NEC SX-Aurora as a Scalable Vector Playground

Kazuhisa Ishizaka, Yoshiyuki Ohno, Yuta Ideguchi, Erich Focht, Simon Moll
simon.moll@emea.nec.com

NEC Corporation
**SX-Aurora TSUBASA Vector Computer**

**High performance computer** ranged from deskside to cloud and supercomputer

**Vector Processor on a Vector Engine PCIe accelerator card**

- **Vector Engine (VE)** (PCIe accelerator card)
- **Vector Processor with 6 HBM2s**

**Theoretical Performance**

**Computation:**
- FP32: 4.30 TFlops
- FP64: 2.15 TFlops

**Memory:**
- Capacity: 48GB
- Bandwidth: 1.2 TB/s
Vector Engine (VE) Scalable Vector ISA

Basics

- **Wide vector registers (256 x 64bit)**
  - 64 Vector registers (%v_i)

- **Full vector predication**
  - 16 Vector mask registers (%vm_i)
  - *Explicit* operand

- **(Active) Vector Length Register**
  - One, global, VL Register
  - *Implicit* dependence

```
%v0
%vm1
%v1
```

```
vfadd.d %v0,%v1,%v2,%vm1
```
Vector Engine Challenges in LLVM

1) Implicit dependency through VL register

```c
vl %s37
vld %v3,8,%s0
vld %v4,8,%s1
vfmad.d %v3,%v3,%s2,%v4
```

How do we implement implicit def-use?

2) Partial update of a destination vector register

```
v0 a255 ...... a128 a127 ...... a2 a1 a0
v1 b255 ...... b128 b127 ...... b2 b1 b0
v2 c255 ...... c128 a127 ...... c2 c1 c0
```

vfadd.d %v0,%v1,%v2 (VL=128)

How do we introduce partial update?

```
not updated updated(v1 + v2)
```
Our Solution: IR and Backend

1. Vector IR based on LLVM-VP
   • VL as parameter

   $$v3 = vfadd.d v4,v5,pt,vl$$
   # for i=0,256
   #   v3[i] = i < vl ? v4[i] + v5[i] : pt[i]

2. Automatic LVL generation in backend
   • Inserting LVL instruction from $vl$ argument in IR
   • Minimizing LVL instruction by current VL inference
Status
- LLVM-VE is available at https://github.com/sx-aurora-dev/llvm
- Scalar code backend + vector intrinsics
- Application: TensorFlow for SX-Aurora

Roadmap
- Upstreaming! https://reviews.llvm.org/D69103
- Vector predication with LLVM-VP (D57504)

Welcome collaborators
- for automatic vectorization, etc.

Come see us at the poster session!
"NEC SX-Aurora as a Scalable Vector Playground"