DECAN: Decremental Performance Analysis Tool via Binary Patching

Souad Koliai
Ph.D. Candidate
University of Versailles
souad.koliai@prism.uvsq.fr

Grigori Fursin
Doctor
University of Versailles
grigori.fursin@prism.uvsq.fr

Tipp Moseley
Doctor
Google Corp.
tipp@google.com

William Jalby
Professor
University of Versailles
william.jalby@prism.uvsq.fr

Optimization process
- Gathering data (ie. code characterization);
- Diagnosing the problem;
- Prescribing a solution.

Tedious process
- Complex modern processors;
- Limited existing methodologies;
- Performance counters not understandable.

Characterization process
- Code analysis to extract code characteristics;
- Applying different types of the code analysis;
- Get different views of the code behavior.

What is DECAN?
- DECAN is a tool for a fine-grained detection of bottlenecks (ie. assembly instruction level);
- DECAN focuses on the hot region of the application;
- DECAN performs on a binary level;
- DECAN uses Pin to patch the binary.

DECAN general concept
1. Measure the instruction binary;
2. Patch (ie.remove) the memory access instructions in the original binary;
3. New binary is generated for each patch
4. Measure new binaries;
5. Measure are represented in a CSV file.

How DECAN removes the SSE memory instructions?

Original code
Loop:
movsd (%rsi,%rax),%xmm1
mulsd %xmm0,%xmm1
addsd (%rsi,%rax),%xmm1
movsd %xmm1,(%rsi,%rax)
inc %rax
jb Loop

One load
- Replace each SSE load with a nop
- Each replacement generates a new binary

One store
- Replace each SSE store with a nop
- Each replacement generates a new binary

All stores
- Replace all SSE stores with nops and generate a new binary

All loads/stores
- Replace all SSE loads and stores with nops and generate a new binary

Grouping
- Remove (ie.replace by nop) all loads that access to the same base address

RBgauss subroutine
- Removing a set of loads accessing to the same base address (AM array) --> lower bound achieved.
- The bottleneck is the AM access and only AM.
- Stride 2 access that causes bottleneck only in AM access.
- Bottleneck precisely pinpointed, applying an optimization on AM access.

Matvec subroutine
- Test DECAN on more applications (SPEC 2006).
- Improve user’s feedback: synthesis of DECAN’s results.
- Extend DECAN to address branch instructions to detect miss-prediction.