Batched, Reproducible and Reduced Precision BLAS – Some Thoughts

By Bobby Cheng

5/19/2016

MATLAB and BLAS

- Cleve invented MATLAB so that his students could use LINPACK and EISPACK
- Uses BLAS routines since day one.
- Move to LAPACK in late ~1990s with ATLAS BLAS
- Now Intel MKL

What does reproducible mean to MATLAB?

- Two consecutive calls to the same function with identical inputs must yield identical results.
  - Same release of MATLAB
  - Same hardware and setting
  - Same just about everything

```
>> B = F(A);
>> C = F(A);
```

- Usability and productivity over performance
- Everything else are considered as portability

What does it mean to the library that MATLAB uses?

- Must itself be reproducible
  - simple
- Or must allow full control of its states, and made to be reproducible
  - random numbers
  - rounding mode
  - threading level
  - reduction
  - memory alignment
- Or limit how the library is called

BATCHED BLAS

- `pagefun` in Parallel Computing Toolbox for GPU

```
M = 3;          % output number of rows
K = 6;          % matrix multiply inner dimension
N = 2;          % output number of columns
P1 = 10;        % size of first page dimension
P2 = 17;        % size of second page dimension
P3 = 4;         % size of third page dimension
P4 = 12;        % size of fourth page dimension
A = rand(M,K,P1,1,P3,'gpuArray');
B = rand(K,N,1,P2,P3,P4,'gpuArray');
C = pagefun(@mtimes,A,B);
s = size(C)     % M-by-N-by-P1-by-P2-by-P3-by-P4
```

`pagefun` cont.

Currently the supported values for FUN are:

- Most element-wise gpuArray functions
- @ctranspose
- @fliplr
- @flipud
- @inv
- @mldivide
- @mrdivide
- @mtimes
- @rot90
- @transpose

What about MATLAB?

```
%Given A,B are 10 x 10 x 10000 arrays
C = zeros(10,10,10000);
for i = 1:10000
    C(:,:,i) = A(:,:,i)*B(:,:,i);
end
```

- Interpreted language means memory allocation costs is runtime cost.
  - Batched BLAS flexible interface may be a concern.
- Memory allocation dominates! (Well, at least on Windows)
- Expressions are often more complicated.
- Error Handling
- Need new data structure for variable size problem.

Reliable NaN Propagation, Traceable Failure

- Once NaNs appear, NaNs must be propagated for traceability.
- BLAS – daxpy (aX + Y)
  - NaN progagate exception if a is zero.
  - Work around this in Batched BLAS could be painful.
- Important enough to now follow IEEE 754
  - `hypot(Inf, NaN)`
  - `1 .^ NaN`

Reduced Precision BLAS

- MATLAB to Simulink to Code generation
- Embedded target (lesser platform)
- Native precision implementation may be important
- Reproducible results would be great
- Portable results would be acceptable

Summary

- MathWorks love what you all are doing.
- Many small problems is an important topic to understand.
- Need more experiment to explore the potential.
- Looking forward to experiment with it through MATLAB

- Value based optimization at BLAS level can be problematic.