Mathematical Programming: one of the most central problems in mathematical optimization.

Many Applications: combinatorial optimization, control theory, structural optimization, quantum chemistry, sensor network location, data mining, etc.

1. SDPARA is a parallel implementation of the interior-point method for Semidefinite Programming (SDP)
   - Parallel computation for two major bottleneck parts
     - **ELEMENTS**: Computation of Schur complement matrix (SCM)
     - **CHOLESKY**: Cholesky factorization of Schur complement matrix (SCM)
   - SDPARA could attain high scalability using 16,320 CPU cores on the TSUBAME 2.0 supercomputer and some techniques of processor affinity and memory interleaving when generating the application of a large-scale problem.

   - **SDPARA** can solve the largest SDP problem
     - **DNN relaxation problem**: QAP10 QAPLIB
     - **High efficiency (7.7% for Electronic)**
     - **Higher efficiency when solving an SDP problem larger than Electronic because the efficiency for Electronic is higher than that for Electronic.

High-Performance General Solver for Extremely Large-scale Semidefinite Programming Problems (SDPs)

1. **CHOLESKY**: Sparse SCM
   - The matrix size $n$ ($n > m$) is distributed with block size $n$
   - The dense matrix $B(n x m)$ is distributed with block size $n$

2. **ELEMENTS**: Computation of Schur complement matrix (SCM)
   - The dense matrix $B(n x m)$ is distributed with block size $n$

Parallel Computation for **CHOLESKY**

For problems with $m \gg n$, high performance **CHOLESKY** is implemented for GPU supercomputers.

Key for petaflops is overlapping computation, PCI-Express communication and MPI communication.

Data Decomposition

- **The dense matrix $B(n x m)$ is distributed with block size $n$**
- Matrix distribution on $n$ processes

Basic Algorithm

For $k = 0, 1, 2, \ldots, \lfloor m/n \rfloor - 1$

1. **Diagonal block factorization**
2. **Panel factorization**
   - **Compute L by GPU DTRSM**
3. **Broadcast L, transpose $L'$, and broadcast $L''$**
4. **Update $L'' = B' \cdot L'$ with fast GPU DGEEMM**

**Design Strategy**

Our target is large problems with $m > 2$ million
- GPU memory is too small. Matrix data are usually placed on host memory
- Blocksize $n$ should be sufficiently large to mitigate GPU-GPU PCIe communication
- We still suffer from heavy PCIe communication
- 3D-Map GPU, PCIe comm., and MPI comm. in Step 2, 3 and 4

**Performance of CHOLESKY on TSUBAME2.0**

- **1 MPI process per GPU**
  - 3 processes per node

**Producer of panel L**

- **Consumers of panel L**
  - **DNN relaxation problem**: QAP10 QAPLIB
  - Using 1360 nodes, 2720 CPUs, 4080 M2050 GPUs
  - 1.018PFLOPS in CHOLESKY (2.33m x 2.33m)
  - 1.018PFLOPS in CHOLESKY (2.33m x 2.33m)

**SDPARA** can solve the largest SDP problem

- **DNN relaxation problem**: QAP11 QAPLIB with 2.33 million constraints
- Using 1360 nodes, 2720 CPUs, 4080 M2050 GPUs
- 1.018PFLOPS in CHOLESKY (2.33m x 2.33m)

**The fastest and largest result as mathematical optimization problems!!**

For inquiries please email to master@graphcrest.jp