**Report of Assignment 3**

**There are 3 steps to complete the implementation of assignment3:**

🡪Identify the data object that used by @malloc

🡪Calculate the Load/store frequency

🡪Optimize the allocation for system with 3 different memory functions

1. **Identify the data object that used by @malloc**
2. By looking into the IR generated from F.dump(), we can find out the users’ chain from the below IR:

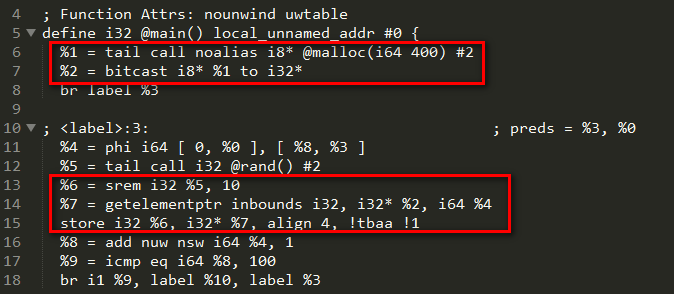
%1 = tail call noalias i8\* @malloc(i64 400) #2

%2 = bitcast i8\* %1 to i32\*

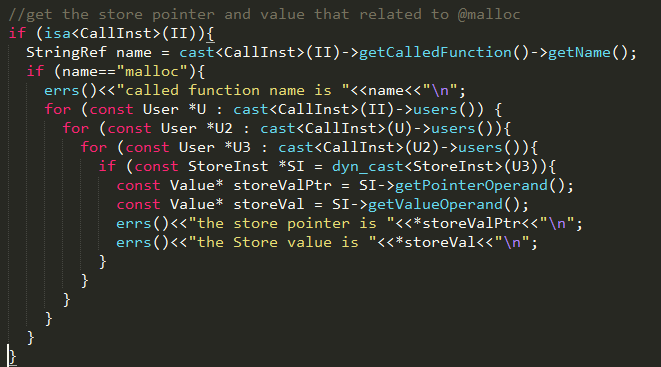
store i32 %6, i32\* %7, align 4, !tbaa !1

the user’s chain would be:

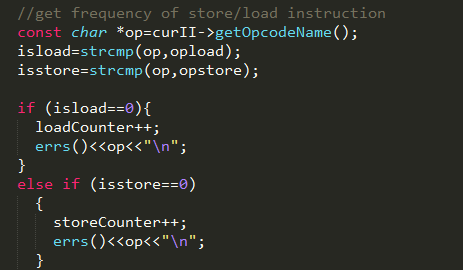
%1 is a user of callInst🡪2% is a user of 1%-->store is a user of 2%



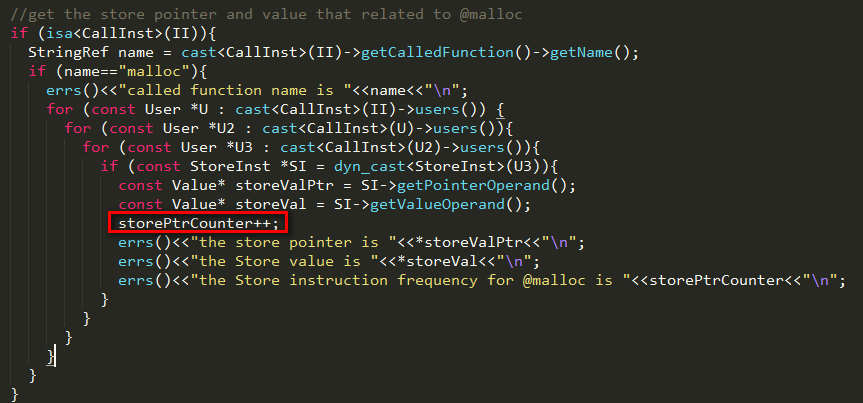
1. According to the users’ chain in step a, we can complete the implement of identify data object that called malloc, and get the corresponding store pointer and value:



1. **Calculate the Load/store frequency for the pointer.**
2. We can get the load/store frequency for the basic block as below



1. For @malloc, we don’t have load instruction, just store, and for the “store” frequency used for the pointer, we can count in user chain.



1. **Optimize the allocation for system with 3 different memory functions**
2. Implementation is not completed yet
3. Here are the scheme：

🡪get the size of value

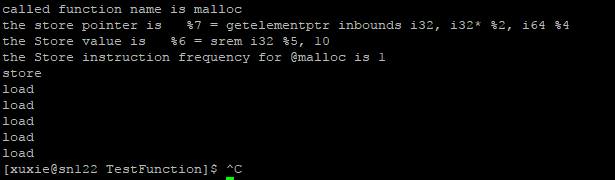
🡪consider the size and frequency of the write memory

1). If the memory read/write with very high frequency and low size, change the IR from @malloc to @malloc\_fast

2). If the memory read/write with very high frequency but large size, change the IR from @malloc to @malloc\_nvm

3). Other situation besides 1) and 2), will just use @malloc

1. **The result of my implementation is**



1. **Code implementation folder structure:**
2. Pass folder is in home/xuxie/workspace/llvm/build/lib/Transforms/MemoryOptimizer
3. Test function is in home/xuxie/workspace/llvm/build/lib/Transforms/TestFunction
4. cd to TestFunction, execute the below command you can get the result:

opt -load /gpfs/home/xuxie/workspace/llvm/build/lib/LLVMMemoryOptimizer.so -MemoryOptimizer < AccessFrqCal.bc > /dev/null