Modeling Singular Valued Decomposition (SVD) Techniques using Parallel Programming with pMATLAB

Miguel Goenaga (Presenter)

Carlos J. González,

Inerys Otero

Juan Valera

Domingo Rodríguez, Advisor

University of Puerto Rico at Mayaguez

HPEC 2009

September 22, 2009

Problem Formulation

This work seeks the decomposition of a circulant matrix with circulant blocks using pMATLAB for filtering operations on hyperspectral images.

- Previous Work
  - ✓ Singular value decomposition transform with and FFT-based algorithm on the Connection Machine CM5 [1].
  - ✓ Kronecker products and SVD approximations in image restoration [2].
  - ✓ Effective algorithms with circulant-block matrices [3].
- Solution Approach

Given a circulant block matrix *A*, it is desirable to solve for the following proposition [1]:

$$A = U_A D_A V_A^T \qquad A = (F_M \otimes I_N)^{-1} U_D D_A V_D^T (F_M \otimes I_P)$$

The SVD of an N^2-by-N^2 circulant block matrix *A* is formulated according to [1]:

$$D_A = (F_N \otimes F_N) A (F_N \otimes F_N)^{-1} \qquad U_A = (F_M \otimes I_N)^{-1} U_D \qquad V_A^T = V_D^T (F_M \otimes I_P)$$

Implementation Results

- Mathematical Model Approach

  A Kronecker product commutation property from [4] was used to arrive at the model:

$$P_{N,R}(A_S \otimes B_R)P_{N,S} = B_R \otimes A_S$$

  The following transformation was obtained:

$$A = P_{N^2,N}(I_N \otimes F_N)^{-1}P_{N^2,N}U_D D V_D^T P_{N^2,N}(F_M \otimes I_P)P_{N^2,N}$$

  The modeling formulation for SVD computation is presented as follows :

$$A = P_{N^2,N}(I_N \otimes F_N)^{-1}U_D D_A V_D^T(I_N \otimes F_N)P_{N^2,N}$$

- Computational Parallel Pseudo-Algorithm

$SV \in \mathbb{R}^{P(N^2)x1} = svdParallel(h \in \mathbb{C}^{NxN})$

1. $h_v \in \mathbb{C}^{NxN};\ \ A, D_{\hat{h}_v} \in \mathbb{C}^{P(N^2)xN^2}$
2. $h_v = round(10 * rndn(N,N));$

   $\%\ Antilexical\ Vectorization\ \%$
3. $A.loc = \ CBMat(\ h_v.loc);$

   $\%\ Circular\ Block\ Matrix\ Generation\ \%$

$SV \in \mathbb{R}^{P(N^2)x1} = svdParallel(h \in \mathbb{C}^{NxN})cont.$

4. $D_{\hat{h}_v}.loc = (F_N \otimes F_N).loc * A.loc * \left(\frac{F_N^* \otimes F_N^*}{N^2}\right).loc$
5. $D_{\hat{h}_v} = agg(put\_local(D_{hv}, D_{hvplocal})\ ;$

   $\%\ Matrix\ Fusion\ \%$
6. $SV.loc = descsort(|D_{\hat{h}_v}|);$

   $\%\ Sort\ in\ Descending\ Order\ \%$

Performance Results

pMATLAB Contributions:

✓ Parallelized calculation of the circulant matrix with circulant blocks.
✓ Parallelized matrix multiplications and the use of Kronecker products.

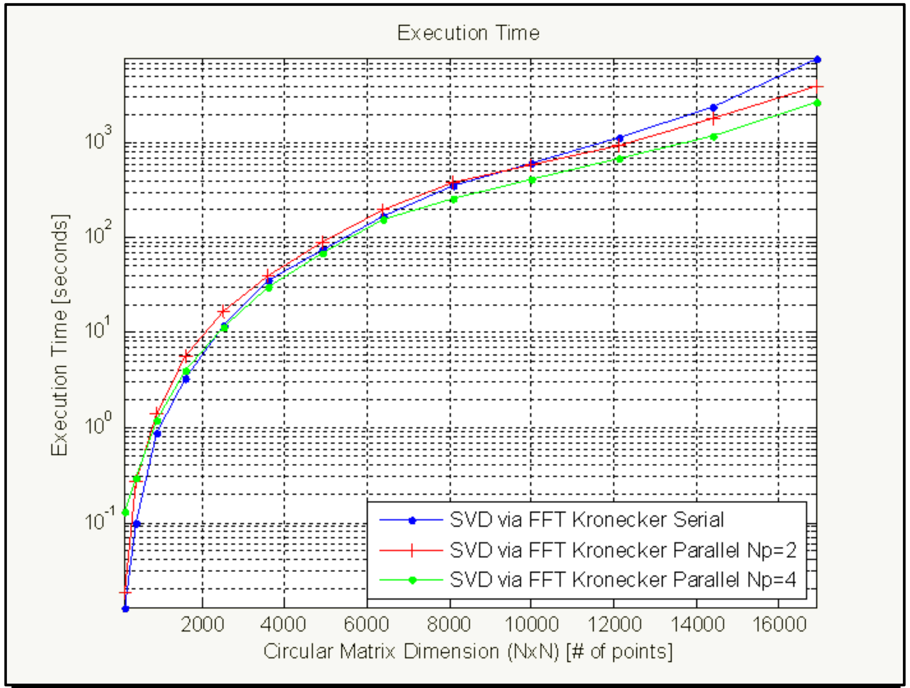

Figure 1: Execution time comparison between SVDs using a serial algorithm and the pMATLAB environment with *Np=2* and *Np=4*.

References:

[1] Cao-Huu, T. and Evequoz, C. "*Singular value decomposition transform with and FFT-based algorithm on the Connection Machine CM5,*" Electrical and Computer Engineering Canadian Conference, CCECE**,** vol. 2, pp.1046-1049, Sept. 5-8, 1995.

[2] Kamm, J. and Nagy, J. "*Kronecker product and SVD approximations in the image restoration",* Linear Algebra and its Applications, vol. 284, iss. 1-3, pp. 177-192, November 15, 1998.

[3] Rjasanow, S. "*Effective Algorithms With Circulant-Block Matrices,"* Linear algebra and its applications, vol. 202, pp. 55-69, 1994.

[4] Van Loan, C. F. *"The ubiquitous Kronecker product,"* Journal of Computational and Applied Mathematics, vol. 123, iss. 1-2, pp. 85-100, November 2000.