Randomization Algorithm to Compute Low-Rank Approximation

Student: Ru HAN (The Chinese University of Hong Kong)
Mentors: Ed D’Azevedo (ORNL), Ichitaro Yamazaki (UTK)

Abstract
A low-rank representation of a matrix provides a powerful tool for analyzing the data represented by the matrix. In this project, we implement "randomized" algorithm to compute the low-rank representation in the LAPCK/MAGMA/cuBLAS-XT software framework.

Randomized SVD Algorithm
q =.randn(n,k+1);
[q,r] = qr(q,0);
for iter=1:(max_iters-1)
    p = A*q;
    q = A'*p;
    [q,r] = qr(q,0);
end
p = A*q;
[p,b] = qr(p,0);
end
[x,s,y] = svdb(b);

Error = ||A-U_kS_kV_k^T||_2
        = (k+1)_b largest singular value of A

Out-of-core Randomized SVD

Method 1. Manual pipelining
• P=A*Q
P=0;
for k=1,2,3,…
    set (A_k to da);
    P=P+A_kQ_k;
end
• Q=A^T*P
for k=1,2,3,…
    set (A_k to da);
    Q_k=A_k^T*P;
end

Method 2. UMA&CUBLAS-XT
UMA is a programming model, Unified Memory Access. Unified Memory creates a pool of managed memory that is shared between the CPU and GPU.

The NVIDIA cuBLAS library is a fast GPU-accelerated implementation of the standard basic linear algebra subroutines (BLAS).

References