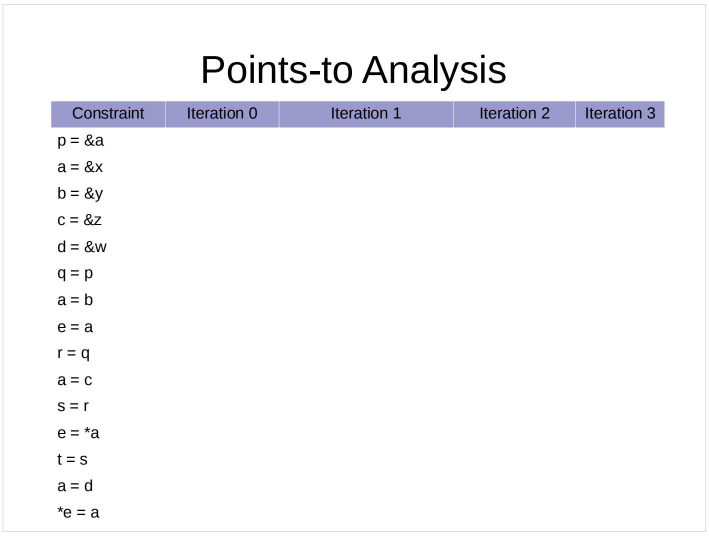
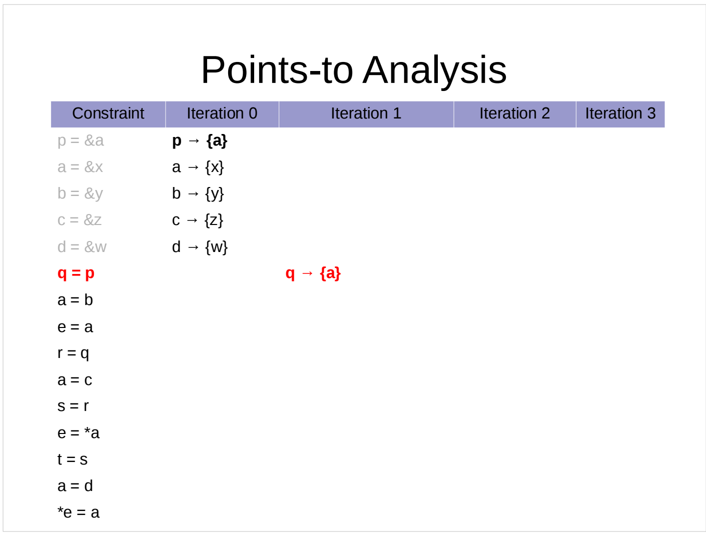


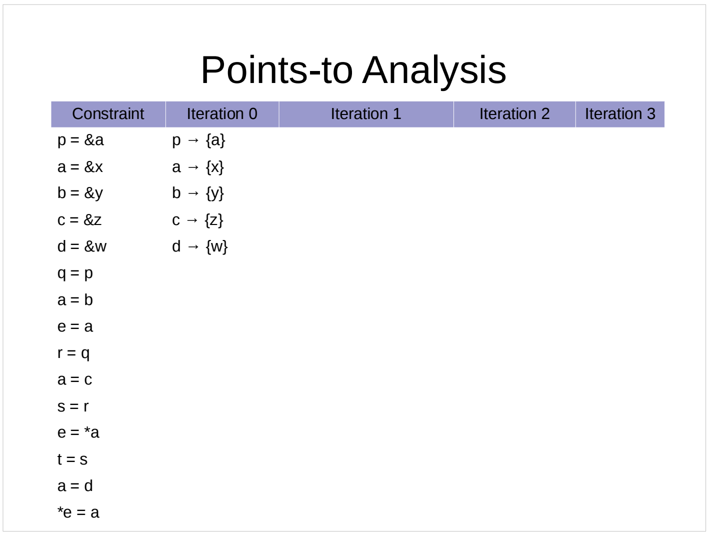
Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a				
a = &x				
b = &y				
c = &z				
d = &w				
q = p				
a = b				
e = a				
r = q				
a = c				
s = r				
e = *a				
t = s				
a = d				
*e = a				



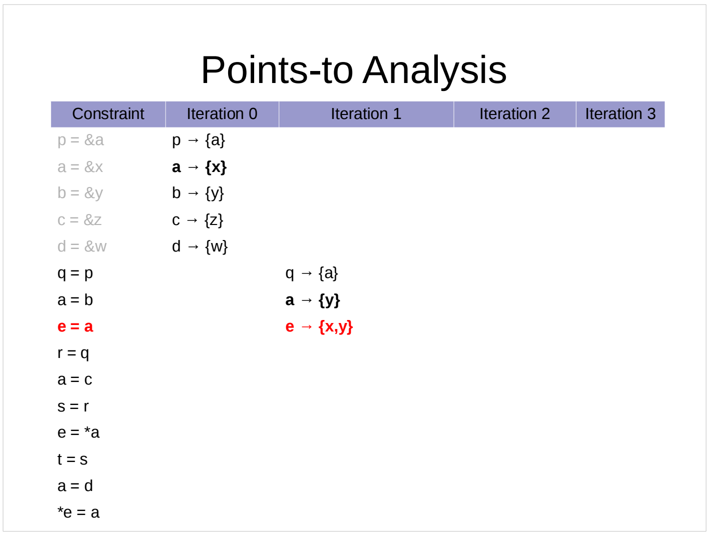
Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a	**p → {a}**			
a = &x	a → {x}			
b = &y	b → {y}			
c = &z	c → {z}			
d = &w	d → {w}			
q = p		**q → {a}**		
a = b				
e = a				
r = q				
a = c				
s = r				
e = *a				
t = s				
a = d				
*e = a				



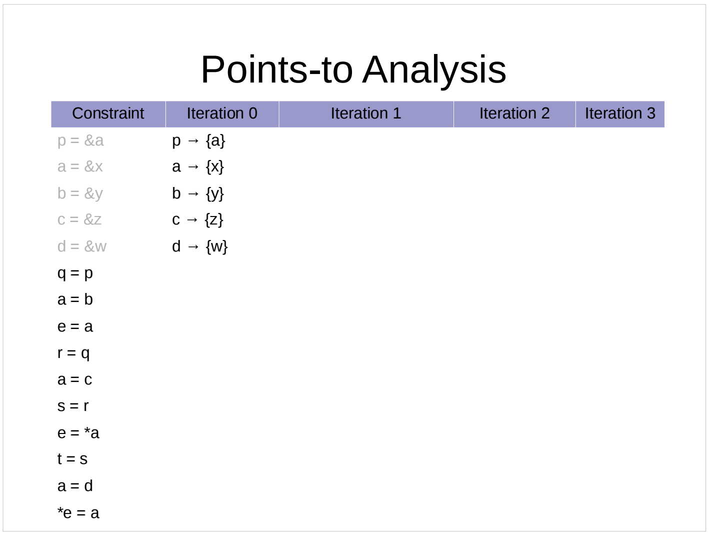
Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a	p → {a}			
a = &x	a → {x}			
b = &y	b → {y}			
c = &z	c → {z}			
d = &w	d → {w}			
q = p				
a = b				
e = a				
r = q				
a = c				
s = r				
e = *a				
t = s				
a = d				
*e = a				



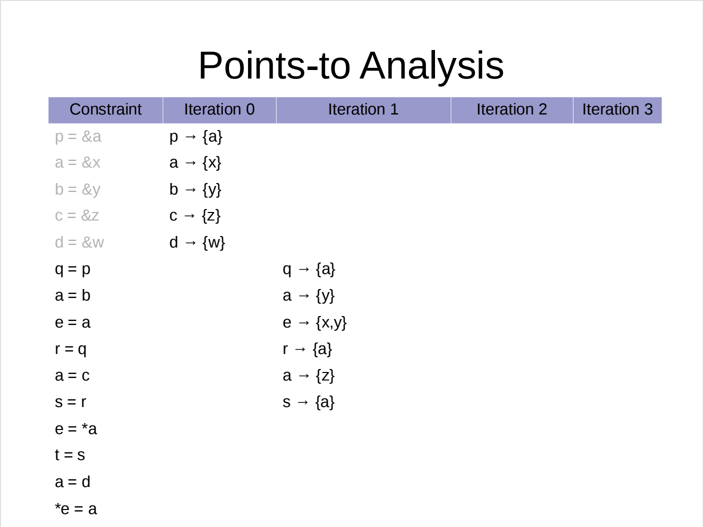
Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a	p → {a}			
a = &x	**a → {x}**			
b = &y	b → {y}			
c = &z	c → {z}			
d = &w	d → {w}			
q = p		q → {a}		
a = b		**a → {y}**		
e = a		**e → {x,y}**		
r = q				
a = c				
s = r				
e = *a				
t = s				
a = d				
*e = a				



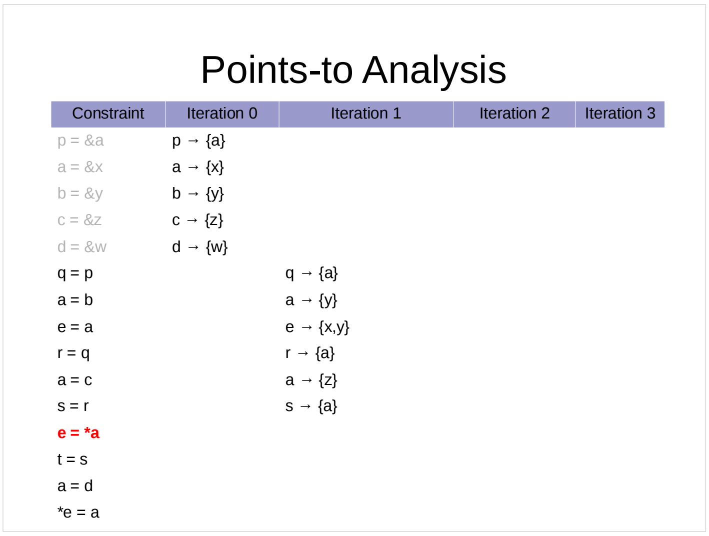
Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a	p → {a}			
a = &x	a → {x}			
b = &y	b → {y}			
c = &z	c → {z}			
d = &w	d → {w}			
q = p				
a = b				
e = a				
r = q				
a = c				
s = r				
e = *a				
t = s				
a = d				
*e = a				



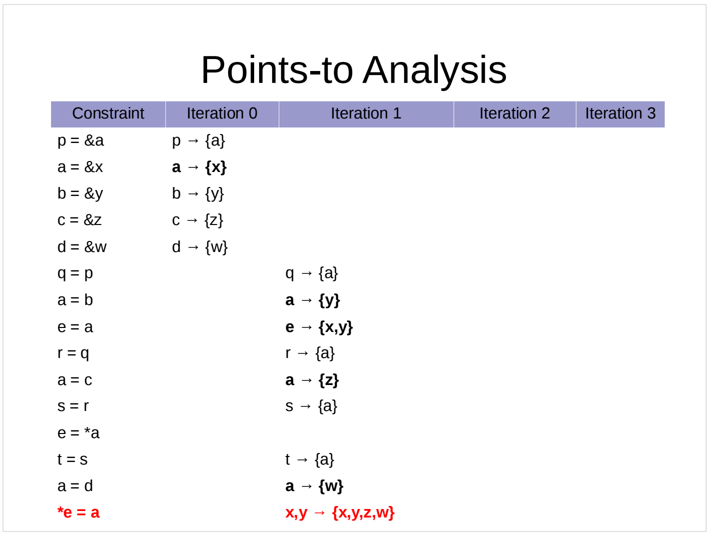
Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a	p → {a}			
a = &x	a → {x}			
b = &y	b → {y}			
c = &z	c → {z}			
d = &w	d → {w}			
q = p		q → {a}		
a = b		a → {y}		
e = a		e → {x,y}		
r = q		r → {a}		
a = c		a → {z}		
s = r		s → {a}		
e = *a				
t = s				
a = d				
*e = a				



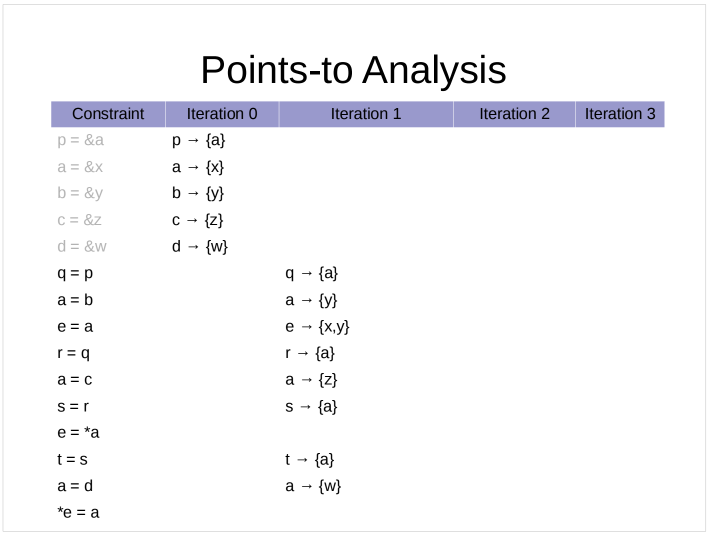
Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a	p → {a}			
a = &x	a → {x}			
b = &y	b → {y}			
c = &z	c → {z}			
d = &w	d → {w}			
q = p		q → {a}		
a = b		a → {y}		
e = a		e → {x,y}		
r = q		r → {a}		
a = c		a → {z}		
s = r		s → {a}		
e = *a				
t = s				
a = d				
*e = a				



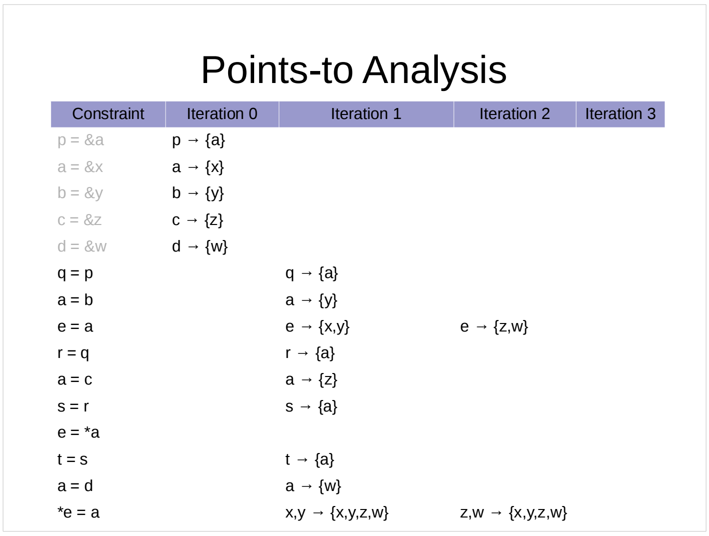
Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a	p → {a}			
a = &x	**a → {x}**			
b = &y	b → {y}			
c = &z	c → {z}			
d = &w	d → {w}			
q = p		q → {a}		
a = b		**a → {y}**		
e = a		**e → {x,y}**		
r = q		r → {a}		
a = c		**a → {z}**		
s = r		s → {a}		
e = *a				
t = s		t → {a}		
a = d		**a → {w}**		
***e = a**		**x,y → {x,y,z,w}**		



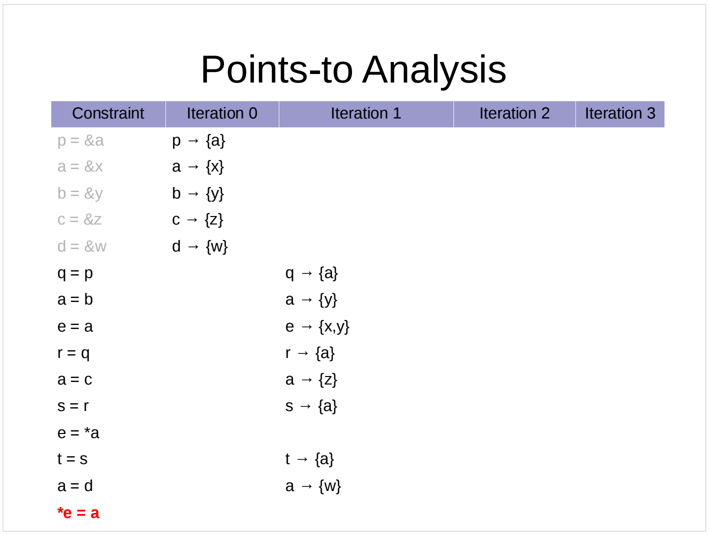
Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a	p → {a}			
a = &x	a → {x}			
b = &y	b → {y}			
c = &z	c → {z}			
d = &w	d → {w}			
q = p		q → {a}		
a = b		a → {y}		
e = a		e → {x,y}		
r = q		r → {a}		
a = c		a → {z}		
s = r		s → {a}		
e = *a				
t = s		t → {a}		
a = d		a → {w}		
*e = a				



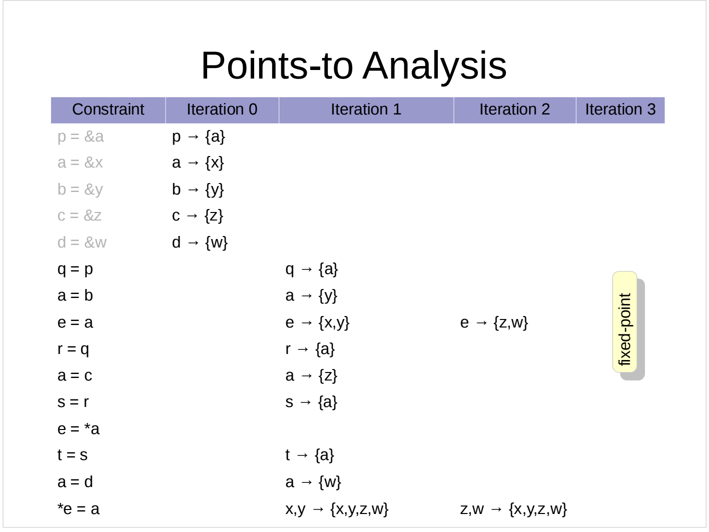
Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a	p → {a}			
a = &x	a → {x}			
b = &y	b → {y}			
c = &z	c → {z}			
d = &w	d → {w}			
q = p		q → {a}		
a = b		a → {y}		
e = a		e → {x,y}	e → {z,w}	
r = q		r → {a}		
a = c		a → {z}		
s = r		s → {a}		
e = *a				
t = s		t → {a}		
a = d		a → {w}		
*e = a		x,y → {x,y,z,w}	z,w → {x,y,z,w}	



Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a	p → {a}			
a = &x	a → {x}			
b = &y	b → {y}			
c = &z	c → {z}			
d = &w	d → {w}			
q = p		q → {a}		
a = b		a → {y}		
e = a		e → {x,y}		
r = q		r → {a}		
a = c		a → {z}		
s = r		s → {a}		
e = *a				
t = s		t → {a}		
a = d		a → {w}		
***e = a**				



Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a	p → {a}			
a = &x	a → {x}			
b = &y	b → {y}			
c = &z	c → {z}			
d = &w	d → {w}			
q = p		q → {a}		
a = b		a → {y}		
e = a		e → {x,y}	e → {z,w}	
r = q		r → {a}		
a = c		a → {z}		
s = r		s → {a}		
e = *a				
t = s		t → {a}		
a = d		a → {w}		
*e = a		x,y → {x,y,z,w}	z,w → {x,y,z,w}	

fixed-point

Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a	p → {a}			
a = &x	a → {x}			
b = &y	b → {y}			
c = &z	c → {z}			
d = &w	d → {w}			
q = p		q → {a}		
a = b		a → {y}		
e = a		e → {x,y}	e → {z,w}	
r = q		r → {a}		
a = c		a → {z}		
s = r		s → {a}		
e = *a				
t = s		t → {a}		
a = d		a → {w}		
*e = a		x,y → {x,y,z,w}	z,w → {x,y,z,w}	

13 steps

fixed-point

Parallel Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a				
a = &x				
b = &y				
c = &z				
d = &w				
q = p				
a = b				
e = a				
r = q				
a = c				
s = r				
e = *a				
t = s				
a = d				
*e = a				

Outline

- ✓ Introduction — Basics! What people already know.
- Parallel Points-to Analysis — Cryptic! What people fail to know.
  - Naïve method
  - Replication-based approach
- Optimizations — Details! What people need not know.
- Results — Horror! What people should not know.

Parallel Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a				
a = &x				
b = &y				
c = &z				
d = &w				
q = p				
a = b				
e = a				
r = q				
a = c				
s = r				
e = *a				
t = s				
a = d				
*e = a				

Parallel Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a				
a = &x				
b = &y				
c = &z				
d = &w				
q = p				
a = b				
e = a				
r = q				
a = c				
s = r				
e = *a				
t = s				
a = d				
*e = a				

Conflicting Constraints

Two constraints conflict if at least one of them writes to a common variable.

Points-to constraints		Read-Set	Write-Set
p = &q	address-of	{ }	{p}
p = q	copy	{q}	{p}
p = *q	load	{q} U {x: q → {x}}	{p}
*p = q	store	{q, p}	{x: p → {x}}

Parallel Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a				
a = &x				
b = &y				
c = &z				
d = &w				
q = p				
a = b				
e = a				
r = q				
a = c				
s = r				
e = *a				
t = s				
a = d				
*e = a				

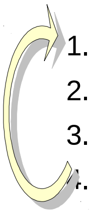
Parallel Points-to Analysis

1. Find conflicting constraints.
2. Schedule constraints.
3. Analyze in parallel.
4. Update points-to information.

Parallel Points-to Analysis

Thread1	Thread2
a = b	q = p
a = c	r = q
a = d	s = r
e = *a	t = s
e = a	
*e = a	

Parallel Points-to Analysis



1. Find conflicting constraints.
2. Schedule constraints.
3. Analyze in parallel.
4. Update points-to information.

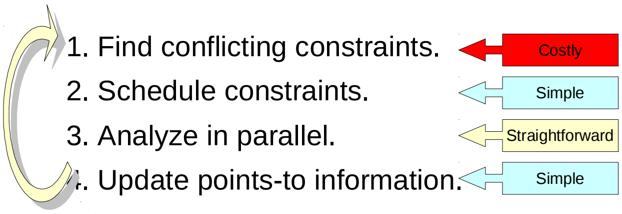
Parallel Points-to Analysis

9 steps

Thread1	Thread2
a = b	q = p
a = c	r = q
a = d	s = r
e = *a	t = s
e = a	
*e = a	

Sequential: 13 steps.
Even if the analysis is provided with 8 cores, the parallel analysis still requires 9 steps.

Parallel Points-to Analysis



1. Find conflicting constraints. — Costly
2. Schedule constraints. — Simple
3. Analyze in parallel. — Straightforward
4. Update points-to information. — Simple

Replication-based Parallel Points-to Analysis

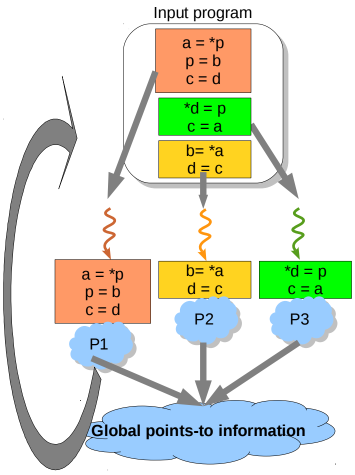
1. Find conflicting constraints.
2. Schedule constraints.
3. Analyze in parallel.
4. Update points-to information.

Replication-based Parallel Points-to Analysis

1. ~~Find conflicting constraints.~~
2. Schedule constraints.
3. Analyze in parallel.
   - Initial reads from the master copy.
   - Writes to local replica.
4. Update points-to information.
   - Merge local replicas with the master copy.

Replication-based Parallel Points-to Analysis

1. ~~Find conflicting constraints.~~
2. Schedule constraints.
3. Analyze in parallel.
4. Update points-to information.



Arbitrary Partition
(points-to constraints)

Parallel Execution
(replicated points-to)

Merge
(global points-to)

Replication-based Parallel Points-to Analysis

1. ~~Find conflicting constraints.~~
2. Schedule constraints. ← Simple
3. Analyze in parallel. ← Straightforward
4. Update points-to information. ← Costly

Why Replication Works

1. Monotonically increasing computation.
   - Points-to sets never shrink.
2. Unordered algorithm.
   - Constraints can be processed in any order.

For instance, a naïve replication doesn't work for flow-sensitive analysis.

Parallel Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
p = &a				
a = &x				
b = &y				
c = &z				
d = &w				
q = p				
a = b				
e = a				
r = q				
a = c				
s = r				
e = *a				
t = s				
a = d				
*e = a				

Replication-based Parallel Points-to Analysis: **3 Threads**

T	Stmt	Itr 1	Merge 1	Itr 2	Merge 2	Itr 3	Merge 3
1	q = p	q' → {a}					
	a = b	a' → {y}					
	e = a	e' → {x, y}					
	r = q	r' → {a}					
2	a = c	a'' → {z}					
	s = r						
	e = *a						
3	t = s						
	a = d	a''' → {w}					
	*e = a						

Parallel Points-to Analysis

Constraint	Iteration 0	Iteration 1	Iteration 2	Iteration 3
q = p				
a = b				
e = a				
r = q				
a = c				
s = r				
e = *a				
t = s				
a = d				
*e = a				

Replication-based Parallel Points-to Analysis: **3 Threads**

T	Stmt	Itr 1	Merge 1	Itr 2	Merge 2	Itr 3	Merge 3
1	q = p	q' → {a}					
	a = b	a' → {y}					
	e = a	e' → {x, y}					
	r = q	r' → {a}	a → {y, z, w}				
2	a = c	a'' → {z}	e → {x, y}				
	s = r		q, r → {a}				
	e = *a						
3	t = s						
	a = d	a''' → {w}					
	*e = a						

Replication-based Parallel Points-to Analysis: **3 Threads**

T	Stmt	Itr 1	Merge 1	Itr 2	Merge 2	Itr 3	Merge 3
1	q = p						
	a = b						
	e = a						
	r = q						
2	a = c						
	s = r						
	e = *a						
3	t = s						
	a = d						
	*e = a						

Replication-based Parallel Points-to Analysis: **3 Threads**

T	Stmt	Itr 1	Merge 1	Itr 2	Merge 2	Itr 3	Merge 3
1	q = p	q' → {a}					
	a = b	a' → {y}					
	e = a	e' → {x, y}		e' → {x,y,z,w}			
	r = q	r' → {a}	a → {y, z, w}		e → {y,z,w}		z,w → {x,y,z,w}
2	a = c	a'' → {z}	e → {x, y}		x → {x,y,z,w}		t → {a}
	s = r		q, r → {a}	s' → {a}	s → {a}		
	e = *a						
3	t = s					t' → {a}	
	a = d	a''' → {w}					
	*e = a			x',y' → {x,y,z,w}		z',w' → {x,y,z,w}	

Replication-based Parallel Points-to Analysis: **3 Threads**

T	Stmt	Itr 1	Merge 1	Itr 2	Merge 2	Itr 3	Merge 3
1	q = p	q' → {a}					
	a = b	a' → {y}					
	e = a	e' → {x, y}		e' → {x,y,z,w}			
	r = q	r' → {a}	a → {y, z, w}		e → {y,z,w}		z,w → {x,y,z,w}
2	a = c	a'' → {z}	e → {x, y}		x → {x,y,z,w}		t → {a}
	s = r		q, r → {a}	s' → {a}	s → {a}		
	e = *a						
3	t = s					t' → {a}	
	a = d	a''' → {w}					
	*e = a			x',y' → {x,y,z,w}		z',w' → {x,y,z,w}	

12 steps

Sequential: 13 steps, Parallel: 9 steps.

Replication-based Parallel Points-to Analysis: **4 Threads**

T	Stmt	Itr 1	Merge 1	Itr 2	Merge 2	Itr 3	Merge 3
1	q = p	q' → {a}					
	e = a	e' → {x}		e' → {y,z,w}			
2	r = q			r' → {a}			
	a = c	a'' → {z}	a → {y, z, w}		e → {y,z,w}		y,z,w → {x,y,z,w}
3	e = *a		e → {x}		x → {x,y,z,w}		s,t → {a}
	s = r		q → {a}		r → {a}		
	t = s					s',t' → {a}	
	a = d	a''' → {w}					
4	*e = a			x' → {x,y,z,w}		y',z',w' → {x,y,z,w}	
	a = b	a' → {y}					

Naïve vs. Replication-based

Pros	Cons
No merging	Costly merging
Lesser iterations	More iterations
General purpose	Monotonic, unordered
Lower memory requirement	Higher memory requirement

Cons	Pros
Costly conflict-detection	No conflict-detection
Limited parallelism	Adaptive parallelism
Unbalanced load	Better load-balancing
Lower parallel performance	Better parallel performance

Replication-based Parallel Points-to Analysis: **4 Threads**

T	Stmt	Itr 1	Merge 1	Itr 2	Merge 2	Itr 3	Merge 3
1	q = p	q' → {a}					
	e = a	e' → {x}		e' → {y,z,w}			
2	r = q			r' → {a}			
	a = c	a'' → {z}	a → {y, z, w}		e → {y,z,w}		y,z,w → {x,y,z,w}
3	e = *a		e → {x}		x → {x,y,z,w}		s,t → {a}
	s = r		q → {a}		r → {a}		
	t = s					s',t' → {a}	
	a = d	a''' → {w}					
4	*e = a			x' → {x,y,z,w}		y',z',w' → {x,y,z,w}	
	a = b	a' → {y}					

9 steps

Sequential: 13 steps, Parallel: 9 steps.
Replication with 3 threads: 12 steps.

Replication-based Parallel Points-to Analysis: **4 Threads**

T	Stmt	Itr 1	Merge 1	Itr 2	Merge 2	Itr 3	Merge 3
1	q = p						
	e = a						
2	r = q						
	a = c						
3	e = *a						
	s = r						
	t = s						
	a = d						
4	*e = a						
	a = b						

Replication-based Parallel Points-to Analysis: **5 Threads**

T	Stmt	Itr 1	Merge 1	Itr 2	Merge 2	Itr 3	Merge 3
1	e = a	e' → {x}		e' → {y,z,w}			
2	q = p	q' → {a}					
3	r = q			r' → {a}			
	a = c	a'' → {z}	a → {y, z, w}		e → {y,z,w}		y,z,w → {x,y,z,w}
4	e = *a		e → {x}		x → {x,y,z,w}		s,t → {a}
	s = r		q → {a}		r → {a}		
	t = s					s',t' → {a}	
	a = d	a''' → {w}					
5	*e = a			x' → {x,y,z,w}		y',z',w' → {x,y,z,w}	
	a = b	a' → {y}					

Replication-based Parallel Points-to Analysis: **5 Threads**

T	Stmt	Itr 1	Merge 1	Itr 2	Merge 2	Itr 3	Merge 3
1	e = a	e' → {x}		e' → {y,z,w}			
2	q = p	q' → {a}					
3	r = q			r' → {a}			
	a = c	a'' → {z}	a → {y, z, w}		e → {y,z,w}		y,z,w → {x,y,z,w}
4	e = *a		e → {x}		x → {x,y,z,w}		s,t → {a}
	s = r		q → {a}		r → {a}		
	t = s					s',t' → {a}	
	a = d	a''' → {w}					
5	*e = a			x' → {x,y,z,w}		y',z',w' → {x,y,z,w}	
	a = b	a' → {y}					

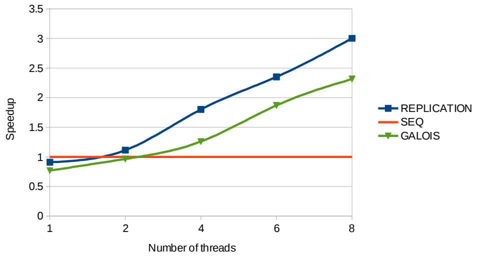
8 steps

Sequential: 13 steps, Parallel: 9 steps.
Replication with 3 threads: 12 steps.
Replication with 4 threads: 9 steps.

Optimizations

- Load Balancing
  - Orphan-and-Adopt approach.
  - *store* constraints are the culprits.
  - Trade-off between load-balancing and thread-communication.
- Parallel Online Cycle Elimination
  - Disjoint cycles can be collapsed in parallel.
- Reducing Replication Cost
  - Single writer.
  - Difference propagation.
  - Constraint affinity.
- Limited Scheduling

Results



Benchmarks: 16 SPEC 2K + 5 open-source