

ISPC

(Intel SPMD Program Compiler)

Introdução à Computação Paralela e Distribuída
MAC5742

Aluno: Grover E. Castro Guzman - 8883532

IME USP - 2015

O que é ISPC?

- É um linguagem de código aberto, baseada em LLVM e um compilador para as arquiteturas SIMD da Intel.
 - Gera códigos vetoriais de alto rendimento para Computação móvel e HPC.
 - SSE (128 bits) /AVX (256 bits)/Xeon Phi (512 bits)
 - Fácil de usar e integrar no código existente.
 - Versão atual 1.8.2

Motivação

- Aproveitar o paralelismo é essencial para obter o rendimento máximo do hardware atual.
 - Multicore: OpenMP, OpenMPI, etc.
 - SIMD: OpenCL, CUDA, etc.
- Produtividade do programador.
 - Utilizar código intrínseco.

Principais características

- Sintaxe familiar baseada em C.
- O código é sequencial, mas é executado em paralelo (SPMD).
- Mistura facilmente o cálculo de operações escalares e vectoriais.
 - Pode-se chamar código C/C++ do código ISPC.
 - Pode-se passar os ponteiros entre o ISPC e C/C++.
 - Não precisa fazer cópias entre a memória utilizada pelo ISPC e o CPU.

Linguagem ISPC

- Dois tipos novos de modificadores foram criados para distinguir entre tipos de dados escalares e vetoriais.
 - Os tipos de dados vetoriais tem uma largura fixa que é definida em tempo de compilação.
 - Escolhido pelo programador em tempo de compilação.
 - Define o número de números de ponto flutuante que podem utilizar os registros SIMD.

Modificadores

- uniform
 - Identifica dados escalares.
 - Cada pista (lane) do SIMD tem o mesmo valor.
- varying
 - Identifica dados vetoriais.
 - Cada pista (lane) do SIMD tem um valor diferente.

Exemplo

C++

```
int count = ...;
int * a = new int[count];
// inicializar a[...]
int sum = array_sum(a, count);
```

Código: main.cpp

ISPC

```
export int array_sum( uniform int a[], uniform int count){
    int partial = 0;
    for( uniform int i = 0; i < count; i += programCount)
        partial += a[i + programIndex];
    return reduce_add(partial);
}
```

Código: array_sum.ispc

Controles de fluxo

- ISPC tem todas as instruções de controle do C/C++.
 - Condicionais
 - If, switch
 - Laços
 - for, while, do...while
- Agregaram outros para obter melhor rendimento.
 - cif, cwhile, cdo, cfor, foreach, foreach_active, foreach_tiled, foreach_unique.

Controles de fluxo

- As condições dos valores uniformes são como C/C++.
- Que acontece se há valores diferentes?
 - O que acontece se o teste é só verdadeiro para algumas pistas(lanes)?

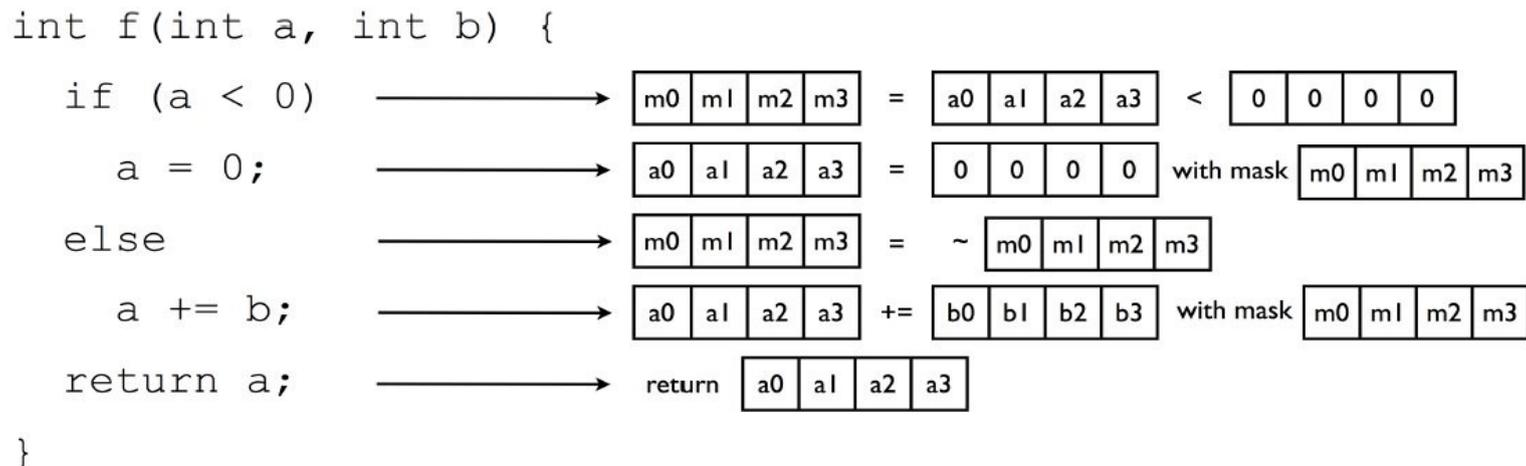
```
uniform int N = ...;

// simple conditional
if (N < 100) {
// do something
}

varying int Q = ...;
// hmm
if (Q < 100) {
// do something per lane
}
```

Controles de fluxo

- O hardware SIMD tem que fazer a mesma coisa para todas as pistas (lanes).
- O ISPC transforma os controles de fluxo em controles de dados.
- Só mudam os valores nas pistas ativas.



Exemplo multicore

```
const uniform int task_size = 4096;

task void sum_task(uniform int a[], uniform int count, uniform int &sum){
    uniform int start = task_size*taskIndex;
    uniform int end = min(task_size*(taskIndex + 1),count);
    int partial = 0;
    foreach(i = start ... end) partial += a[i];
    uniform int local_sum = reduce_add(partial);
    atomic_add_global(&sum,local_sum);
}

export array_sum(uniform int a[],uniform int count){
    int n_tasks = count/task_size;
    uniform int sum = 0;
    launch [n_task] sum_tasks(a,count,sum);
    return sum;
}
```

Código: task_array_sum.ispc

Usuários do ISPC

- Intel
- Embree
- Dreamworks Animation
- Pixar
- Autodesk
- SURFsara



Referências

- <https://github.com/ispc/ispc/>
 - Guia do Usuário, código fonte e exemplos.
- <http://pharr.org/matt/>
 - Página web de um dos criadores do ISPC, tem apresentações de onde foram obtidas as imagens, os códigos de exemplo para esta apresentação e outros conteúdos do ISPC.
- [ISPC na DreamWorks](#)