Interpreter Optimization: Dynamic Replication

- Each instruction has its own dispatch body (handler)
  - Dynamic – make a copy for each instruction, flush icache \textit{dynamically}
    - Concatenate dispatch bodies
    - Requires that code be relocatable
    - Note that this is one dispatch body for each unique instruction in a program
      - Repeated execution of the same instruction will use the same dispatch routine – thus is more predictable

![Diagram of Dynamic Replication]

- Diagram showing decode & dispatch process with VM code and VM routines (handlers) for instructions iload, iadd, iload.
Interpretation Optimization: Static Replication

- Each instruction has its own dispatch body
  - Static – make multiple copies for each operation, reroute execution of instructions to different copies --- use a greedy algorithm for rerouting
    - Note that this has no notion of a program – this is done at interpreter build time
      - So we have to guess how many copies of each dispatch routine to make
      - Figuring this out: Run a bunch of programs, profile them, collect data on the most important instructions and the number of different instances they are likely to have