Registers and Threads

M. Anton Ertl
TU Wien

The Setting

- Multi-Threaded Programs
- Non-blocking GC
- (At first) Don't worry about unbounded memory

The Problem

- How do we find the roots of other threads?
- We cannot look into their registers!

The Solution

- Compiler stores register-only roots explicitly
- ... in root variables (at most one per register)

How much storing is necessary?

- Allocator needs to store
  until the program stores the pointer in memory

- Pointer user needs to store if it does not know better
  e.g., list pointer passed into function through register
  only head needs to be stored in traversal
  think through calling convention

Reducing leakage

- Overwrite outdated root variables

- on loop back edge during traversal to avoid unbounded leaks

- unroll to reduce overhead

Cost

```
volatile struct list *root; /* global */

for (i=0; i<n; i++) {
  l=l->next;
  root=l; /* keep root variable current */
}
```

- On Xeon 5450
  3 cycles per iteration

- without **and with** `root=l;`