**Work Items**

Contact [Lars Bergstrom](mailto:Lars.Bergstrom@uchicago.edu) to learn more about any project or get started. Independent research credit is available for University of Chicago undergraduates and there may be NSF REU financial support, depending on the student and quarter.

**In progress**

Nora Sandler - Reflow analysis  
Stephen Rosen - Flattening in BOM

**Smaller, undergraduate-friendly projects**

Tuning the garbage collector. How large should the nursery be relative to cores and caches? How large should the global heap be relative to the number of processors and total available RAM?

Porting the Programming Language Shootout benchmarks [[1](http://shootout.alioth.debian.org)] to Manticore. Identifying holes in our generated code pointed out by these benchmarks.

Better handling of small allocated constants (i.e. boxed 0.0). These should be allocated only once per program run.

Porting Matthew's summer school notes into a Real Manual/walkthrough

Replace our current perfcounter work with LIKWID support for better measurement of characteristics such as L3 cache miss rates separated by program, minor, major, and global GC.

Monomorphisation [[2](http://mlton.org/Monomorphise)] turns a polymorphic program into a monomorphic one by instantiating polymorphic values and datatypes for each type at which they are used. We would like to do this on our AST representation, as it allows significantly more optimization in later phases of the compiler.

I/O is currently handled synchronously. We need to add async I/O support, in particular allowing a vproc to perform other work until that thread is ready to run again.

**Major bugs that need fixing**

The round-robin and work-stealing schedulers do not interact well and there is a race condition on 2 cores that can lead to infinite spinning.

**Heavy-duty projects**

- Control-flow analysis
- Add location info throughout the representations for debugging
- Switch to an LLVM-based backend
- Automatic generation of primops.
- Hierarchical namespaces for hlops
- Compiler optimizations
- Runtime system
- Scheduler
- Atomicity
- Compile on Windows