



UCSD

Bridging the Parallelization Gap: Automating Parallelism Discovery and Planning

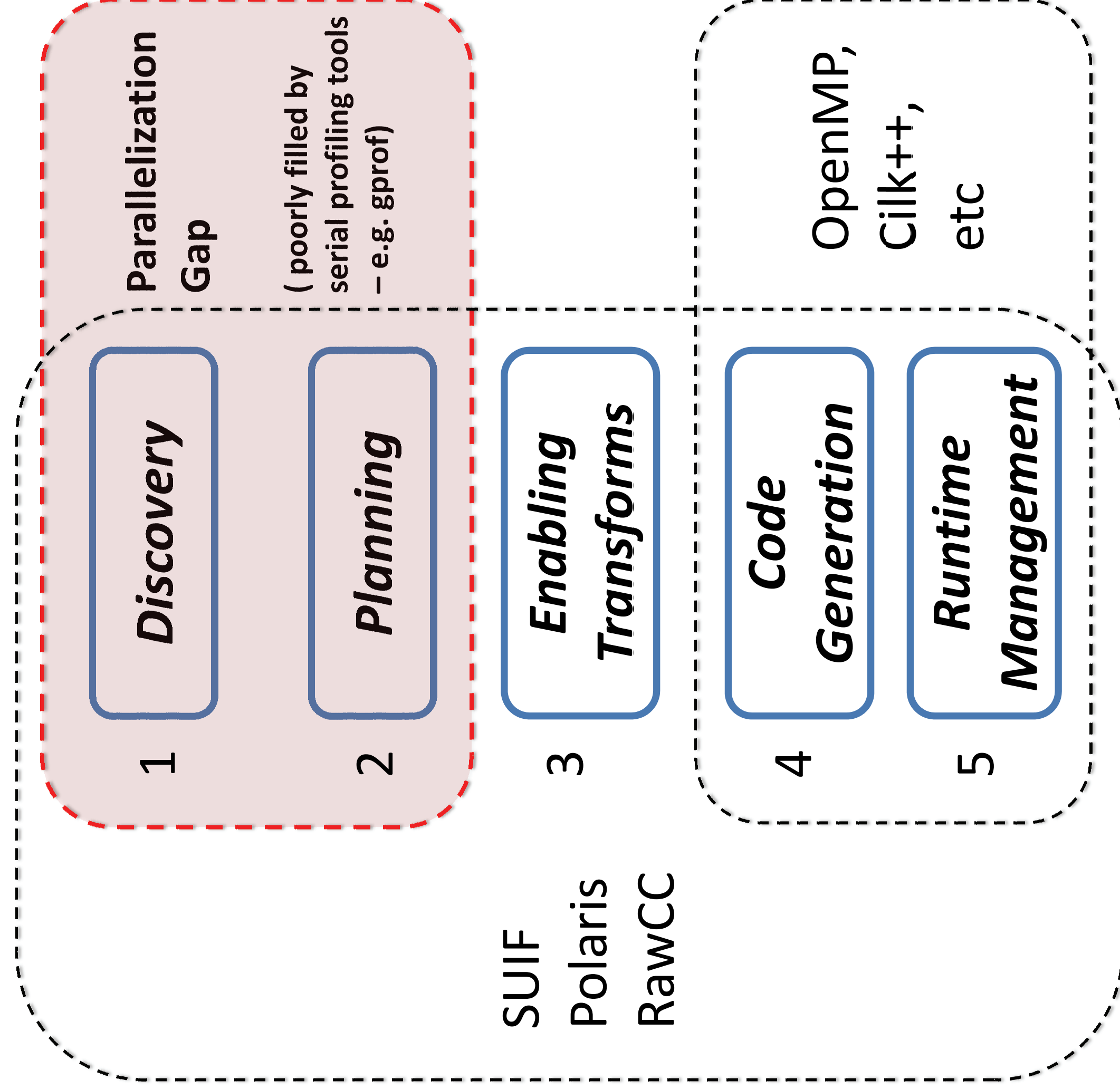
Saturnino Garcia, Donghwan Jeon, Chris Louie, Sravanthi Kota Venkata, and Michael Bedford Taylor

Computer Science and Engineering Department

University of California, San Diego

The Problem

Parallelization gap created by lack of parallelism discovery / planning tools



A taxonomy of parallelization

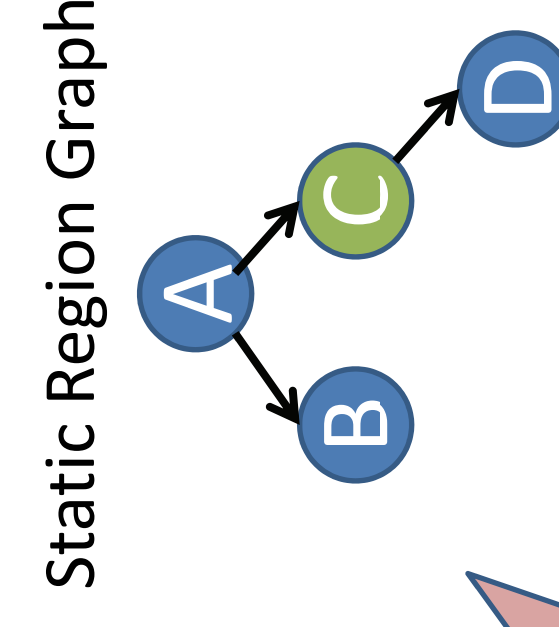
Shortcomings of Current Tools

Problem 1: High work coverage does not correlate with **parallelizability**

% cumulative time	seconds	self seconds	calls	ms/call	self ms/call	total ms/call	name
33.34	0.02	0.02	7208	0.00	0.00	0.00	open
16.67	0.03	0.01	244	0.04	0.12	0.12	offsettime
16.67	0.04	0.01	8	1.25	1.25	1.25	memccpy
16.67	0.05	0.01	7	1.43	1.43	1.43	write

A sample gprof output

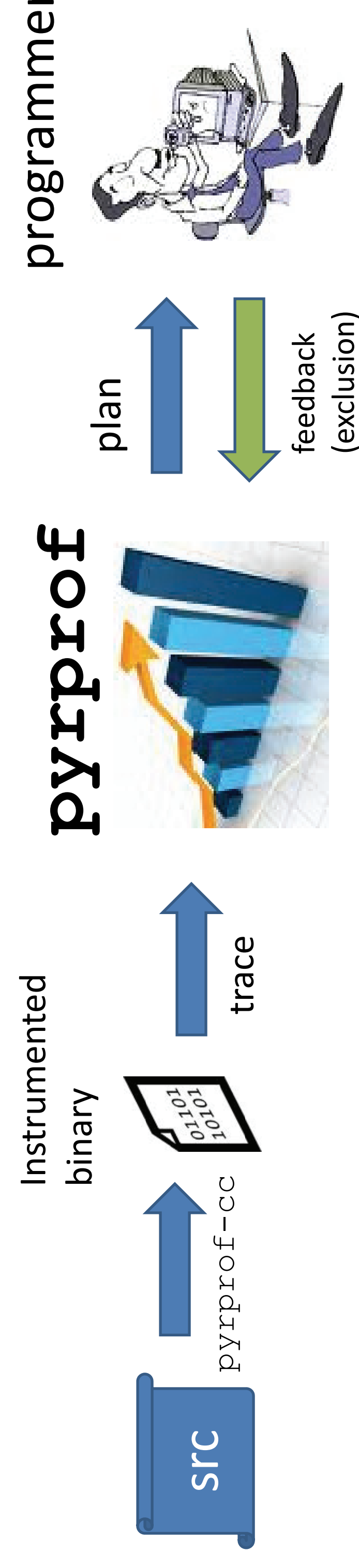
Problem 2: In planning, **region structure** and **parallelization status** must be considered



Parallelizing region D would not be profitable when C is already parallelized

Bridging the Gap - pyrprof

1. Follow time-tested **gprof usage model**
2. Profile not only **work** but also **parallelism**
3. Leverage **region structure** and **parallelization status**



```
$> pyrprof mpeg.trace -exclude=exclude.txt -n 5
The following regions will be excluded from recommendations: D, E
```

Rank	ID	Cum	Incr	File	Lines	Function	Type
1	A	3.14	3.14	motion.c	208	220 ptmotion estimation loop	loop
2	B	4.40	1.40	motion.c	211	220 ptmotion estimation loop	loop
3	G	5.50	1.25	transform.c	176	233 pttransform loop	loop
4	H	7.17	1.30	transform.c	249	305 pttransform loop	loop
5	C	9.60	1.34	putpic.c	376	612 ptputpic loop	loop

Cum. GNTES = GNTES compared to the serial version
Incr. GNTES = GNTES compared to the previous step

Case Study – mpeg encoder (ALPBench)

Iteration	Excluded Regions	Top 5 Regions	Confirmed Parallelizable	Action
1	{}	{A, E, B, D, I}	{A}	exclude E
2	{E}	{A, B, D, G, H}	{A, B}	exclude D
3	{D, E}	{A, B, G, H, C}	{A, B, G, H, C}	done

Planning Effectiveness Comparison

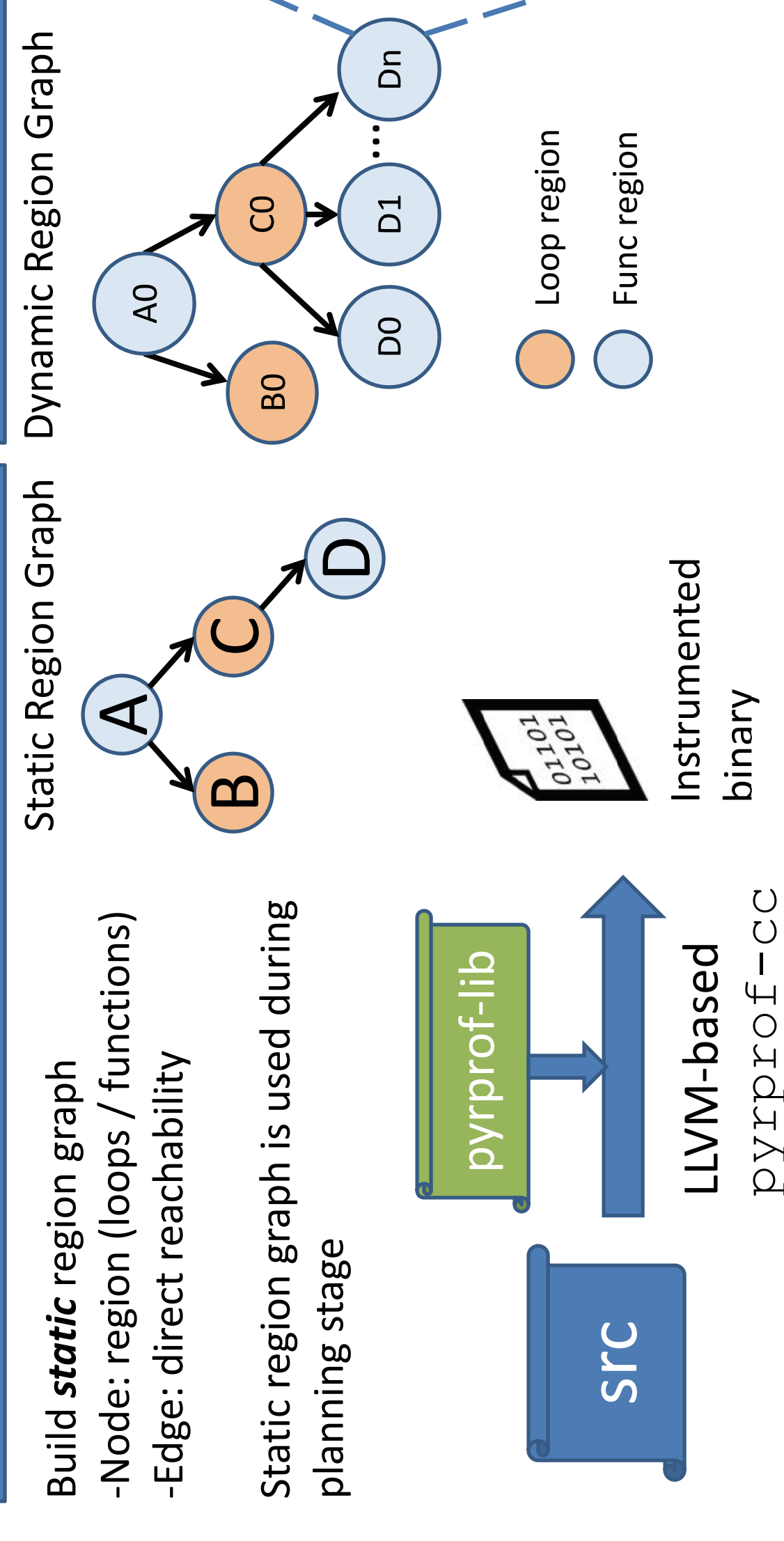
Region	gprof	pyrprof Initial	pyrprof interactive
A	7	1	1
B	8	3	2
G	27	41	3
H	31	29	4
C	35	35	5
F	42	13	6
D	3	4	excluded
E	5	2	excluded
Exploited Avg	28.4	23.8	3.8

Programmer examines region E and finds out it is not easily parallelizable

- Exploited region in parallel version of ALP
- Not exploited in ALP, but parallelizable
- Difficult to parallelize
- Rank: position in which region appears in tool output
- Average rank of 5 exploited regions in ALP

Discovery Stage

Phase 1: Region ID and Instrumentation



Instruments source code with calls to pyrprof profiling library functions

- LLVM used to inline and highly optimize instrumented code

For each **dynamic** region, build control / data flow graph (CDFG)

- Critical Path: longest path through CDFG
- Work: cost of operations in a region

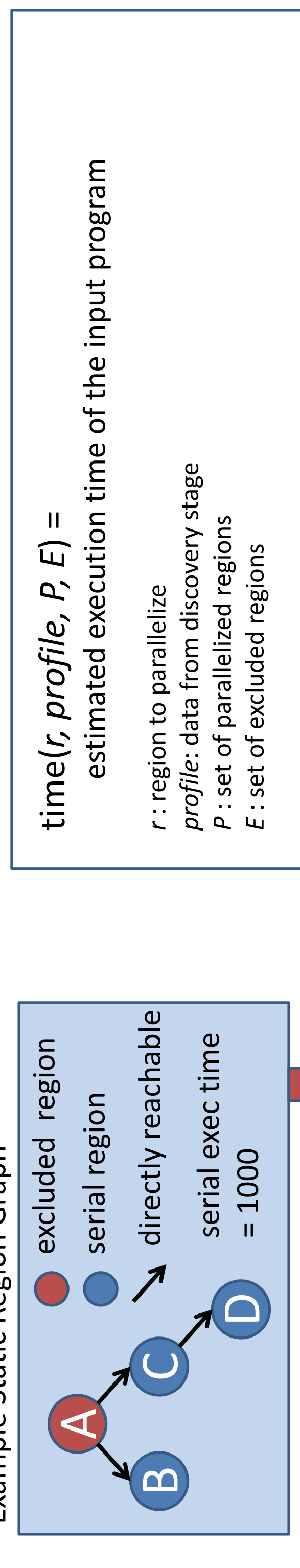
Parallelism = Work / Critical Path

Redundant profile data compressed using a dictionary-based technique

Planning Stage

Goal: provide an ordered list of regions ranked by the impact of their parallelization

Execution Time Estimation Model



$time(r, profile, P, E) =$ estimated execution time of the input program

r : region to parallelize
profile: data from discovery stage
P : set of parallelized regions
E : set of excluded regions

Estimated Times for Potential Parallelizations

Region	Step 1	Step 2	Step 3
B	800	400	
C	600		
D	700	590	390

$P = \{ \}$ $P = \{ C \}$ $P = \{ B, C \}$

Proposed Algorithm: Iteratively select the region that minimizes the total execution time

Parallelization of C reduces the benefit of parallelizing D

Rank	Region	time	Cum. GNTES	Incr. GNTES
0	serial	1000	1.0	1.0
1	C	600	1.66	1.66
2	B	400	2.5	1.5
3	D	390	2.56	1.03

Cum. GNTES = GNTES compared to the serial version
Incr. GNTES = GNTES compared to the previous step

Availability

pyrprof is available for free download at:

<http://parallel.ucsd.edu/pyrprof>