Performance engineering with performance profiling tool

Seyed Hasan Mohaghegh Beheshti

Simple Task

Sorting

Bubble Sort

```
void bubble_sort(int *a, int n)
{
    int i, t, s = 1;
    while (s)
    {
        s = 0;
        for (i = 1; i < n; i++)
        {
            if (a[i] < a[i - 1])
            {
                t = a[i];
                a[i] = a[i - 1];
                a[i - 1] = t;
                s = 1;
            }
        }
    }
}
```

Performance Measurement

Use Clock & Time

Start / Stop Watch

```
static struct timeval tm1;

static inline void start()
{
    gettimeofday(&tm1, NULL);
}

static inline void stop()
{
    struct timeval tm2;
    gettimeofday(&tm2, NULL);
    unsigned long long t = 1000 * (tm2.tv_sec - tm1.tv_sec) +
                                   (tm2.tv_usec - tm1.tv_usec) / 1000;
    printf("%llu ms\n", t);
}
```

Demo

“Premature optimization is the root of all evil”

– Donald Knuth

What is the bottleneck

CPU - Memory - Network - Disk - User

Use Performance Profiling Tools

Performance profiling tools types

Static Instrumentation	gprof
Dynamic Instrumentation	callgrind, cachegrind, DTrace
Performance Counters	oprofile, perf
Heap Profiling	massif, google-perftools

And other tools for other spaces, "Network", "I/O" …

Perf

Perf Commands

- perf top - Show hotspots
- perf stat - Show event counters
- perf record - Recording & sampling
- perf report - Show recording results
- perf annotate - Show annotated performance in code
  Special compile flag needed if you want see your written code lines.
- …

Demo

Now time to

"Optimization"

Compiler Features

https://linux.die.net/man/1/g++

https://linux.die.net/man/1/g++

Optimization Options

-falign-functions[=*n*] -falign-jumps[=*n*] -falign-labels[=*n*] -falign-loops[=*n*] -fassociative-math -fauto-inc-dec -fbranch-probabilities -fbranch-target-load-optimize -fbranch-target-load-optimize2 -fbtr-bb-exclusive -fcaller-saves -fcheck-data-deps -fconserve-stack -fcprop-registers -fcrossjumping -fcse-follow-jumps -fcse-skip-blocks -fcx-fortran-rules -fcx-limited-range -fdata-sections -fdce -fdce -fdelayed-branch -fdelete-null-pointer-checks -fdse -fdse -fearly-inlining -fexpensive-optimizations -ffast-math -ffinite-math-only -ffloat-store -fforward-propagate -ffunction-sections -fgcse -fgcse-after-reload -fgcse-las -fgcse-lm -fgcse-sm -fif-conversion -fif-conversion2 -findirect-inlining -finline-functions -finline-functions-called-once -finline-limit=*n* -finline-small-functions -fipa-cp -fipa-cp-clone -fipa-matrix-reorg -fipa-pta -fipa-pure-const -fipa-reference -fipa-struct-reorg -fipa-type-escape -fira-algorithm=*algorithm* -fira-region=*region* -fira-coalesce -fno-ira-share-save-slots -fno-ira-share-spill-slots -fira-verbose=*n* -fivopts -fkeep-inline-functions -fkeep-static-consts -floop-block -floop-interchange -floop-strip-mine -fmerge-all-constants -fmerge-constants -fmodulo-sched -fmodulo-sched-allow-regmoves -fmove-loop-invariants -fmudflap -fmudflapir -fmudflapth -fno-branch-count-reg -fno-default-inline -fno-defer-pop -fno-function-cse -fno-guess-branch-probability -fno-inline -fno-math-errno -fno-peephole -fno-peephole2 -fno-sched-interblock -fno-sched-spec -fno-signed-zeros -fno-toplevel-reorder -fno-trapping-math -fno-zero-initialized-in-bss -fomit-frame-pointer -foptimize-register-move -foptimize-sibling-calls -fpeel-loops -fpredictive-commoning -fprefetch-loop-arrays -fprofile-correction -fprofile-dir=*path* -fprofile-generate -fprofile-generate=*path* -fprofile-use -fprofile-use=*path* -fprofile-values -freciprocal-math -fregmove -frename-registers -freorder-blocks -freorder-blocks-and-partition -freorder-functions -frerun-cse-after-loop -freschedule-modulo-scheduled-loops -frounding-math -frtl-abstract-sequences -fsched2-use-superblocks -fsched2-use-traces -fsched-spec-load -fsched-spec-load-dangerous -fsched-stalled-insns-dep[=*n*] -fsched-stalled-insns[=*n*] -fschedule-insns -fschedule-insns2 -fsection-anchors -fsee -fselective-scheduling -fselective-scheduling2 -fsel-sched-pipelining -fsel-sched-pipelining-outer-loops -fsignaling-nans -fsingle-precision-constant -fsplit-ivs-in-unroller -fsplit-wide-types -fstack-protector -fstack-protector-all -fstrict-aliasing -fstrict-overflow -fthread-jumps -ftracer -ftree-builtin-call-dce -ftree-ccp -ftree-ch -ftree-coalesce-inline-vars -ftree-coalesce-vars -ftree-copy-prop -ftree-copyrename -ftree-dce -ftree-dominator-opts -ftree-dse -ftree-fre -ftree-loop-im -ftree-loop-distribution -ftree-loop-ivcanon -ftree-loop-linear -ftree-loop-optimize -ftree-parallelize-loops=*n* -ftree-pre -ftree-reassoc -ftree-sink -ftree-sra -ftree-switch-conversion -ftree-ter -ftree-vect-loop-version -ftree-vectorize -ftree-vrp -funit-at-a-time -funroll-all-loops -funroll-loops -funsafe-loop-optimizations -funsafe-math-optimizations -funswitch-loops -fvariable-expansion-in-unroller -fvect-cost-model -fvpt -fweb -fwhole-program --param *name=value* -O -O0 -O1 -O2 -O3 -Os

Demo

Feedback-Directed Optimization

https://linux.die.net/man/1/g++

Optimization Options

-falign-functions[=*n*] -falign-jumps[=*n*] -falign-labels[=*n*] -falign-loops[=*n*] -fassociative-math -fauto-inc-dec -fbranch-probabilities -fbranch-target-load-optimize -fbranch-target-load-optimize2 -fbtr-bb-exclusive -fcaller-saves -fcheck-data-deps -fconserve-stack -fcprop-registers -fcrossjumping -fcse-follow-jumps -fcse-skip-blocks -fcx-fortran-rules -fcx-limited-range -fdata-sections -fdce -fdce -fdelayed-branch -fdelete-null-pointer-checks -fdse -fdse -fearly-inlining -fexpensive-optimizations -ffast-math -ffinite-math-only -ffloat-store -fforward-propagate -ffunction-sections -fgcse -fgcse-after-reload -fgcse-las -fgcse-lm -fgcse-sm -fif-conversion -fif-conversion2 -findirect-inlining -finline-functions -finline-functions-called-once -finline-limit=*n* -finline-small-functions -fipa-cp -fipa-cp-clone -fipa-matrix-reorg -fipa-pta -fipa-pure-const -fipa-reference -fipa-struct-reorg -fipa-type-escape -fira-algorithm=*algorithm* -fira-region=*region* -fira-coalesce -fno-ira-share-save-slots -fno-ira-share-spill-slots -fira-verbose=*n* -fivopts -fkeep-inline-functions -fkeep-static-consts -floop-block -floop-interchange -floop-strip-mine -fmerge-all-constants -fmerge-constants -fmodulo-sched -fmodulo-sched-allow-regmoves -fmove-loop-invariants -fmudflap -fmudflapir -fmudflapth -fno-branch-count-reg -fno-default-inline -fno-defer-pop -fno-function-cse -fno-guess-branch-probability -fno-inline -fno-math-errno -fno-peephole -fno-peephole2 -fno-sched-interblock -fno-sched-spec -fno-signed-zeros -fno-toplevel-reorder -fno-trapping-math -fno-zero-initialized-in-bss -fomit-frame-pointer -foptimize-register-move -foptimize-sibling-calls -fpeel-loops -fpredictive-commoning -fprefetch-loop-arrays -fprofile-correction -fprofile-dir=*path* -fprofile-generate -fprofile-generate=*path* -fprofile-use -fprofile-use=*path* -fprofile-values -freciprocal-math -fregmove -frename-registers -freorder-blocks -freorder-blocks-and-partition -freorder-functions -frerun-cse-after-loop -freschedule-modulo-scheduled-loops -frounding-math -frtl-abstract-sequences -fsched2-use-superblocks -fsched2-use-traces -fsched-spec-load -fsched-spec-load-dangerous -fsched-stalled-insns-dep[=*n*] -fsched-stalled-insns[=*n*] -fschedule-insns -fschedule-insns2 -fsection-anchors -fsee -fselective-scheduling -fselective-scheduling2 -fsel-sched-pipelining -fsel-sched-pipelining-outer-loops -fsignaling-nans -fsingle-precision-constant -fsplit-ivs-in-unroller -fsplit-wide-types -fstack-protector -fstack-protector-all -fstrict-aliasing -fstrict-overflow -fthread-jumps -ftracer -ftree-builtin-call-dce -ftree-ccp -ftree-ch -ftree-coalesce-inline-vars -ftree-coalesce-vars -ftree-copy-prop -ftree-copyrename -ftree-dce -ftree-dominator-opts -ftree-dse -ftree-fre -ftree-loop-im -ftree-loop-distribution -ftree-loop-ivcanon -ftree-loop-linear -ftree-loop-optimize -ftree-parallelize-loops=*n* -ftree-pre -ftree-reassoc -ftree-sink -ftree-sra -ftree-switch-conversion -ftree-ter -ftree-vect-loop-version -ftree-vectorize -ftree-vrp -funit-at-a-time -funroll-all-loops -funroll-loops -funsafe-loop-optimizations -funsafe-math-optimizations -funswitch-loops -fvariable-expansion-in-unroller -fvect-cost-model -fvpt -fweb -fwhole-program --param *name*=*value* -O -O0 -O1 -O2 -O3 -Os

Demo

Use **perf record** command for compiler

Introducing operf.py

`**pmu-tools** is my toolkit to make access to these raw events **more user-friendly** for **Intel CPUs**, and provide **some additional functionality** for **perf**`

AutoFDO

Demo

Better implementation of sorting algorithm

Merge Sort

$O(n^2)$ VS $O(n.\log(n))$

Demo

Getting rid of merge sort branches

```
while (i <= j1 && j <= j2)
{
    if (a[i] < a[j])
        temp[k++] = a[i++];
    else
        temp[k++] = a[j++];
}
```

Getting rid of merge sort branches

```
while (i <= j1 && j <= j2)
{
    if (a[i] < a[j])
        temp[k++] = a[i++];
    else
        temp[k++] = a[j++];
}
```

```
while (i <= j1 && j <= j2)
{
    int cmp = (a[i] <= a[j]);
    int min = a[j] ^ ((a[i] ^ a[j]) & (-cmp));
    temp[k++] = min;
    i += cmp;
    j += !cmp;
}
```

Demo

Overview

- Find hotspots
- Find better algorithm
- Better implementation
- Compiler tools

Even More

Overview

- Find hotspots
- Find better algorithm
- Better implementation
- Compiler tools
- Use all resources if you can

Parallelism

- Use Intel CPU SIMD
- Parallel programming
  - OpenMP
  - pThread
  - Cilk-Plus
- Many-Core & GPU

SIMD

```
float fSRes;
float fVRes;

float *v1, *v2;
v1 = new float [VECTOR_SIZE];
v2 = new float [VECTOR_SIZE];

if (!v1 || !v2) {
    printf ("Memory allocation error!!\n");
    return 1;
}
// Initialize vectors with random numbers
for (long i = 0; i < VECTOR_SIZE; i++)
{
    v1[i] = (float) rand();
    v2[i] = (float) rand();
}
```

SIMD

Serial

```
for (long i = 0; i < VECTOR_SIZE; i++)
    fSRes += (v1[i] * v2[i]);
```

Parallel

```
__m128 sum = _mm_set1_ps(0.0f);
for (long i = 0; i < VECTOR_SIZE; i+=4)
    sum = _mm_add_ps (sum, _mm_mul_ps (_mm_loadu_ps (&v1[i]), _mm_loadu_ps (&v2[i])));
sum = _mm_hadd_ps (sum, sum);
sum = _mm_hadd_ps (sum, sum);
fVRes = _mm_cvtss_f32 (sum);
```

Demo

Intel VTune Amplifier XE

Intel parallelized program profiler

Play With Perf

Play With Perf

- perf top
- perf record
- perf stat -r 5 stop 5
- perf report in file
- perf annotate

Demo