Scalable Matrix Architecture (Vector Extension Variant)

José Moreira
IBM Corporation

Summary (look at single-precision floating-point first)

- Let N be the size in (32-bit) words of a scalable vector:
  - $N = 4$: 128-bit vectors
  - $N = 8$: 256-bit vectors
  - $N = 16$: 512-bit vectors
- The architecture defines 8 accumulator registers
  - Each accumulator is organized as 4 rows of N columns each – each element is a word
  - Each accumulator overlaps 4 scalable vector registers ($8 \times 4 = 32$)
  - $\mathrm{ACC[AT]}[i][j]$ represents the word at row i, column j of accumulator AT
  - $\mathrm{ACC}[0{:}7][0{:}3][0{:}N-1]$ represents the space of words in accumulator file
- Outer-product operations are of the form

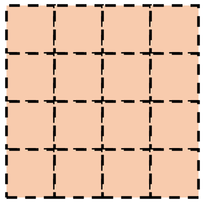
$$A\langle m_x, m_y\rangle = \pm x y^{\mathrm{T}} \pm A$$

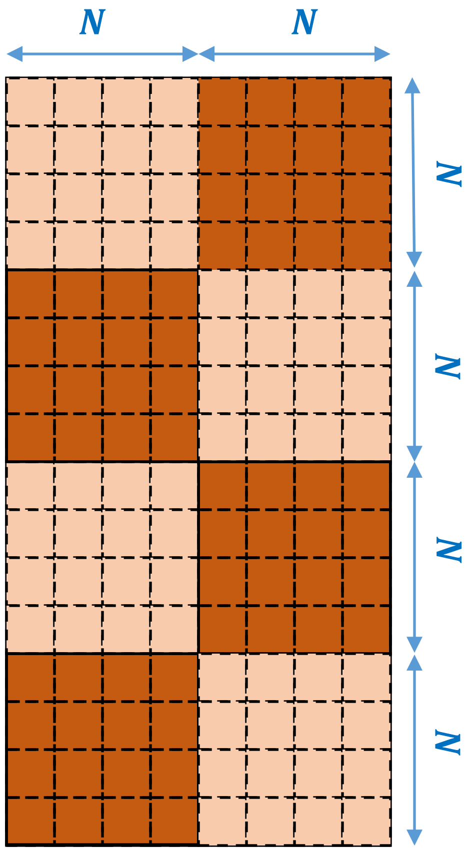
$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{41} & \cdots & a_{4n} \end{bmatrix} = \pm \begin{bmatrix} x_1 \\ \vdots \\ x_4 \end{bmatrix} \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix} \pm \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{41} & \cdots & a_{4n} \end{bmatrix}$$

  where
  - A is an accumulator register ($\mathrm{ACC}[i \in [0,8)]$)
  - x is a 4-element column vector, contained in a scalable vector register
  - y^{T} is an N-element row vector, contained in a scalable vector register
  - m_x is a mask for the rows of the accumulator, contained in a vector (mask) register
  - m_y is a mask for the columns of the accumulator, contained in a vector (mask) register

Example

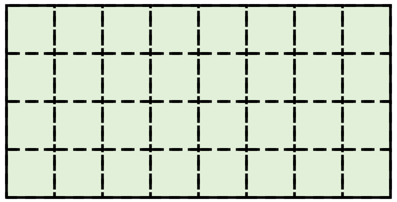
$N = 4$ (128-bit/16-byte scalable vectors)

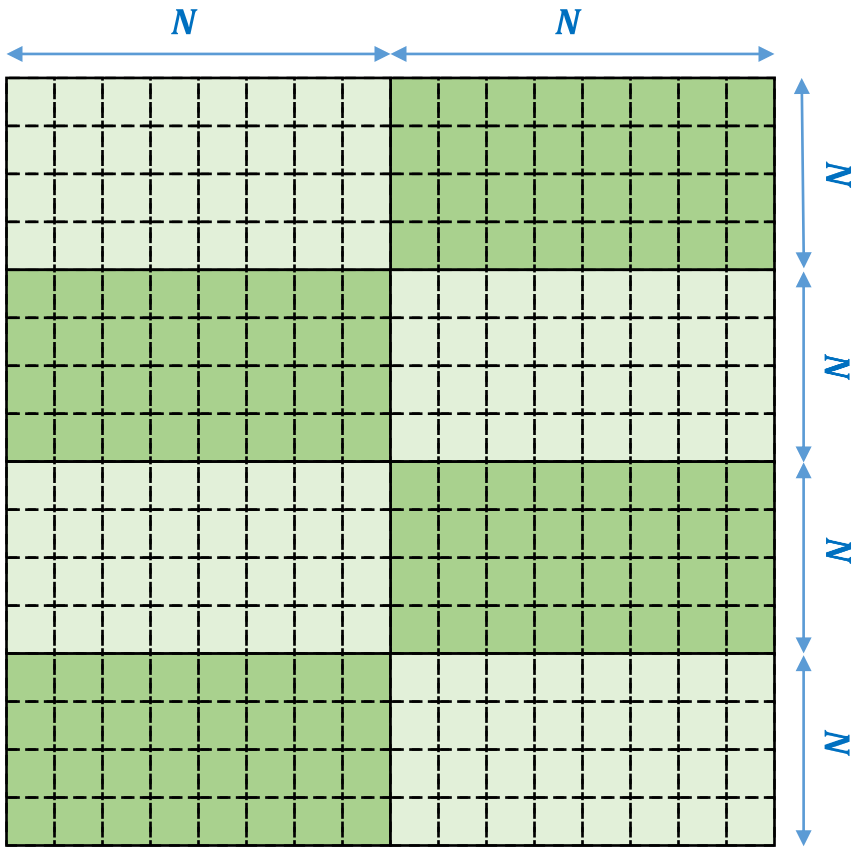




feature	value
each accumulator	4×4
accumulator file	16×8

$N = 8$ (256-bit/32-byte scalable vectors)





feature	value
each accumulator	4×8
accumulator file	16×16

Note: In this variant, the accumulators only scale along one axis – total accumulator space is equal to vector register file

SGEMM middle kernel ($\mathbf{16 \times 2}\boldsymbol{N}$ result panels)

$$A = XY^{\mathrm{T}}$$

$$\begin{bmatrix} [16 \times 2N] & \cdots & [16 \times 2N] \\ \vdots & \ddots & \vdots \\ [16 \times 2N] & \cdots & [16 \times 2N] \end{bmatrix} = \begin{bmatrix} [16 \times K] \\ \vdots \\ [16 \times K] \end{bmatrix} \times [[K \times 2N] \quad \cdots \quad [K \times 2N]]$$

$$[16 \times K] = \begin{bmatrix} X[0] + 0 & X[1] + 0 & \cdots \\ X[0] + 1 & X[1] + 1 & \cdots \\ \vdots & \vdots & \cdots \\ X[0] + 4 & X[1] + 4 & \cdots \\ \vdots & \vdots & \cdots \\ X[0] + 8 & X[1] + 8 & \cdots \\ \vdots & \vdots & \cdots \\ X[0] + 12 & X[1] + 12 & \cdots \\ \vdots & \vdots & \cdots \end{bmatrix}$$

$$[K \times 2N] = \begin{bmatrix} Y[0] + 0 & Y[0] + 1 & \cdots & Y[0] + N & \cdots \\ Y[1] + 0 & Y[1] + 1 & \cdots & Y[1] + N & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

SGEMM inner-most kernel ($16 \times 2N$ panel)

void sgemm_kernel(X,Y,m_X[0:3],m_Y[0:1],N,K) {

…

vector<float> u[0:1], v[0:3];

for (k=0; k<K; k++) {

$v[0]\langle m_X[0]\rangle \leftarrow X[k]; v[1]\langle m_X[1]\rangle \leftarrow X[k]+4; v[2]\langle m_X[2]\rangle \leftarrow X[k]+8; v[3]\langle m_X[3]\rangle \leftarrow X[k]+12;$

$u[0]\langle m_Y[0]\rangle \leftarrow Y[k]; u[1]\langle m_Y[1]\rangle \leftarrow Y[k]+N;$

$A[0]\langle m_X[0], m_Y[0]\rangle \leftarrow v[0] \times u^{\mathrm{T}}[0] + A[0];$

$A[1]\langle m_X[0], m_Y[1]\rangle \leftarrow v[0] \times u^{\mathrm{T}}[1] + A[1];$

$A[2]\langle m_X[1], m_Y[0]\rangle \leftarrow v[1] \times u^{\mathrm{T}}[0] + A[2];$

$A[3]\langle m_X[1], m_Y[1]\rangle \leftarrow v[1] \times u^{\mathrm{T}}[1] + A[3];$

$A[4]\langle m_X[2], m_Y[0]\rangle \leftarrow v[2] \times u^{\mathrm{T}}[0] + A[4];$

$A[5]\langle m_X[2], m_Y[1]\rangle \leftarrow v[2] \times u^{\mathrm{T}}[1] + A[5];$

$A[6]\langle m_X[3], m_Y[0]\rangle \leftarrow v[3] \times u^{\mathrm{T}}[0] + A[6];$

$A[7]\langle m_X[3], m_Y[1]\rangle \leftarrow v[3] \times u^{\mathrm{T}}[1] + A[7];$

}

…

}