FLAGON : Fortran-9X Library for GPU Numerics

Nail Gumerov, Yuancheng Luo, Ramani Duraiswami, Kate DeSpain, Bill Dorland, Qi Hu

What Is Flagon:

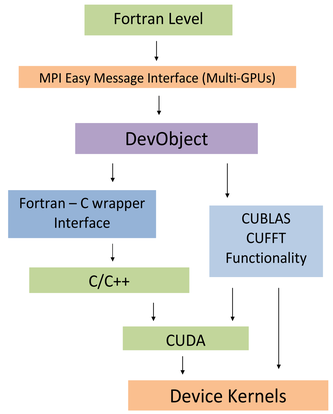
- Flagon is a library and a middle-ware framework that allows use of NVIDIA CUDA in FORTRAN
- Flagon is to ease programming on the GPU from high level languages, including Fortran 9x, C++, and Matlab
- Flagon is implemented on Intel Fortran on Linux, Windows and MAC OS

Flagon Features:

- External CUDA kernel loaders and generic kernel callers allow custom functions to be added and used
- Pointer like data structures that can be manipulated on the host, used in function arguments etc
- Minimizes data transfers between host and device
- Multi-GPU communications
- Current integrated libraries
  - CUBLAS/CUFFT
  - CUDPP
  - MPICH2
  - Matrix factorization: LU/QR

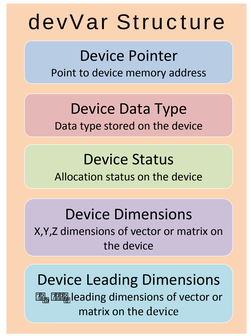
Flagon Framework:

- FORTRAN Layer
  - Device Variables (devVar) communicate with lower levels
  - Fortran interfaces and wrappers pass parameters to C/C++ level
  - May directly call CUBLAS/CUFFT library functions
- MPI Layer
  - Launches multiple processes via MPICH2
  - Each MPI process controls one GPU card
  - GPUs talk to their control processes while messages exchange among the control processes



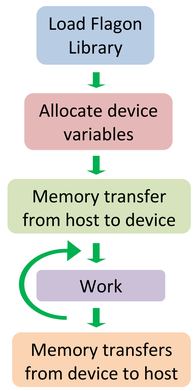
- C/C++ Layer
  - Communicates with CUDA kernels
  - Setup function calls, parameter passing to kernels
  - Module management of external functions
- CUDA Layer
  - Performs operations on the device

Flagon Device Variables:



- User instantiates device variables in Fortran
- Encapsulates parameters and attributes of the data structure transferred between host and device
- Tracks (via pointers) allocated memory on the device
- Stores data attributes (type and dimensions) on the host and device

Flagon Work-Cycle:



- Compile and link library to user Fortran code
- Load library into memory
- Allocate device variables and copy host data to device
- Work-cycle allows subsequent computations to be performed solely on the device (call CUDA functions, etc)
- Data transfer from device to host when done
- Discard/free data on the device

Using Flagon

- Code conversion:

Original Fortran Code

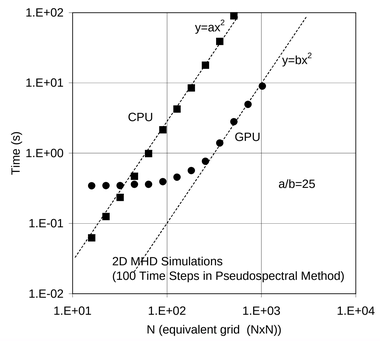
```
subroutine nlterms(isave,a,b,nl)

use com_module


complex,dimension(:,:),intent(in) ::a,b
complex,dimension(:,:),intent(out)::  nl
integer ::  isave

if(isave /= 1) then
     kxa2=ikx*a
     kya2=iky*a
     call fftwnd_f77_one(plan_f,kxa2,kxa2)
     call fftwnd_f77_one(plan_f,kya2,kya2)
endif

if(isave /= 2) then
     kxb2=ikx*b
     kyb2=iky*b
     call fftwnd_f77_one(plan_f,kxb2,kxb2)
     call fftwnd_f77_one(plan_f,kyb2,kyb2)
end if

nl=kxa2*kyb2-kxb2*kya2
call fftwnd_f77_one(plan_b,nl,nl)
nl=nl*scale

end subroutine nlterms
```

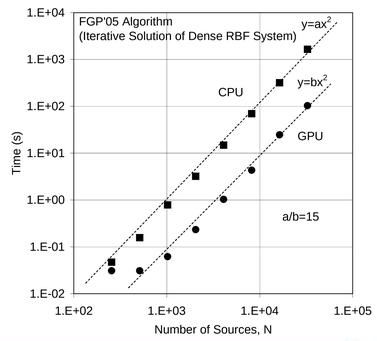
Modified Fortran Code

```
subroutine dev_nlterms(isave,dv_a,dv_b,dv_nl)

use dv_com_module
use mod_devObject

type(devVar),intent(in) ::  dv_a,dv_b
type(devVar),intent(out)::  dv_nl
integer ::  isave

if(isave /= 1) then
     call devf_mul1(dv_kxa2,dv_kya2,dv_ikx,dv_iky,a)
     call devf_fft(dv_kxa2,fftplan)
     call devf_fft(dv_kya2,fftplan)
endif

if(isave /= 2) then
     call devf_mul1(dv_kxb2,dv_kyb2,dv_ikx,dv_iky,b)
     call devf_fft(dv_kxb2,fftplan)
     call devf_fft(dv_kyb2,fftplan)
endif

call devf_mul2(dv_nl,dv_kxa2,dv_kyb2,dv_kya2,dv_kxb2)
call devf_ifft(dv_nl,fftplan)
call devf_mul(dv_nl,dv_nl,dv_scale)

end subroutine dv_nlterms
```

Example of code conversion using Flagon: Left: original Fortran-90 code; Right: Flagon code.

- Easy to build custom functions
  1. Method One
     - Write the CUDA and C/C++ files to define custom functions
     - Define the Fortran interfaces for these functions
     - Call custom functions in Fortran code via the defined interface
  2. Method Two
     - Write compatible CUDA functions for Flagon
     - Compile the [cuFile].cu into [cuFile].cubin
     - Load [cuFile].cubin during runtime (call fc_LoadDevFunc( ) fc_GetDevFuncPtr( ) devf_gen_pointwise_full( )or call devf_explicit_execute( ))
- Performance Summary





UMIACS
University of Maryland Institute for Advanced Computer Studies

Flagon Website:
http://sourceforge.net/projects/flagon/