\; val_i \in dom(p_i)$$

Atune‘s tuning cycle



$S = dom(p_1) \times ... \times dom(p_n)$

$c = (val_1, ..., val_n) \quad val_i \in dom(p_i)$

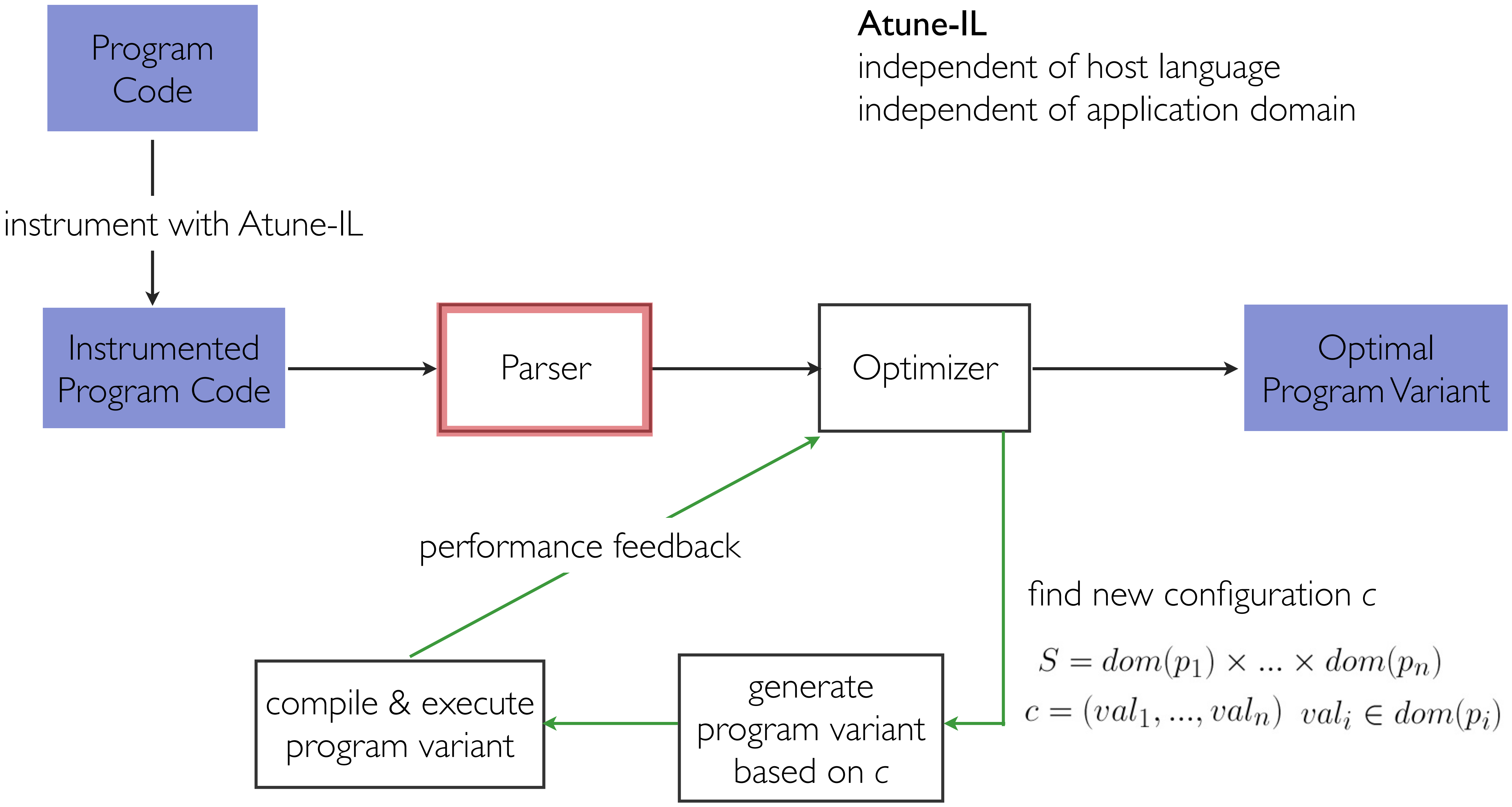
Atune-IL

independent of host language

independent of application domain

Atune‘s tuning cycle

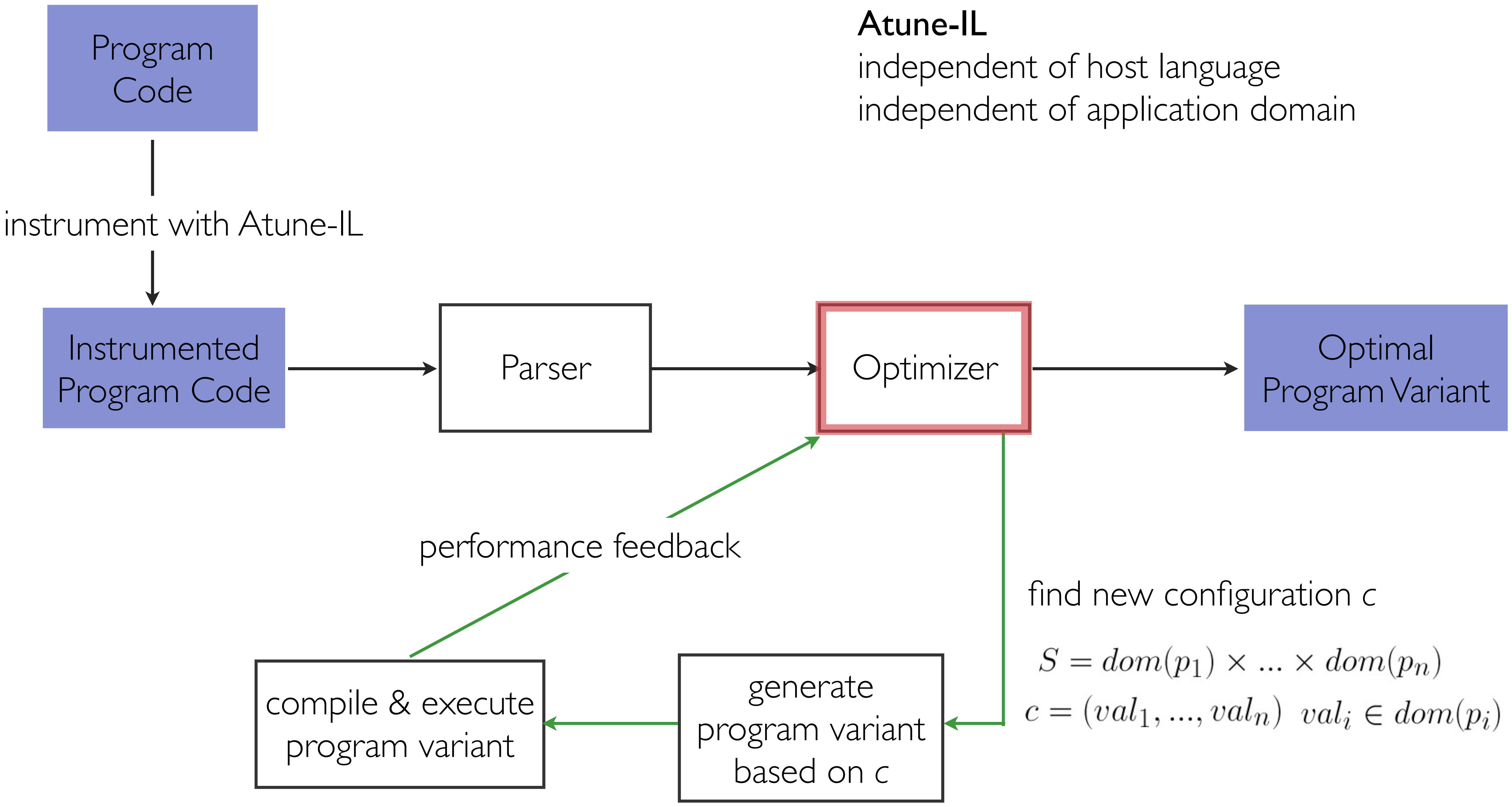
Atune-IL
independent of host language
independent of application domain



$$S = dom(p_1) \times ... \times dom(p_n)$$

$$c = (val_1, ..., val_n) \quad val_i \in dom(p_i)$$

Atune‘s tuning cycle



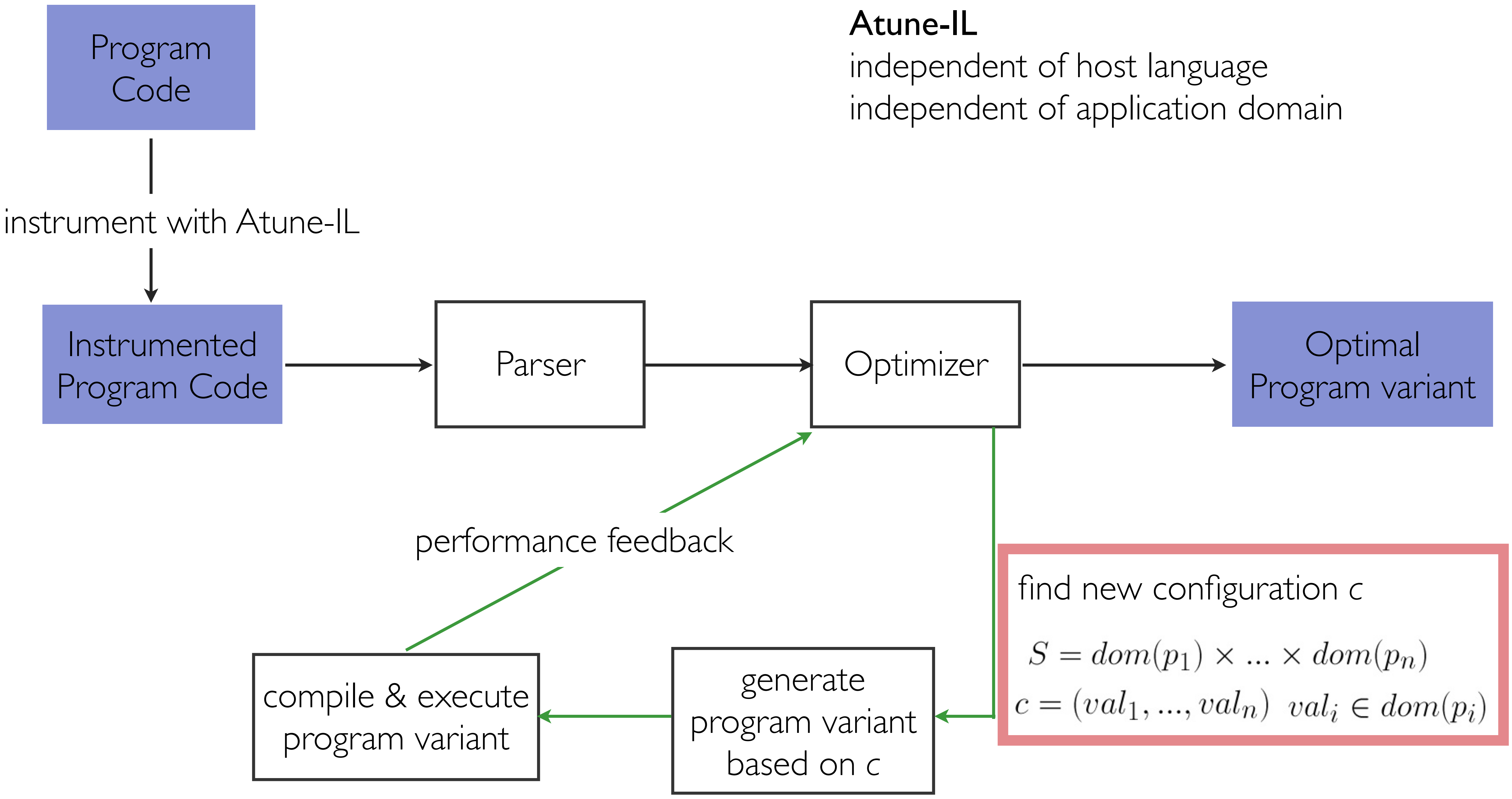
Atune-IL
independent of host language
independent of application domain

$S = dom(p_1) \times ... \times dom(p_n)$

$c = (val_1, ..., val_n) \;\; val_i \in dom(p_i)$

Atune‘s tuning cycle

Atune-IL
independent of host language
independent of application domain



Program Code

instrument with Atune-IL

Instrumented Program Code

Parser

Optimizer

Optimal Program variant

performance feedback

compile & execute program variant

generate program variant based on c

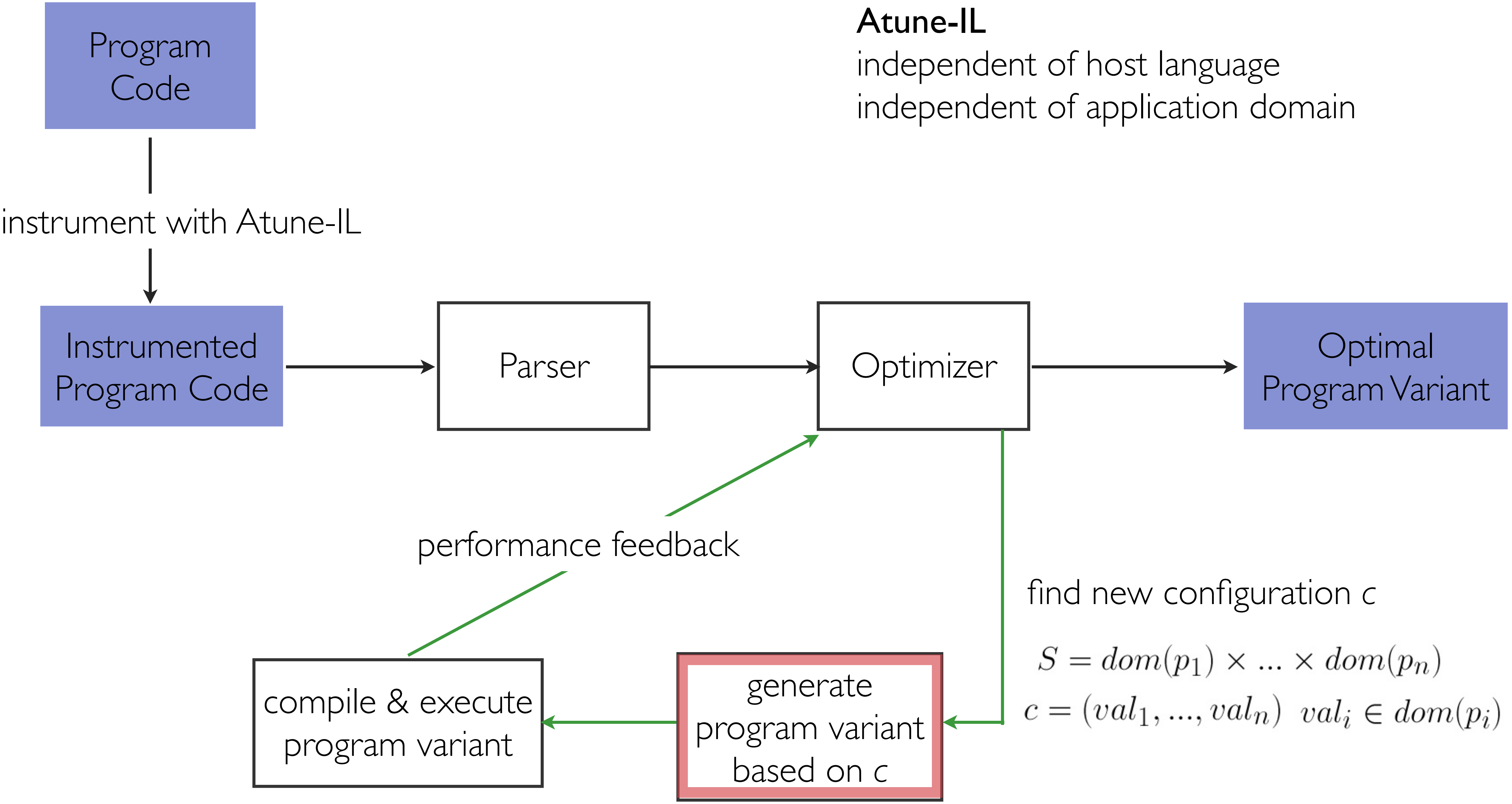
find new configuration c

$$S = dom(p_1) \times ... \times dom(p_n)$$

$$c = (val_1, ..., val_n) \quad val_i \in dom(p_i)$$

Atune‘s tuning cycle

Atune-IL
independent of host language
independent of application domain

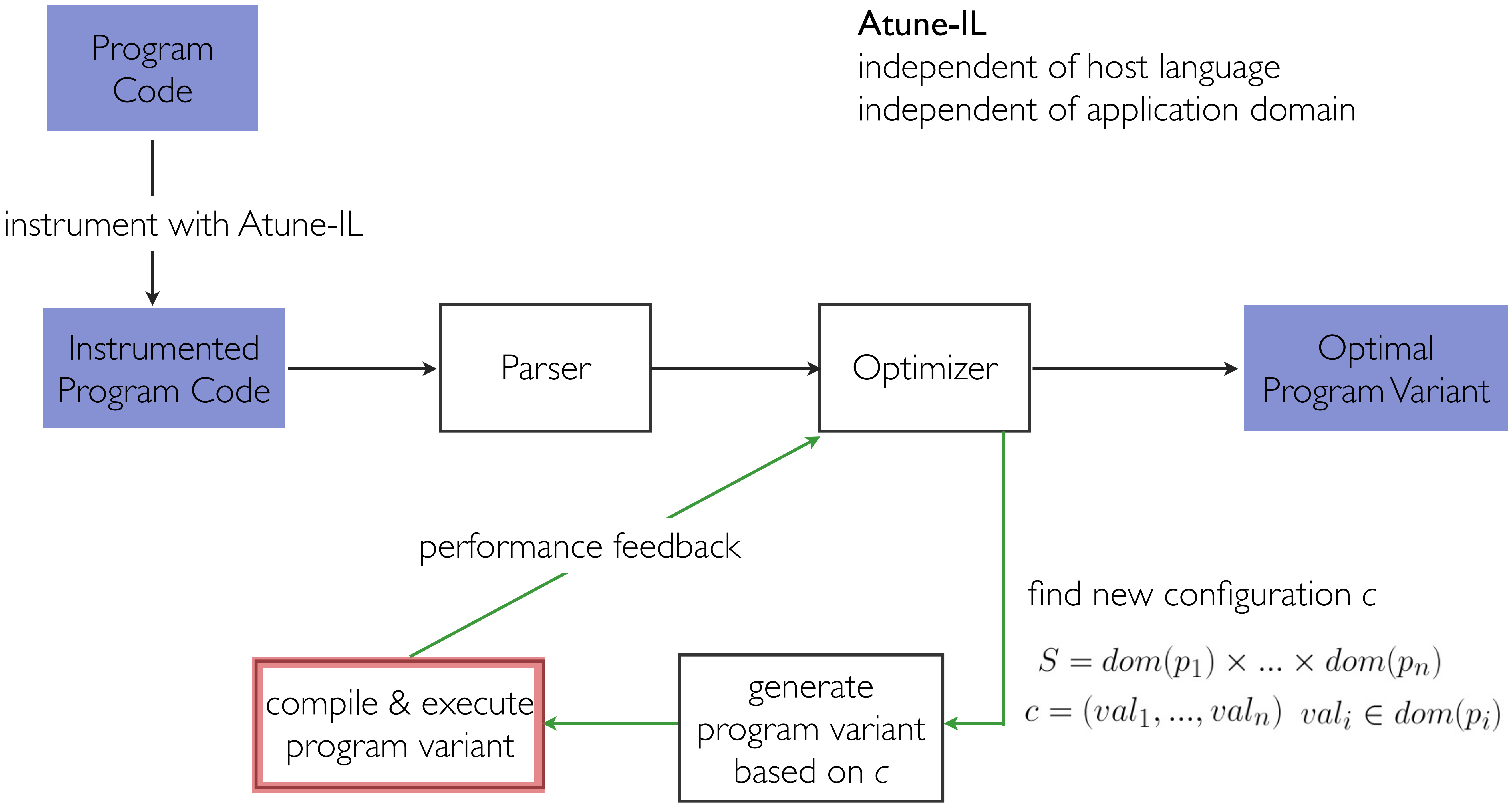


$$S = dom(p_1) \times ... \times dom(p_n)$$

$$c = (val_1, ..., val_n) \quad val_i \in dom(p_i)$$

Atune‘s tuning cycle

Atune-IL
independent of host language
independent of application domain



Program Code

instrument with Atune-IL

Instrumented Program Code

Parser

Optimizer

Optimal Program Variant

performance feedback

find new configuration c

$$S = dom(p_1) \times ... \times dom(p_n)$$

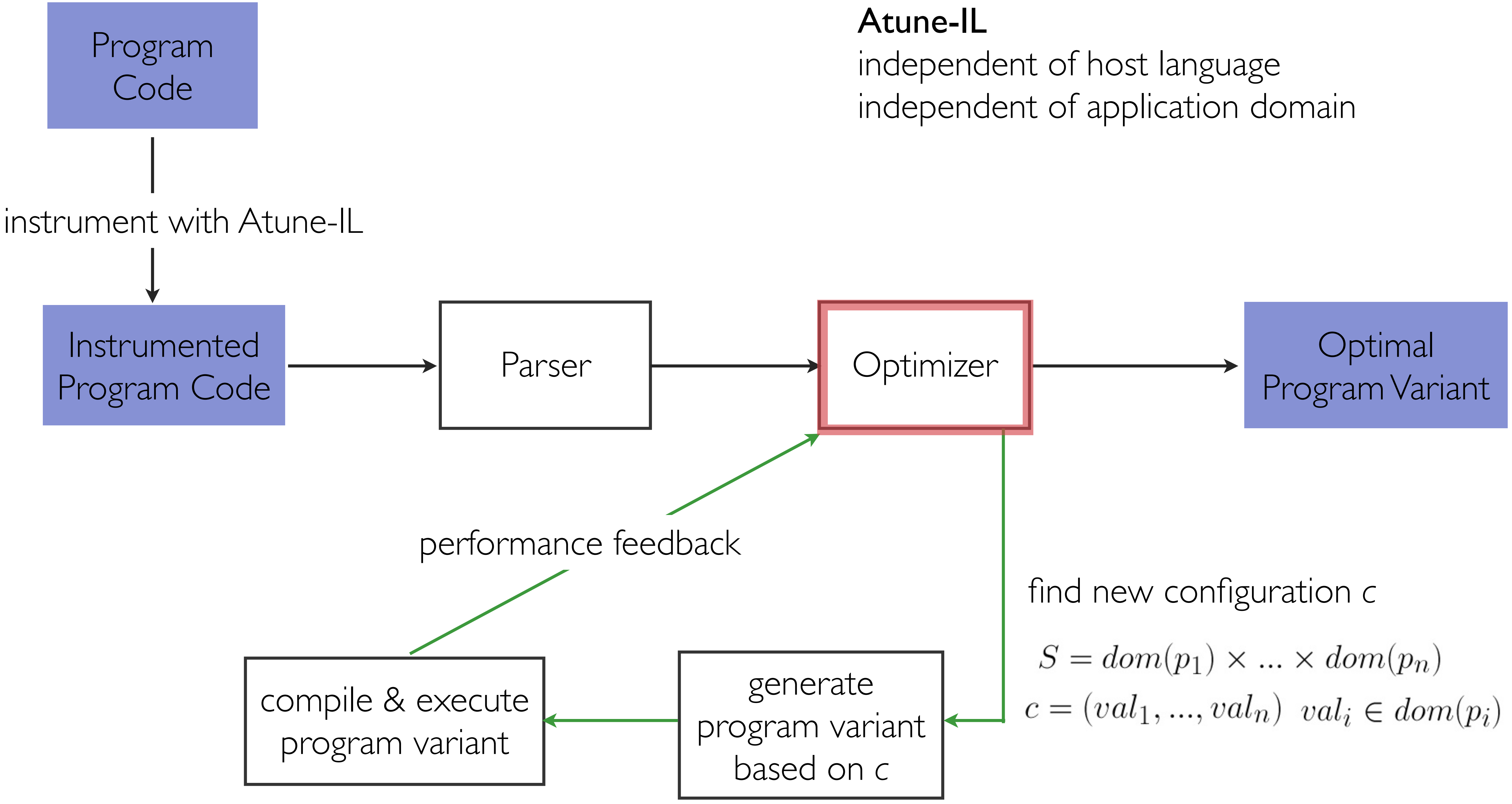
$$c = (val_1, ..., val_n) \quad val_i \in dom(p_i)$$

generate program variant based on c

compile & execute program variant

Atune‘s tuning cycle

Atune-IL
independent of host language
independent of application domain

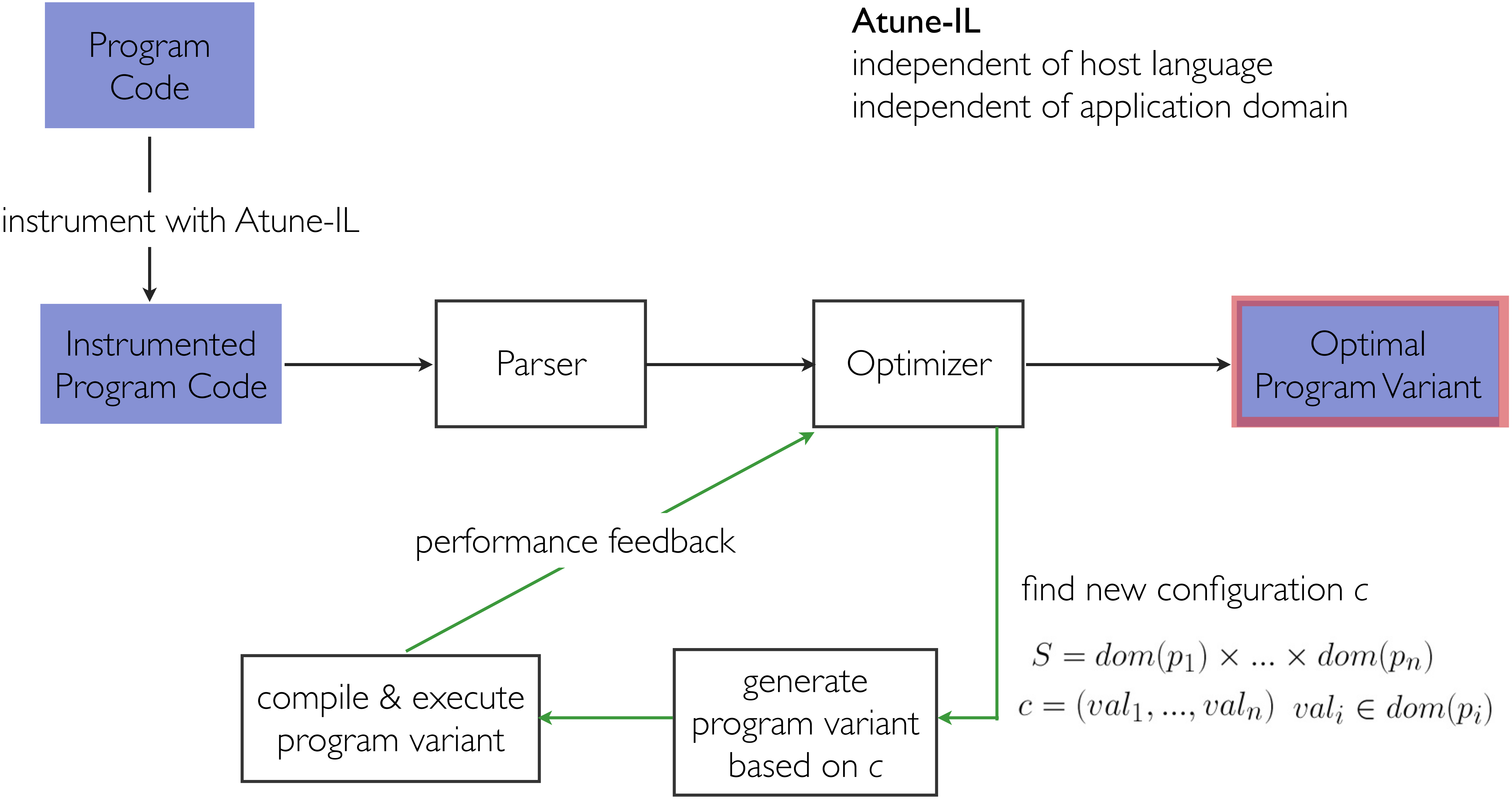


$S = dom(p_1) \times ... \times dom(p_n)$

$c = (val_1, ..., val_n) \;\; val_i \in dom(p_i)$

Atune‘s tuning cycle

Atune-IL
independent of host language
independent of application domain

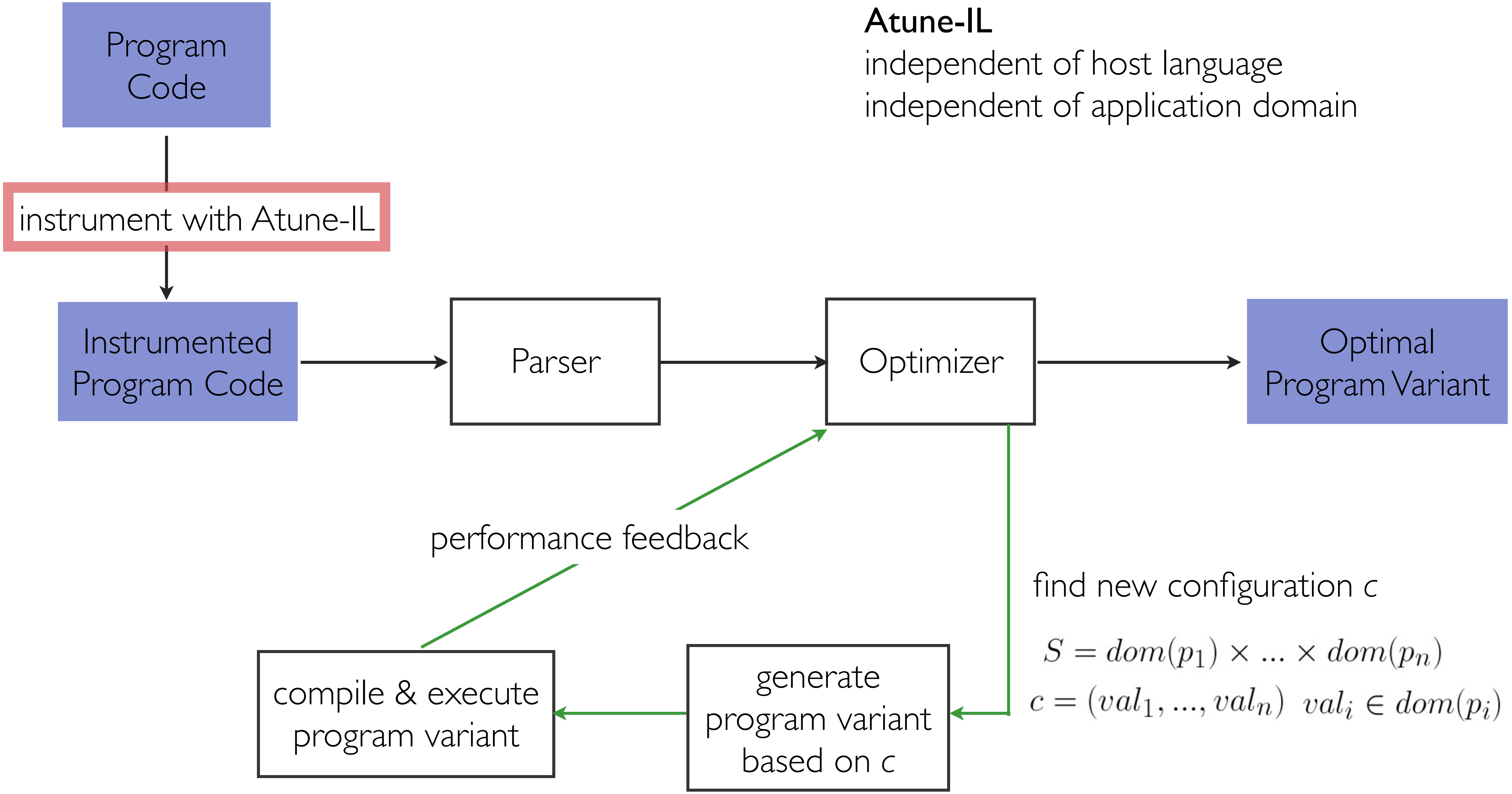


$$S = dom(p_1) \times ... \times dom(p_n)$$

$$c = (val_1, ..., val_n) \;\; val_i \in dom(p_i)$$

Atune‘s tuning cycle

Atune-IL
independent of host language
independent of application domain



Program Code

instrument with Atune-IL

Instrumented Program Code

Parser

Optimizer

Optimal Program Variant

find new configuration c

$$S = dom(p_1) \times ... \times dom(p_n)$$

$$c = (val_1, ..., val_n) \quad val_i \in dom(p_i)$$

generate program variant based on c

compile & execute program variant

performance feedback

Numeric Parameters

- SETVAR keyword

```
public void SETVAR_Example()
{
  int numThreads = 2;



  for (int i=1; i <=numThreads; i++){
    Thread.Create(StartCalculation);
  }
  WaitAll();
}
```

Numeric Parameters

- SETVAR keyword

```
public void SETVAR_Example()
{
  int numThreads = 2;
  #pragma atune SETVAR numThreads TYPE int
  VALUES 2-16 STEP 2

  for (int i=1; i <=numThreads; i++){
    Thread.Create(StartCalculation);
  }
  WaitAll();
}
```

2, 4, ..., 16 Threads

Architectural Parameters

- SETVAR keyword

```
public void SETVAR_Example2()
{
  SortAlgorithm sortAlgo = new ParallelMergeSort();

  #pragma atune SETVAR sortAlgo TYPE generic VALUES
  „new QuickSort()“, „new ParallelMergeSort()“

  if (sortAlgo != null)
    sortAlgo.run();
}
```

Parameter Dependencies

- DEPENDS keyword

```
public void DEPENDS_Example()
{
  SortAlgorithm sortAlgo = new ParallelMergeSort();

  #pragma atune SETVAR sortAlgo TYPE generic VALUES
  „new QuickSort()“, „new ParallelMergeSort()“

    int depth = 2;

    #pragma atune SETVAR depth TYPE int VALUES 2-8

  if (sortAlgo != null)
    sortAlgo.run(depth);
}
```

14 combinations

Parameter Dependencies

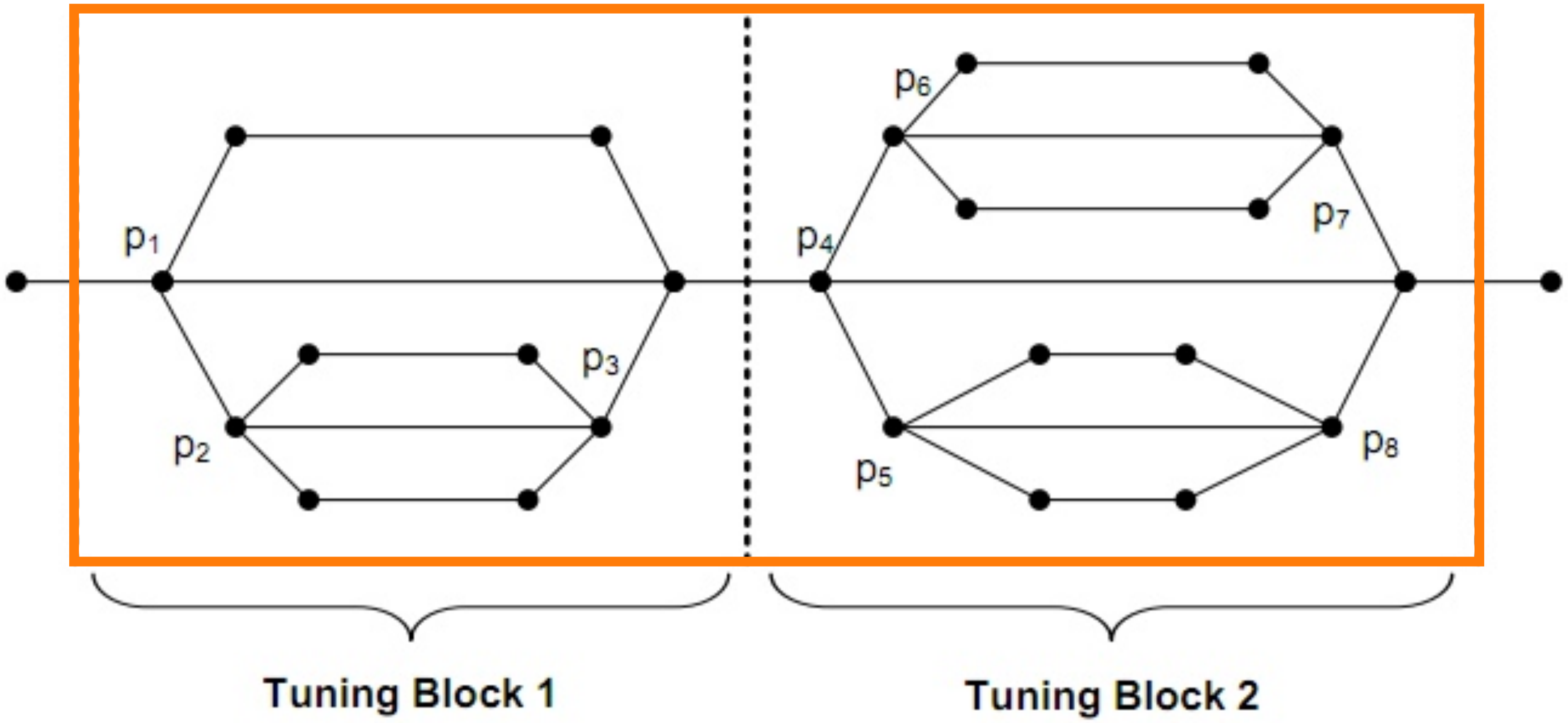
- DEPENDS keyword

```
public void DEPENDS_Example()
{
  SortAlgorithm sortAlgo = new ParallelMergeSort();

  #pragma atune SETVAR sortAlgo TYPE generic VALUES
  „new QuickSort()“, „new ParallelMergeSort()“

    int depth = 2;

    #pragma atune SETVAR depth TYPE int VALUES 2-8
    DEPENDS sortAlgo VALUES “new ParallelMergeSort()“

  if (sortAlgo != null)
    sortAlgo.run(depth);
}
```

8 combinations
instead of 14!

Tuning Blocks

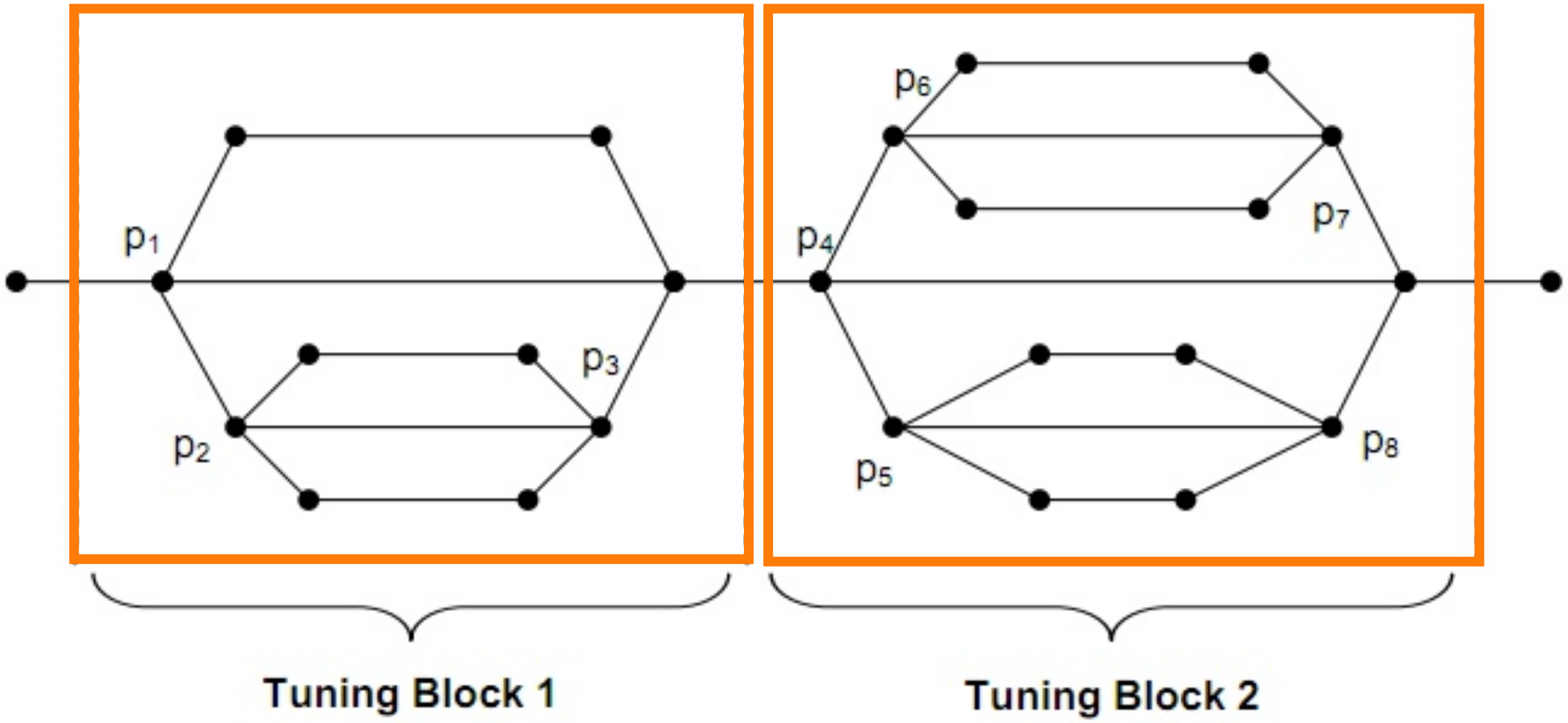
- Define independent sections
  - Tuning-Approach 1: $dom(p_1) \times ... \times dom(p_8)$
  - **Tuning-Approach 2:**
    - Tuning-Block 1: $dom(p_1) \times ... \times dom(p_3)$
    - Tuning-Block 2: $dom(p_4) \times ... \times dom(p_8)$



http://www.ipd.uni-karlsruhe.de/multicore/research/download/ATuneIL-Autotuning.pdf

Tuning Blocks

- Define independent sections
  - Tuning-Approach 1: $dom(p_1) \times ... \times dom(p_8)$
  - Tuning-Approach 2:
    - Tuning-Block 1: $dom(p_1) \times ... \times dom(p_3)$
    - Tuning-Block 2: $dom(p_4) \times ... \times dom(p_8)$



http://www.ipd.uni-karlsruhe.de/multicore/research/download/ATuneIL-Autotuning.pdf

Tuning Blocks

- STARTBLOCK keyword

```
public void TUNINGBLOCKS_Example()
{
    // other tuning parameters...


     #pragma atune STARTBLOCK parallelSection
    int numThreads = 2;
    #pragma atune SETVAR numThreads
    TYPE int VALUES 2-16 STEP 2

    for (int i=1; i <=numThreads; i++){
      Thread.Create(StartCalculation);
    }
    WaitAll();
     #pragma atune ENDBLOCK

}
```

Nested Tuning Blocks

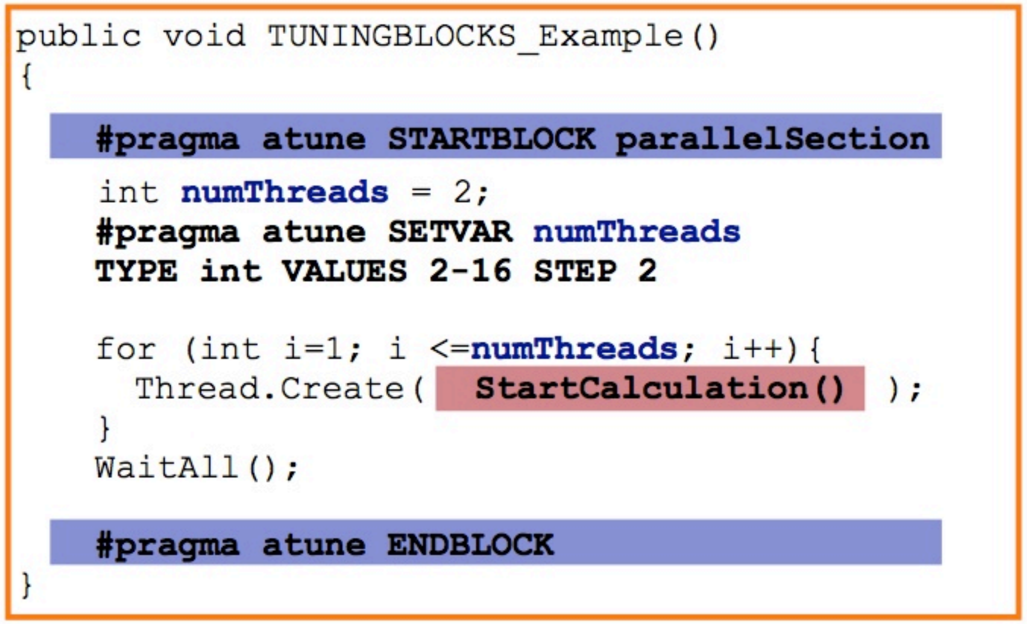
- INSIDE keyword
  - optimization starts with the innermost block
  - combines inner and outer parameters successively

```
public void TUNINGBLOCKS_Example()
{
    #pragma atune STARTBLOCK parallelSection
    int numThreads = 2;
    #pragma atune SETVAR numThreads
    TYPE int VALUES 2-16 STEP 2

    for (int i=1; i <=numThreads; i++){
      Thread.Create(  StartCalculation()  );
    }
    WaitAll();

    #pragma atune ENDBLOCK
}
```

Nested Tuning Blocks

- INSIDE keyword
  - optimization starts with the innermost block
  - combines inner and outer parameteters successively

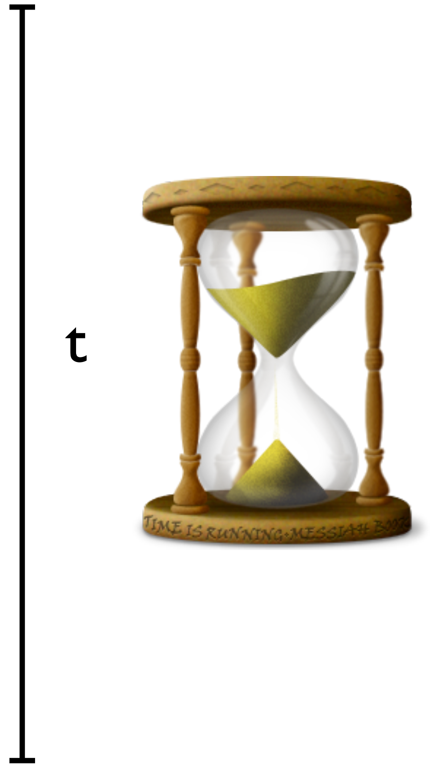
```
public void  StartCalculation()
{
    #pragma atune STARTBLOCK
    nestedSection INSIDE parallelSection

    /* calculation with own tuning
    parameters */

    #pragma atune ENDBLOCK
}
```



```
public void TUNINGBLOCKS_Example()
{
   #pragma atune STARTBLOCK parallelSection
   int numThreads = 2;
   #pragma atune SETVAR numThreads
   TYPE int VALUES 2-16 STEP 2

   for (int i=1; i <=numThreads; i++){
     Thread.Create( StartCalculation() );
   }
   WaitAll();

   #pragma atune ENDBLOCK
}
```

Monitoring Probes

```
public void TUNINGBLOCKS_Example()
{
     #pragma atune STARTBLOCK parallelSection
      #pragma atune GAUGE execTime
    int numThreads = 2;
    #pragma atune SETVAR numThreads
    TYPE int VALUES 2-16 STEP 2

    for (int i=1; i <=numThreads; i++){
      Thread.Create(StartCalculation());
    }
    WaitAll();
      #pragma atune GAUGE execTime
     #pragma atune ENDBLOCK
}
```



http://www.iconfinder.com/icondetails/48883/256/

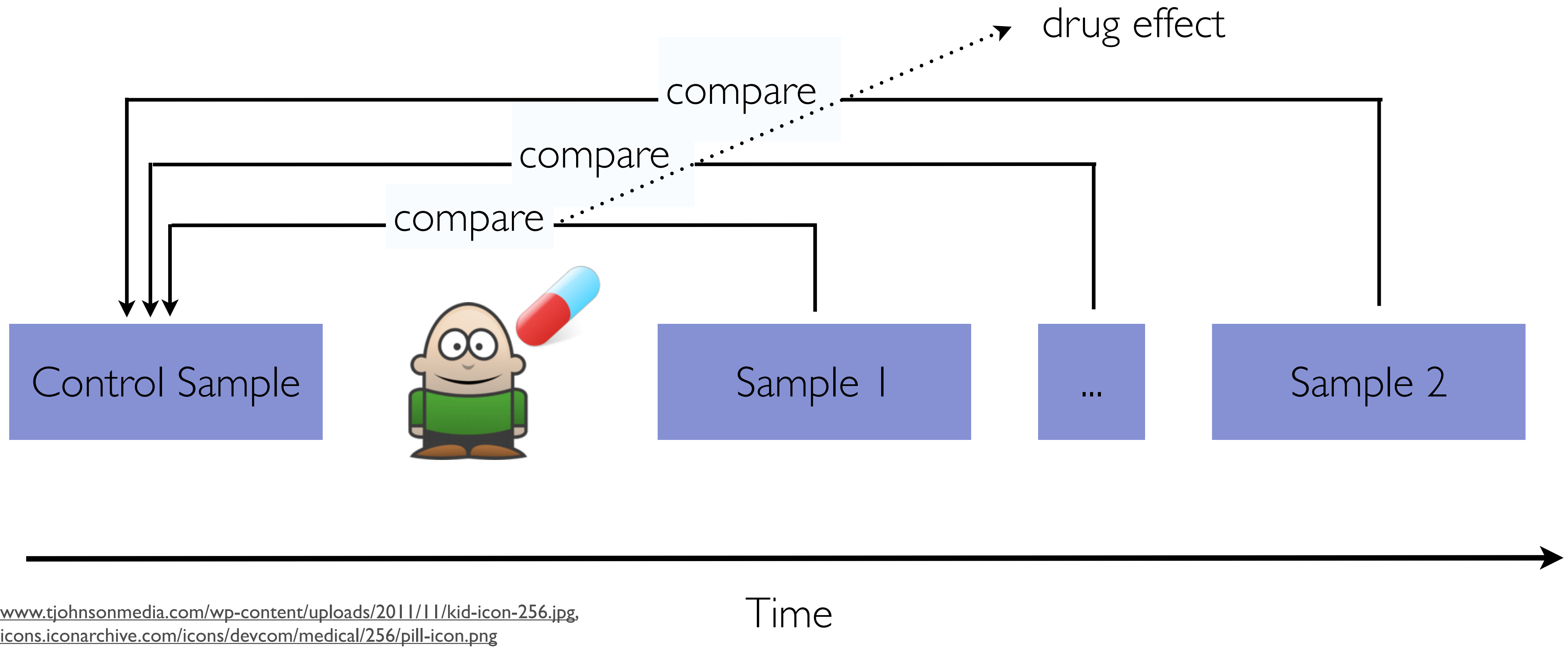
Context

- ✓ Motivation
- ✓ Introduction to Auto-Tuning
- ✓ Atune‘s Tuning Cycle
- ✓ Atune-IL
- ▸ Case Study
  - ▸ Results
- ▸ Pros & Cons

http://images.productwiki.com/upload/images/safari_iphone_app-400-400.jpg

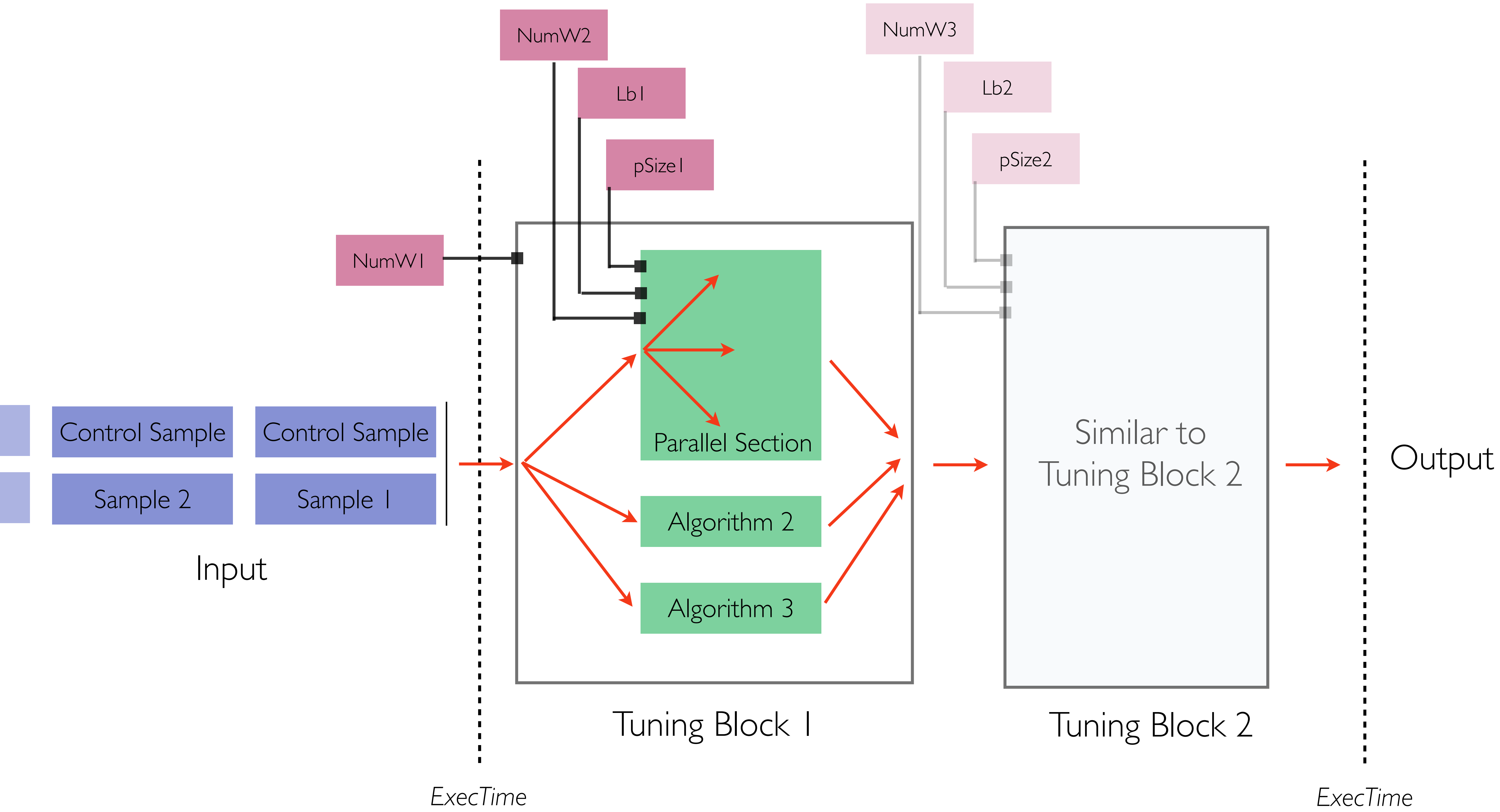
Case Study

- MetaboliteID (Agilent Technologies)
- Identify effects caused by a drug on a very low level
  - by comparing control samples to metabolite samples

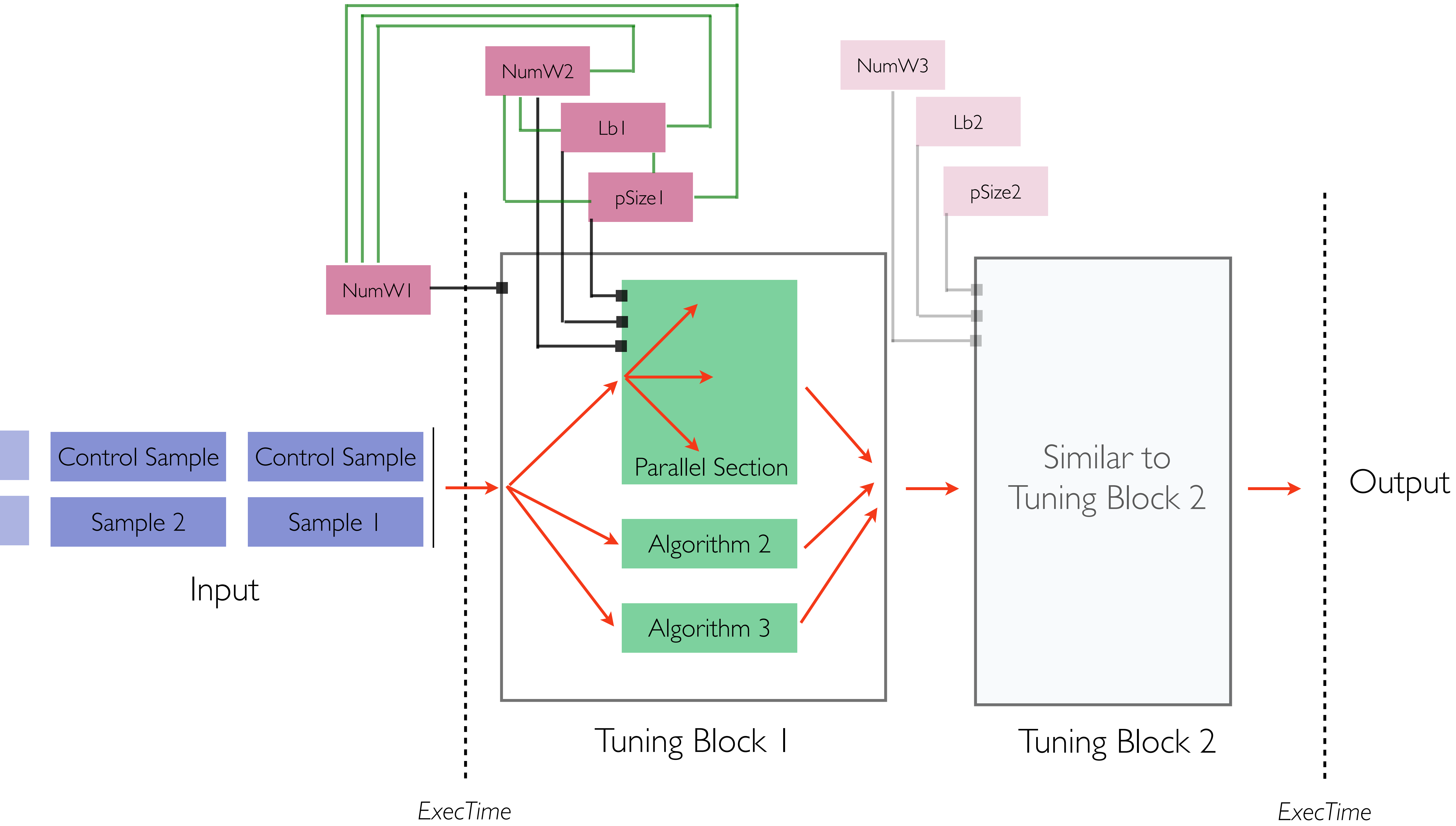


http://www.tjohnsonmedia.com/wp-content/uploads/2011/11/kid-icon-256.jpg,
http://icons.iconarchive.com/icons/devcom/medical/256/pill-icon.png

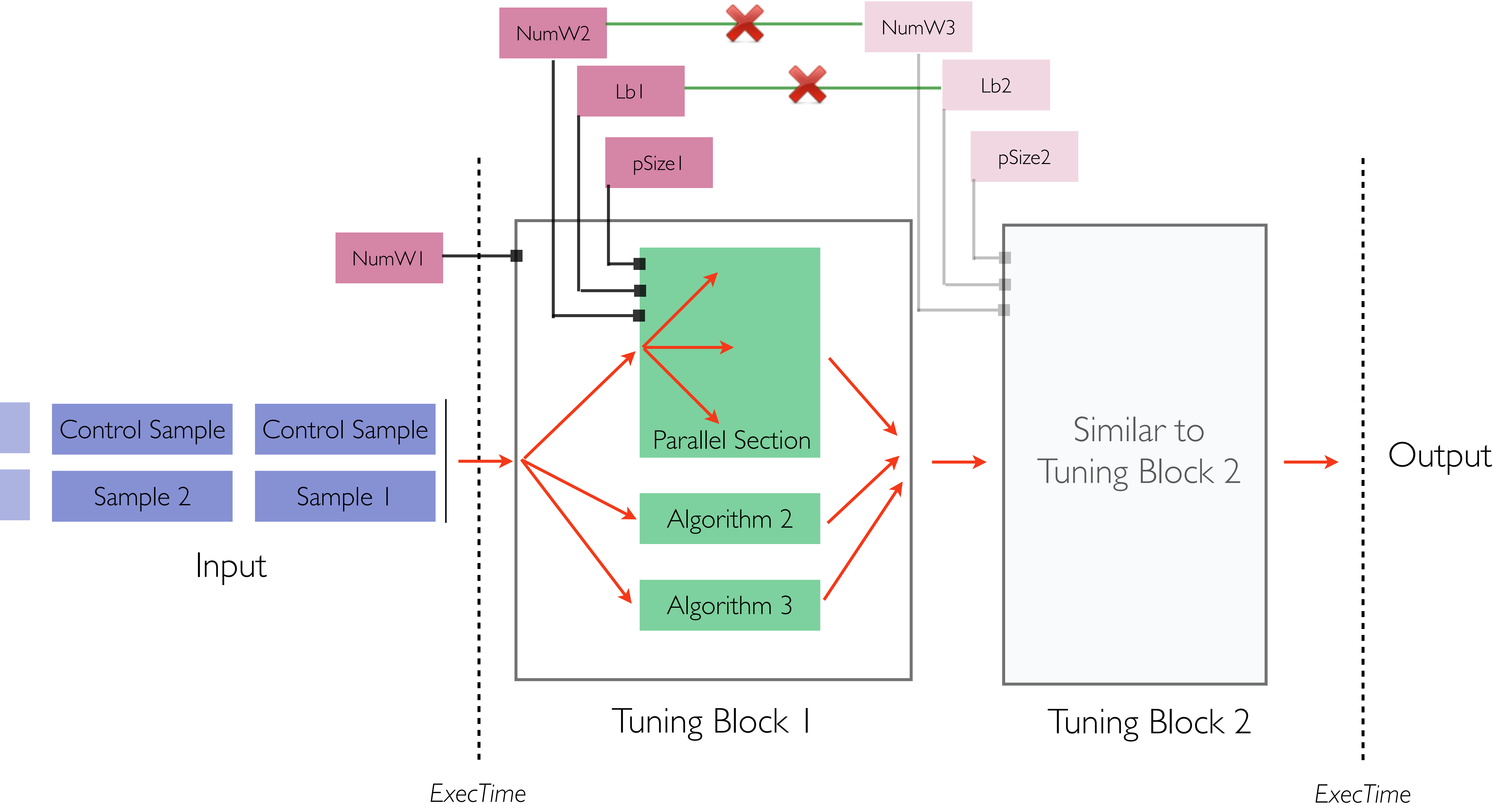
Metabolite ID



Metabolite ID



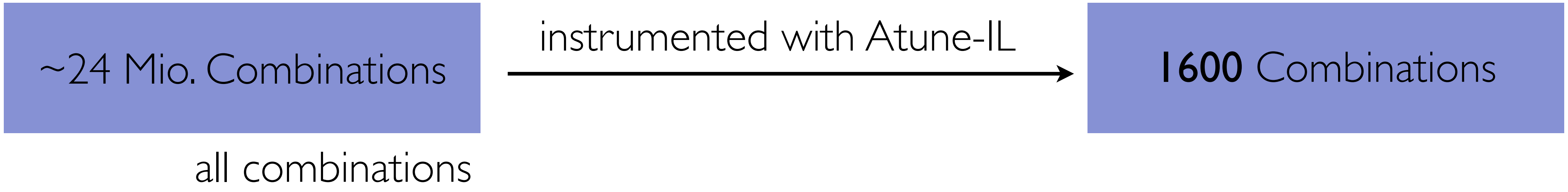
Metabolite ID



http://www.iconfinder.com/icondetails/11746/32/_icon

Case Study: Results I

- Search space reduction



~24 Mio. Combinations

all combinations

instrumented with Atune-IL

1600 Combinations

- Difference in execution time
  - 45% between the best and the worst configuration (8 core machine)

Case Study: Results 2

- Implementation effort

747 LOC

manually implemented

25 LOC

used Atune-IL

Related Work

- POET[1]
  - independent of application domain / host language
  - optimization on source code level
- XLanguage
  - #pragma approach
  - C / C++ code transformations
  - loop unrolling

[1] Parameterized Optimizing for Empirical Tuning

Pros and Cons

- ✓ Drastical search space reduction
- ✓ Host-language independent
- ✓ Independent of application domain
- ✓ Portability, maintenance

Pros and Cons

- ✓ Drastical search space reduction
- ✓ Host-language independent
- ✓ Independent of application domain
- ✓ Portability, maintenance

- Nothing available on the web
- Portability, maintenance
- Paper is incomplete / wrong
  - WEIGHT not specified
  - mixed START/DEFAULT
- LOC: hardly depends on programming style
  - template files ignored
- Section 6.4 „Results“ is weak