

malloc() & free() in Scoped Memory

Adding Manual Memory Management to the Fiji™ VM

Prepared For:

CSE 605 - Advanced Concepts in Programming Languages
Dr. Lukasz Ziarek, Department of Computer Science & Engineering
University at Buffalo, The State University of New York

Prepared By:

Janhavi Patil (jpatil@buffalo.edu)
Mihir Mehta (mihirmeh@buffalo.edu)
Shreyas Chavan (schavan2@buffalo.edu)
Scott Florentino (scottflo@buffalo.edu)

Index

- Recap of our Implementation
- Changes implemented in Fiji™
- Tests and Results
- Remaining Implementation and Future Scope

Implementation Recap

- 100% Fragmentation-tolerant Allocation
 - Avoids need for cleanup
- Region for dynamic memory allocation
- Sets as Tracking Structure (Linked Lists)
- C Structures cast into Primitive and Array Blocks

Changes implemented in FijiVM™

- Wrapper Classes for hooking up Java to C
- Methods available to programmer
- Added Primitive allocation/ deallocation
- Added Array allocation/ deallocation

Primitive Allocation/ Deallocation

- Wrapper Classes
- Native methods declared in `UMUtils`
- Native methods defined in `fivmr_memoryareas.c`
- What these methods do ?
 - `allocate(value)`
 - `deallocate(pointer to the value)`
 - `get(pointer to the value)`
 - `set(pointer to the value, new value)`

Array Allocation/ Deallocation

- Implemented as proposed
 - BUT does not use Java Array semantics
- Utilize Fiji Runtime array safety checks
- Small & large array sizes tested
- Provide calculation for overhead
- Begin with Integer arrays

Tests

- Correctness
 - Simple & Corner cases
 - Single Threaded
 - Ensure no leaking
 - Asserts
- Performance
 - Basic Benchmark toolset
 - Utilizes complete fragmentation
 - Tests for both Heap & Scoped Memory

Test Results

- Correctness: passing
- Memory leakage: passing
- Initial Benchmark Results:

	Heap (CMR)	Heap (HF)	Unmanaged Memory
Array Allocation	150,656 ns	418,748 ns	38,900 ns
Array Element Mutation	7,003,764 ns	6,294,357 ns	7,786,323 ns
Matrix Multiplication	62,008,351 ns	98,474,551 ns	370, 922,406 ns

Remaining Implementation and Future scope

- Additional Error Checking/ Exception Handling
- Tests for Error Checking/ Exception Handling
- Multithreading
- Nested Scopes
- Possibly completely avoid memory waste in arrays