CAMA
Cache-Aware Memory Allocation for WCET Analysis

Jörg Herter  Jan Reineke  Reinhard Wilhelm

Department of Computer Science
Saarland University

ECRTS, July 2008
Current Situation

- WCET analyses strive for safe and precise bounds on execution times of programs
- Such analyses need to derive bounds on cache behavior
- Challenges:

```c
...  
int x = malloc(8);  
int y = malloc(4);  
...  
for (int i = 0; i < 100; i++) {  
    x->data = y->data + 2;  
...  
```  

- How long will `malloc` take?
  1. Allocation to cache sets unknown!
  2. Effects of calls to `malloc` on cache?
  3. Is the access to `y` a cache hit?
Self-Evident Question

predictable cache behavior + dynamic memory allocation
A Possible Solution?

Step 1

Replace the memory allocator by a predictable one that

- can explicitly allocate to a given cache set
  - by adding new argument: `malloc(size, cache set)`
- causes minor, predictable cache pollution
  - by using a segregated-lists allocator
- has constant execution times
  - by using a segregated-lists allocator
A Possible Solution?

Step 2
Compute shapes of data structures and their mapping to cache sets; e.g.:

- at most 2 lines per cache set affected by list traversal
- bounded information loss about the cache
- able to infer cache hits on further traversals
Thanks!

Thank you!