Compiler Hints for Co-accessed Variables (Update)

Group
Valerie Choung (vchoung@andrew.cmu.edu)
Daniel Ramos (drramos@andrew.cmu.edu)
(and Edward Chen (ejchen) working on an overlapping project)

URL
http://nicebowlofsoup.com/coaccess

Major Changes: None

What We Have Accomplished So Far:

- A compiler pass for identifying co-accesses for frequently used objects that are either heap or stack allocated. We track the number of uses for a co-access.
- A compiler pass for dependency analysis, for moving the calculation of the sizes passed to malloc. We can move the calculations around, say, to the start of a basic block.
- A compiler pass for (conservatively) identifying calls to malloc that could be merged.

Meeting Our Milestone:
We met most of our milestone goals and also have some work on goals past the milestone. (So, we didn't do exactly what our milestone goals were but did roughly the same amount of work.)

Surprises:
The part of the milestone we didn't complete yet was actually replacing the malloc calls with a single malloc and then (offset) pointer dereferences. This is because I realized that not all malloc calls even with statically known sizes are easy to coalesce, since the calls to malloc can be split among several basic blocks, as can their corresponding calls to free. The conservative approach is to only coalesce the malloc calls if all the malloc calls are contained within a basic block *and* all the free calls are contained within a basic block. If we have time, we will think of a better heuristic / approach for determining if a set of mallocs should be merged.

Revised Schedule (which is pretty the same as before)
We will pair program the more complex logic, and split the small tasks for this project.

~~*Week of 3/28:* Simple static co-access analysis with some notion of strength.~~
Week of ***4/18****:* Coalesce calls to *malloc* with sizes known at compile time.
~~*Week of 4/11:* Dependency analysis for *malloc* sizes not known at compile time and move size computations around accordingly.~~
Week of 4/18: Design co-access analysis for variables allocated within a loop. (can be simple)
Week of 4/25: Implement co-access analysis for variables allocated within a loop.

Resources
No need for anything else. We are using the same VM environment provided for our assignments.