Chapter 2: Case study

Version 6: Prefetching [advanced]

Earlier we discussed hardware prefetching: for example, when the CPU detects that the program seems to be doing linear reading, it may automatically start to fetch data that we might need in the future. Instead of relying on hardware prefetching, we can also do software prefetching: we can explicitly add instructions that ask the CPU to fetch data that we might need in the future.

In GCC, we can use the built-in function `__builtin_prefetch` to give such hints. It is important to keep in mind that these hints will get translated into machine language instructions, and this means that there will be more work for the CPU. However, as we saw in the previous section, the execution ports that are associated with memory accesses are currently underused, so it makes sense to try out if software prefetching would help.

Implementation

Here is the innermost loop, with the prefetch instructions added. Based on benchmarks, prefetching information that will be needed 20 iterations later seemed to give a good performance.

```
for (int k = 0; k < n; ++k) {
    constexpr int PF = 20;
    __builtin_prefetch(&vd[n*ia + k + PF]);
    __builtin_prefetch(&vt[n*ja + k + PF]);
    float8_t a000 = vd[n*ia + k];
    float8_t b000 = vt[n*ja + k];
    float8_t a100 = swap4(a000);
    float8_t a010 = swap2(a000);
    float8_t a110 = swap2(a100);
    float8_t b001 = swap1(b000);
    vv000 = min8(vv000, a000 + b000);
    vv001 = min8(vv001, a000 + b001);
    vv010 = min8(vv010, a010 + b000);
    vv011 = min8(vv011, a010 + b001);
    vv100 = min8(vv100, a100 + b000);
    vv101 = min8(vv101, a100 + b001);
    vv110 = min8(vv110, a110 + b000);
    vv111 = min8(vv111, a110 + b001);
}
```

Note that we do not need to worry about addressing memory that is out of bounds when we use prefetch hints; it is harmless.

Assembly code

We can see the prefetch instructions also in the assembly code:

```
LOOP:
        vmovaps     -640(%rdx), %ymm2
        prefetcht0  (%rax)
        addq        $32, %rax
        vmovaps     -672(%rax), %ymm3
        prefetcht0  (%rdx)
        addq        $32, %rdx
        vpermilps   $177, %ymm2, %ymm0
        cmpq        %rcx, %rax
        vperm2f128  $1, %ymm3, %ymm3, %ymm13
        vaddps      %ymm2, %ymm3, %ymm15
        vpermilps   $78, %ymm3, %ymm14
        vaddps      %ymm0, %ymm3, %ymm3
        vpermilps   $78, %ymm13, %ymm1
        vminps      %ymm15, %ymm11, %ymm11
        vminps      %ymm3, %ymm7, %ymm7
        vaddps      %ymm14, %ymm2, %ymm3
        vaddps      %ymm14, %ymm0, %ymm14
        vminps      %ymm3, %ymm10, %ymm10
        vaddps      %ymm13, %ymm2, %ymm3
        vaddps      %ymm13, %ymm0, %ymm13
        vaddps      %ymm1, %ymm2, %ymm2
        vaddps      %ymm1, %ymm0, %ymm0
        vminps      %ymm14, %ymm6, %ymm6
        vminps      %ymm3, %ymm9, %ymm9
        vminps      %ymm13, %ymm5, %ymm5
        vminps      %ymm2, %ymm8, %ymm8
        vminps      %ymm0, %ymm4, %ymm4
        jne         LOOP
```

Results

We did not have much room left for improvement anymore, but prefetching nevertheless gives a visible performance improvement: