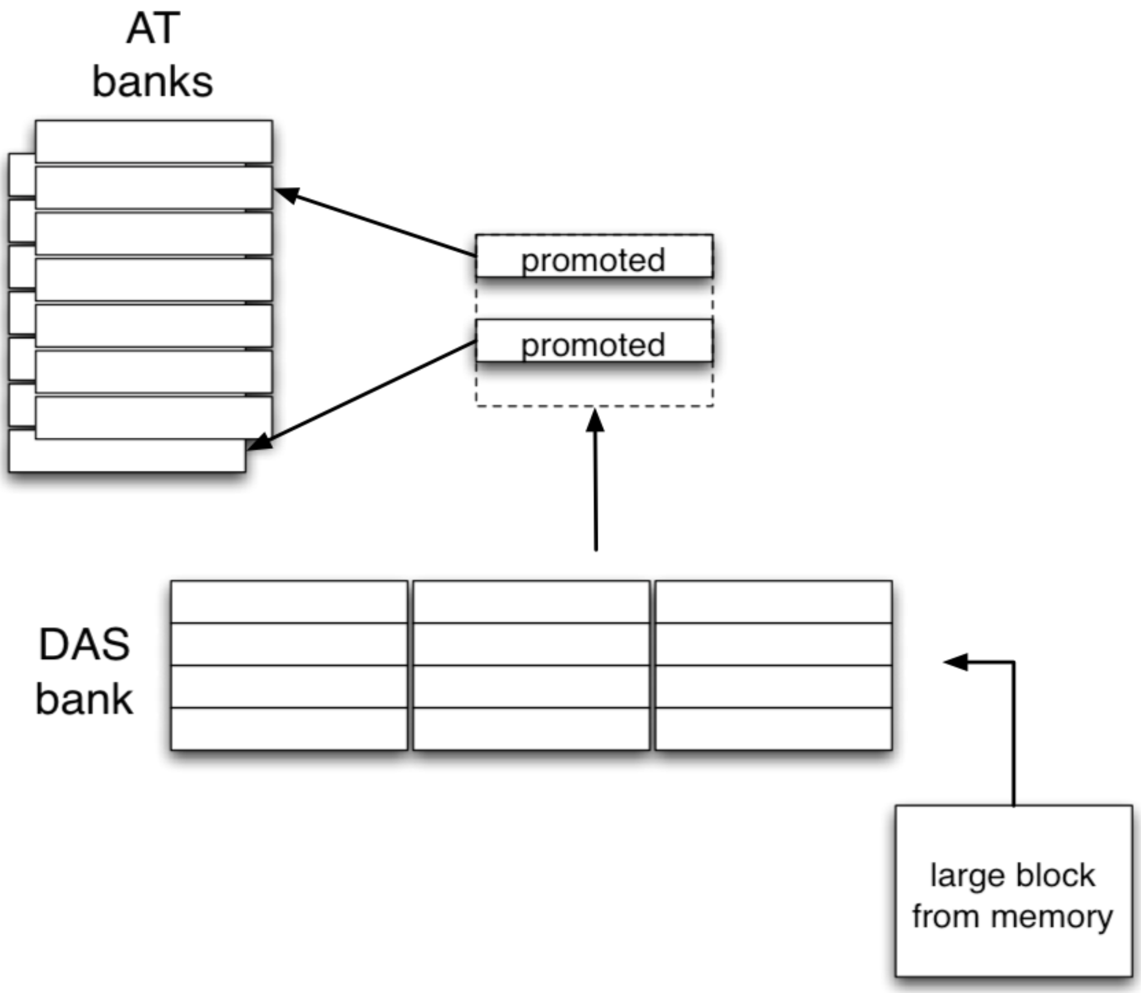
Applying
R-STAGE to DASAT

M Tyler Maxwell
University of Massachusetts Amherst

DASAT

- Dynamically Aggressive Spatial, Adaptive Temporal
- Exploits both spatial and temporal locality
  - Small blocks to exploit temporal locality
  - Large blocks to exploit spatial locality
- Heuristic-driven variable size prefetch
- Hit rates comparable to a conventional cache **4X** its size

DASAT Structure



Parameters of DASAT

n	# blocks AT
m	# large blocks DAS
wpb	# words per block
sbplb	# blocks per large block
b1	prediction bound 1
b2	prediction bound 2
b3	prediction bound 3
promo	promotion threshold
hitmax	max value for hit counter

Parameter Bounds

n	2^a	$2 \leq a \leq 17$
m	2^b	$2 \leq b \leq 8$
wpb	2^c	$0 \leq c \leq 6$
sbplb	2^d	$1 \leq d \leq 5$
b1	b_1	$b_1 = 0$
b2	b_2	$b_1 < b_2 < 15$
b3	b_3	$b_2 < b_3 < 20$
promo	p	$0 \leq p \leq 6$
hitmax	2^e	$0 \leq e \leq 3$

This space contains 7,487,690 points

Which Parameters are Best?

- Choose a point that gives best possible performance for (process, benchmark, miss penalty)
- Exhaustive search would take ~40,000 CPU years
- Goodness function (eAMAT) is a function of hit rate and DASAT speed

Computing eAMAT

$$eAMAT = \left[hitRate * \max(atTime, dasTime)\right] + \left[(1 - hitRate) * (atTime + missPenalty)\right]$$

$$hitRate = S(\vec{P})$$

$$atTime = C_{AT}(\vec{P})$$

$$dasTime = C_{DAS}(\vec{P})$$

$$missPenalty = k$$

Computing eAMAT (cont.)

- C_{AT} and C_{DAS} are computed offline by CACTI3.0
- S is computed by trace event simulation, so it is time-intensive
- Define two lengths of simulation
  - S_f : Full (4.6B refs, ~2 days)
  - S_p : Partial (500M refs, ~2 hours)

Standard Hill Climbing

$$\text{path } m_i: \quad \overbrace{S(\vec{P}_{starting})\cdots S\cdots S\cdots S(\vec{P}_{localopt})}^{q}$$

(where $\cdots$ is a search of i neighbors)

$$T_{HC} = m \times qi \times T(S)$$

Regression as a Heuristic

- We can substitute a regression curve for the hit rate surface (*much* faster)
- Need k source S_p points for generated curve
- Can do this in parallel using c CPUs
- Empirical results show $k > 150$ approximates DASAT's 9-space
- Hillclimb on regression, then perform simulation

Regression Hillclimbing

$$\text{path } m_i : \quad \overbrace{R(\vec{P}_{starting}) \cdots R \cdots R \cdots R(\vec{P}_{localopt})}^{q} \cdots S_p(\vec{P}_{localopt})$$

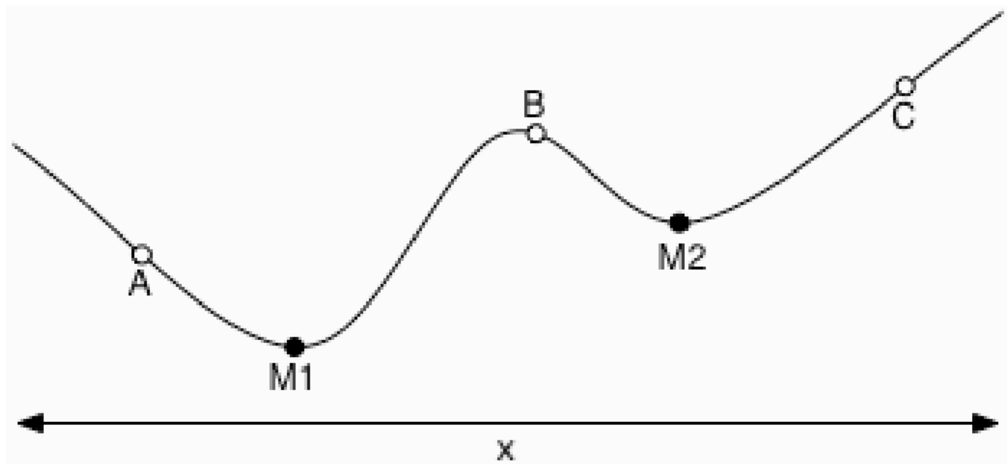
$$T_{REG-HC} = m \times T(S_p) + \frac{k \times T(S_p)}{c}$$

STAGE

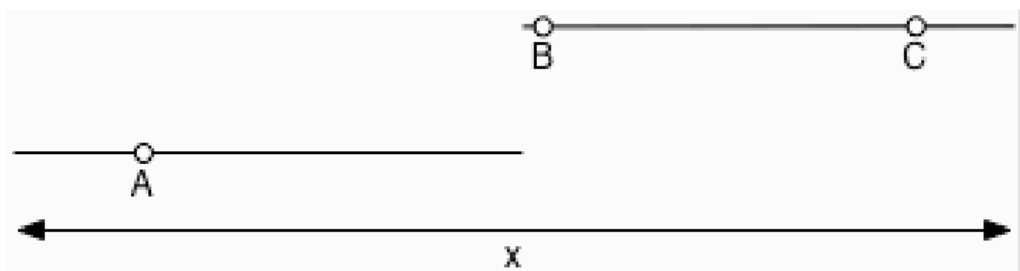
- Motivation: Increase starting point quality and thus decrease necessary m
- Train a feature space that predicts expected maximal goodness of starting at P_{start}
- STAGE works well if search space is patterned

A Picture of Feature Space

Normal Space



Feature Space



Applying STAGE

- Define architecture space to be the set of all possible P points
- Define feature space to be some projection P into V
- For every path m_i, train corresponding feature space points on arrived maximal goodness value
- To select a new starting point, hillclimb on feature space

R-STAGE

- Combines regression curve with *m*-reducing STAGE algorithm
- Provides 10-40% eAMAT improvements in several weeks
- More work can be done to further reduce the number of paths *m*

eAMAT Results (R-STAGE)

Benchmark	Base (ns)	Opt (ns)	%improv
apsi	1.7519	1.0433	40.4%
bzip2	1.2315	0.9792	20.5%
compress	1.0623	0.9208	13.3%
javac	1.0112	0.8771	13.3%
mpegaudio	1.0453	0.8729	16.5%
wupwise	1.1829	0.9294	21.4%