OpenMP SIMD directives

```
void v_add (float *c, float *a, float *b)
{
#pragma omp simd
   for (int i = 0; i <= MAX; i++)
     c[i] = a[i] + b[i];
}
```

Vectorization solutions

1. auto-vectorization (use a compiler switch and hope it vectorizes)

2. give your compiler hints and hope it vectorizes

3. code explicitly

openMP 4.0 #pragma omp simd

Cilk^{TM} Plus array notations

SIMD instruction intrinsics

Kernels: OpenMP 4.0 #pragma omp declare

Optimization Reports

-opt-report

-opt-report-file=[<name>|stdout|stderr]

“Vectorization Advisor” – Advisor XE

Application Tuning – Amplifier XE

Questions

1. What is the difference between parallelism and vectorization? Can we think vectorization as a kind of parallelism?

2. What would the compiler do if my loop or the length of a vector is larger than the cache line? Will it break the vector into small parts automatically?

Answers

1. No. Because a[i] and b[i] are not necessarily in different places of memory than c[i], every time an element is written into c[i] it may have changed an element of a[i] or b[i]. The original program was written to do the addition once at a time, so the compiler needs to take that completely literally.

2. Yes. The declare SIMD construct can be applied to a function to enable the creation of one or more versions that can process multiple arguments using SIMD instructions from a single invocation from a SIMD loop.