## Continuous Compilation for Aggressive and Adaptive Code Transformation



Bruce R. Childers

University of Pittsburgh Pittsburgh, Pennsylvania, USA <u>childers@cs.pitt.edu</u> http://www.cs.pitt.edu/coco

This work is a collaboration of many people! Sponsored by Next Generation Software, National Science Foundation.











































Benchmark		P	RE	LICM			
	ТР	СР	%Accacy	TP	СР	%Accacy	
gzip	48	43	89.58	45	38	84.44	
vpr	303	291	96.04	230	217	94.35	
mcf	51	44	86.27	52	43	82.69	
parser	293	210	87.87	75	68	90.67	
vortex	530	431	81.13	346	303	87.57	
bzip2	56	44	78.57	88	79	89.77	
twolf	475	433	91.12	345	306	88.70	

## PRE & LICM with Heuristic

	He	uristic-	driven F	PRE	Heuristic-driven LICM					
Benchmark	0	4	8	16	0	4	8	16		
gzip	3.50	3.75	3.78	4.10	2.90	3.29	5.40	3.27		
vpr	1.22	0.75	1.81	1.83	-0.40	-0.38	0.52	0.69		
mcf	2.37	2.35	2.31	2.22	2.50	2.62	2.58	2.47		
parser	1.25	1.50	1.70	1.35	2.55	2.86	1.99	2.23		
vortex	4.73	5.25	4.66	3.86	4.88	5.69	4.99	5.28		
bzip2	7.35	7.52	8.19	7.91	7.02	7.35	6.70	4.57		
twolf	1.07	0.88	1.14	0.02	0.52	0.38	2.14	1.91		
Best performance										





## Finding Good Optimization Orders

## • Planning with genetic algorithms

- Adaptively changing optimization order is effective
- Search for "best" optimization sequence
- Score solutions by executing the resulting program
- Typically, only small programs are considered
- Almagor [LCTES 2004), Kulkarni [PLDI 2004]
- Using FPO to improve planning performance
  - Use Almagor GA approach
  - Score solutions with models, avoiding need to run program
  - As good sequences (performance) as experimental scoring
  - Faster compilation times
  - Larger, more realistic programs

























