Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Forgetful: Eliminating Small, Transient Memory Allocations

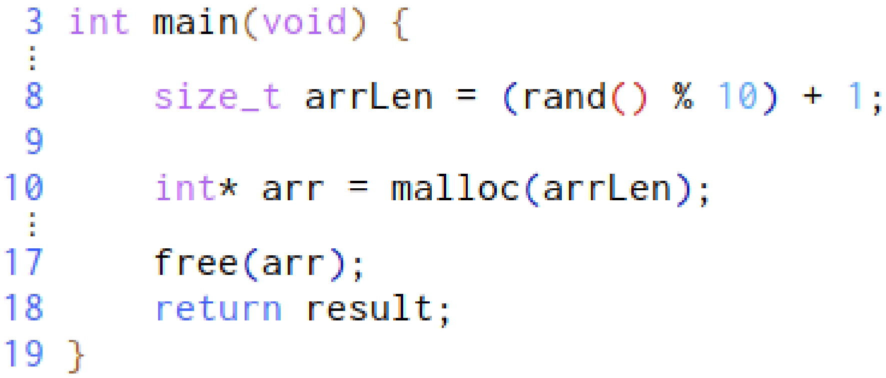
Dário Tavares Antunes
Dr. David M. Abrahamson

Date 04/05/17

Background

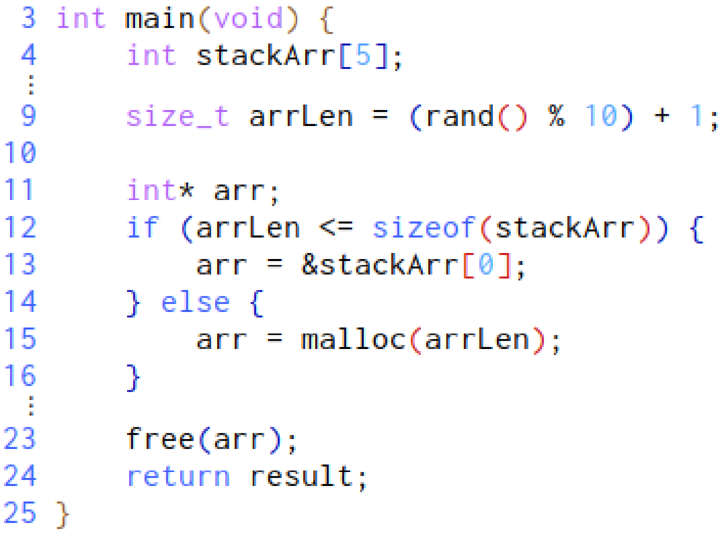
Inspired by changes to *cURL*

- Lots of malloc calls removed and “*7.54.1-DEV repeatedly performed **30% faster***”
- Generalise, make it easier to find opportunities to do the same
- Benchmark it



```c
int main(void) {
⋮
    size_t arrLen = (rand() % 10) + 1;

    int* arr = malloc(arrLen);
⋮
    free(arr);
    return result;
}
```



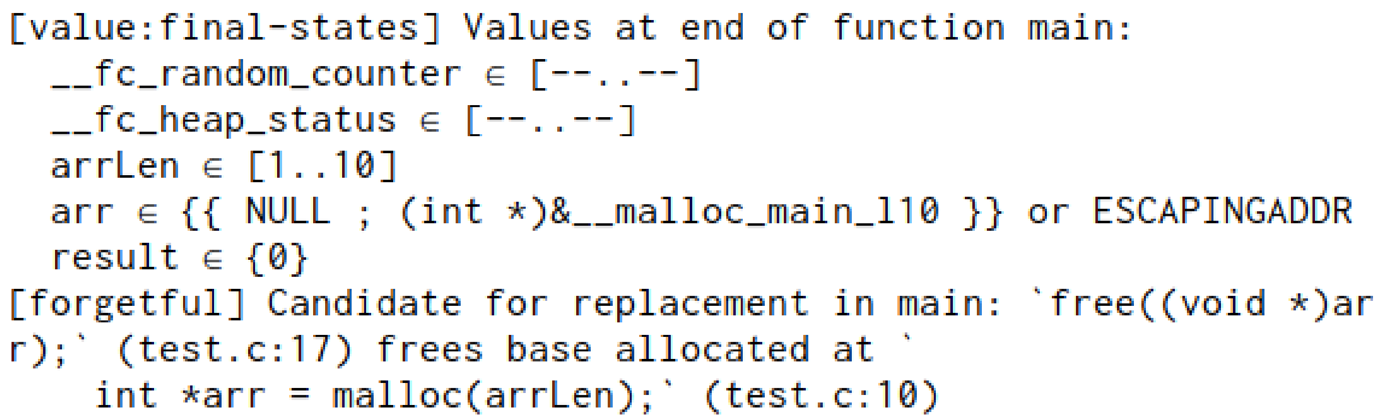
```c
int main(void) {
    int stackArr[5];
⋮
    size_t arrLen = (rand() % 10) + 1;

    int* arr;
    if (arrLen <= sizeof(stackArr)) {
        arr = &stackArr[0];
    } else {
        arr = malloc(arrLen);
    }
⋮
    free(arr);
    return result;
}
```

Implementation

Static Analysis Plugin

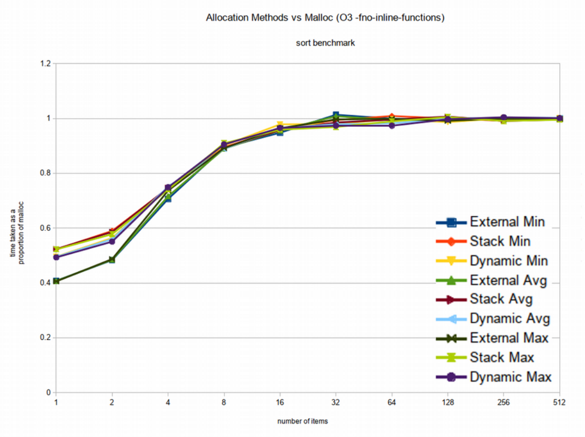
- Created the *Forgetful* plugin to track memory issues
- Finds short-lived allocations to replace with stack allocations
- Lets the user decide what’s worth changing

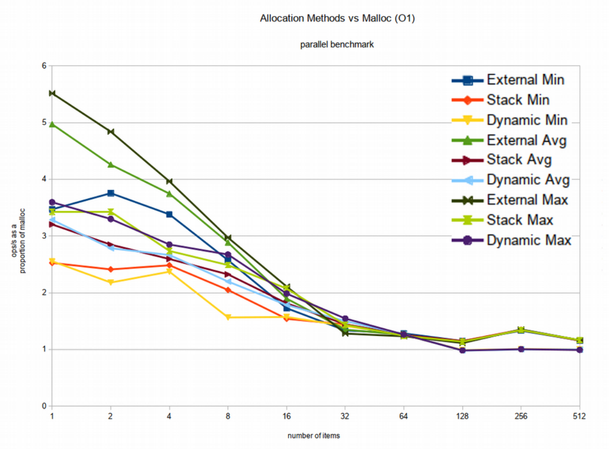


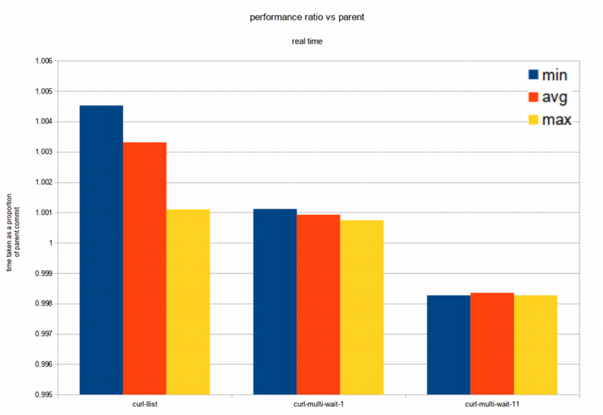
```
[value:final-states] Values at end of function main:
  __fc_random_counter ∈ [--..--]
  __fc_heap_status ∈ [--..--]
  arrLen ∈ [1..10]
  arr ∈ {{ NULL ; (int *)&__malloc_main_l10 }} or ESCAPINGADDR
  result ∈ {0}
[forgetful] Candidate for replacement in main: `free((void *)ar
r);` (test.c:17) frees base allocated at `
    int *arr = malloc(arrLen);` (test.c:10)
```

Results

Not as good as expected







✓ Sort Benchmark

✓ Parallel Benchmark

✗ Real-World Benchmark

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Thank You