HIP
Performance Portability
&
GPUS

Dr. Joseph Schoonover

CEO, Cloud-HPC Systems Architect
Fluid Numerics, LLC
Boulder, CO

joe@fluidnumerics.com

FluidNumerics

HPC GPU Programming Ecosystem

Compiler →

API ↓

	GCC	Intel	PGI	XL	CLANG
OpenCL	🟩	🟩	🟩	🟩	🟩
OpenMP	🟩	🟥	🟥	🟩	🟩
OpenACC	🟥	🟥	🟩	🟥	🟩
CUDA	🟩	🟥	🟥	🟥	🟥
HIP	🟩	🟥	🟥	🟥	🟩

HPC GPU Programming Ecosystem

Architecture

API

	Nvidia	AMD	Arm	Intel
OpenCL	🟩	🟩	🟩	🟩
OpenMP	🟩	🟩	🟥	🟥
HIP	🟩	🟩	🟥	🟥
OpenACC	🟩	🟥	🟥	🟥
CUDA	🟩	🟥	🟥	🟥

We don’t always know the what the next system will be.

Cloud platforms are delivering more GPU options

Rewriting code can be expensive

How proactive can we be ?

ROCm

Radeon **O**pen **Com**pute

https://rocm.github.io

Open Source platform for programming AMD and Nvidia GPUs

- OpenCL
- LLVM based C/C++ compilers (Branch off clang)
    - hcc
- GPU offloading wrapper
    - hipcc
- Accelerated Math and Deep Learning Libraries
- Multi-GPU support
    - Peer-to-peer multi-GPU operation with RDMA
- Profilers and Debuggers
- System monitoring tools
    - rocm-smi

ROCm

Radeon **O**pen **Com**pute

https://rocm.github.io

Debian and RPM repositories available

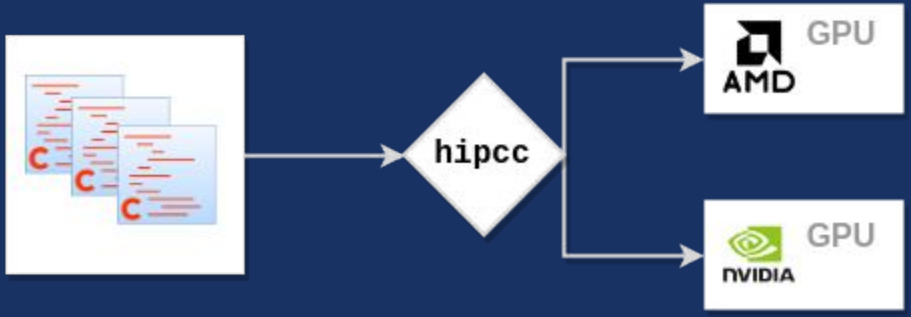
```
$ sudo {yum|apt-get} install rocm-dkms
```

HIP

Heterogeneous-compute **I**nterface for **P**ortability

https://github.com/ROCm-Developer-Tools/HIP

Interface for compiling C/C++ for AMD and Nvidia GPUs



FluidNumerics

HIP

API comparisons with CUDA

CUDA

```
cudaMalloc((void**)&nodes_dev, N*sizeof(float) );
```

HIP

```
hipMalloc((void**)&nodes_dev, N*sizeof(float) );
```

HIP

API comparisons with CUDA

CUDA

```
dim3 threadsPerBlock(nthreads,nthreads,nthreads);
dim3 blocks(n_elements);
GPUKernel<<<blocks,threadsPerBlock>>>( input );
```

HIP

```
dim3 threadsPerBlock(nthreads,nthreads,nthreads);
dim3 blocks(n_elements);
hipLaunchKernelGGL(GPUKernel, dim3(blocks), dim3(threadsPerBlock), 0, 0, input);
```

HIP

Compilation comparisons with CUDA

CUDA

```
$ nvcc source_code.cu
```

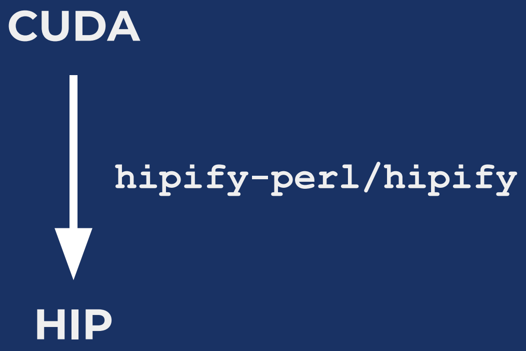
HIP

```
$ hipcc source_code.cu
```

FluidNumerics

HIP

HIPify existing CUDA Code



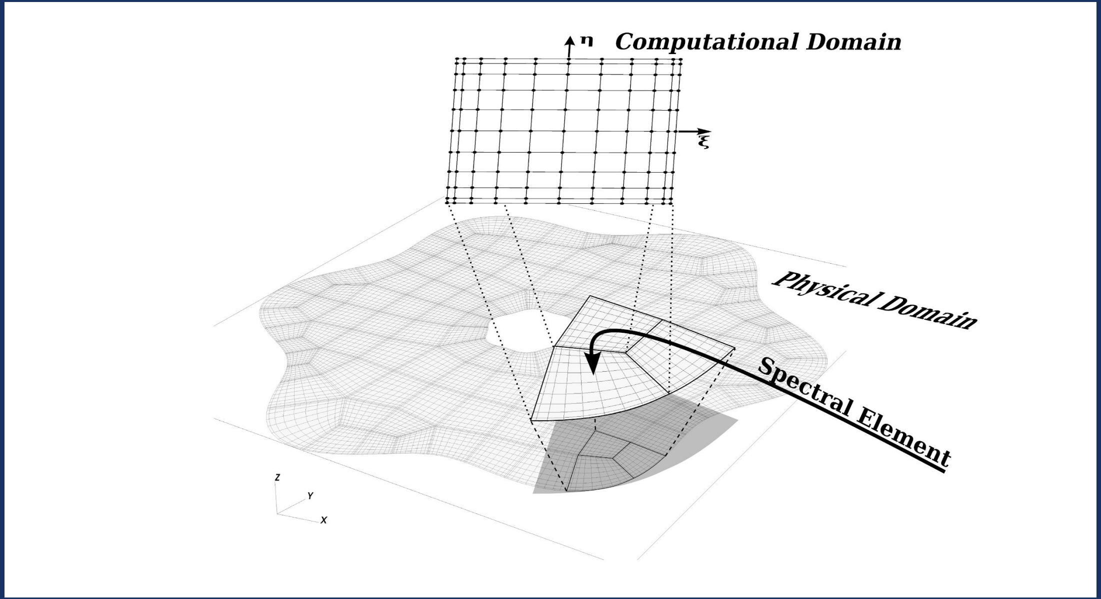
CUDA

`hipify-perl/hipify`

HIP

Spectral Element Divergence (3-D)

Cost & Performance Study



FluidNumerics

Spectral Element Divergence (3-D)

Cost & Performance Study

Serial Code

```
for( iel=0; iel<n_elements; iel++ ){
  for( k=0; k<N; k++ ){
    for( j=0; j<N; j++ ){
      for( i=0; i<N; i++ ){
        df[SP_3D_INDEX(i,j,k,iel,N)] = 0.0;
        for( ii=0; ii<=N; ii++){
          df[3D_INDEX(i,j,k,iel,N)]=df[3D_INDEX(i,j,k,iel,N)] +
            D[INDEX(ii,i,N)]*f[VEC_3D_INDEX(0,ii,j,k,iel,N)] +
            D[INDEX(ii,j,N)]*f[VEC_3D_INDEX(1,i,ii,k,iel,N)] +
            D[INDEX(ii,k,N)]*f[VEC_3D_INDEX(2,i,j,ii,iel,N)] ;

        }
      }
    }
  }
}
```

FluidNumerics

Spectral Element Divergence (3-D)

Cost & Performance Study

CUDA Kernel

```
int iel = blockidx.x;
int i   = threadidx.x;
int j   = threadidx.y;
int k   = threadidx.z;

df[SP_3D_INDEX(i,j,k,iel,N)] = 0.0;
for( ii=0; ii<=N; ii++){
  df[3D_INDEX(i,j,k,iel,N)]=df[3D_INDEX(i,j,k,iel,N)] +
    D[INDEX(ii,i,N)]*f[VEC_3D_INDEX(0,ii,j,k,iel,N)] +
    D[INDEX(ii,j,N)]*f[VEC_3D_INDEX(1,i,ii,k,iel,N)] +
    D[INDEX(ii,k,N)]*f[VEC_3D_INDEX(2,i,j,ii,iel,N)] ;
 }
```

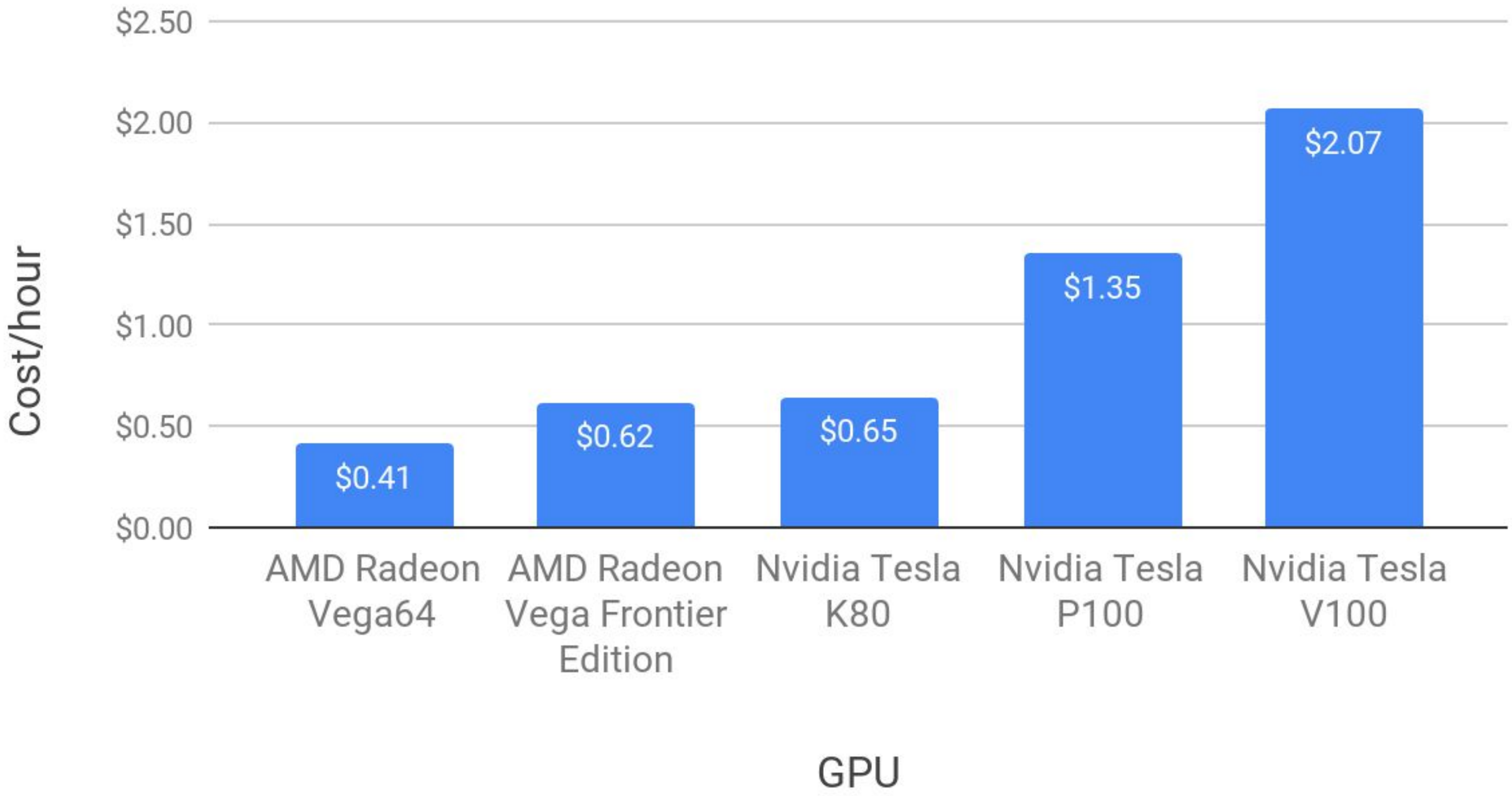
Spectral Element Divergence (3-D)

Cost & Performance Study

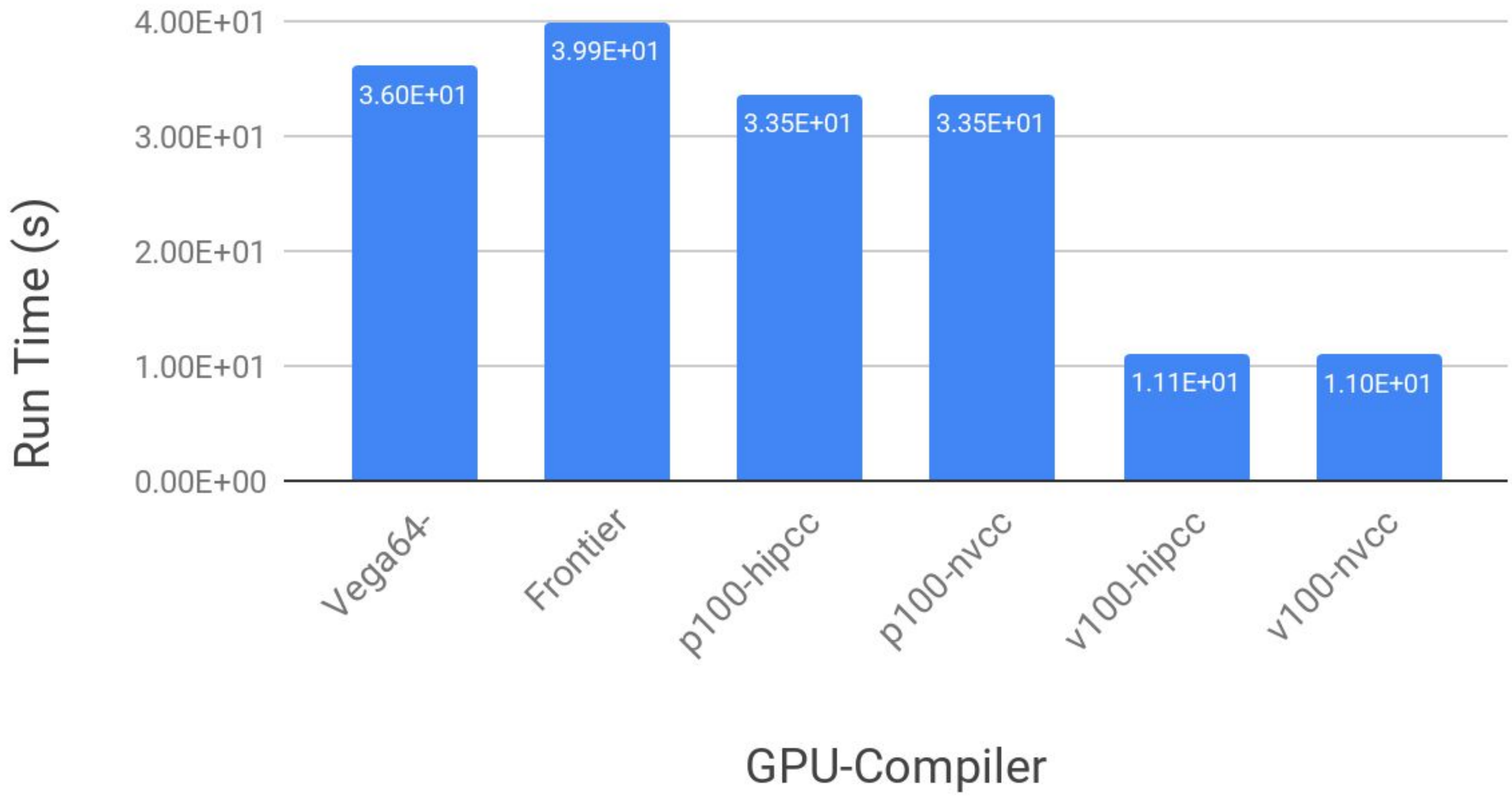
- API's Compared
  - CUDA - nvcc compiled
  - HIP - hipified CUDA, hipcc compiled
- Compute Platforms
  - Google Cloud Platform
    - Nvidia Tesla P100
    - Nvidia Tesla V100
  - GPUEater (gpueater.com)
    - Radeon Vega 64
    - Radeon Vega Frontier Edition

FluidNumerics

Cost/hour vs. GPU

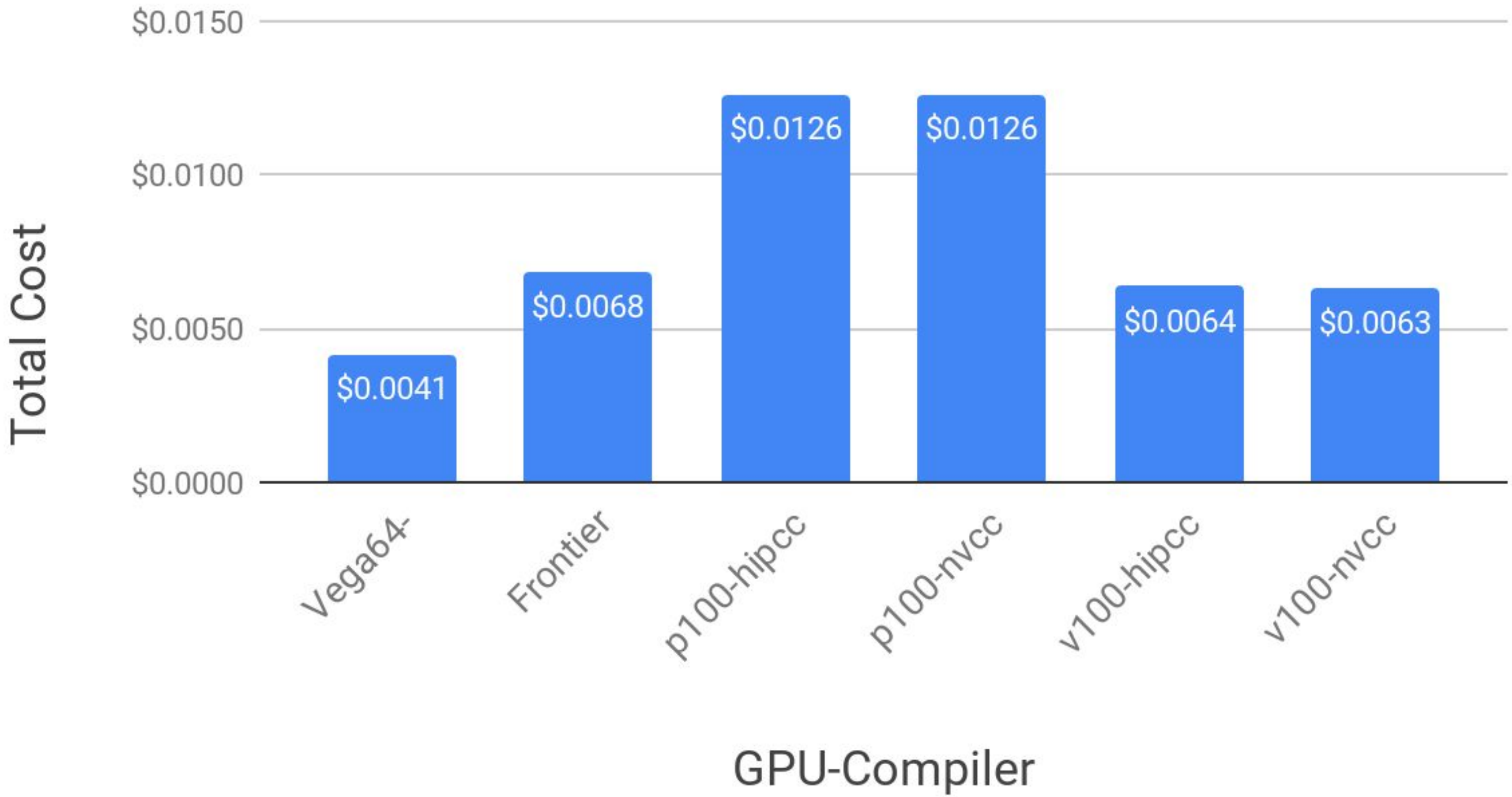


Divergence Runtimes : hipcc and nvcc



FluidNumerics

Cloud Costs : hipcc and nvcc



Spectral Element Divergence (3-D)

Cost & Performance Study

- **Total cost for cost & performance study was less than $5 in compute resources.**
- **HIP kernels give consistent performance with CUDA kernels on identical platforms**
- **V100's result in shortest time-to-solution**
- **Radeon Vega64 gives P100-like performance at the least expensive cost**
- **Hardware matters - Kernel performance varies across hardware**

FluidNumerics

Further Questions

Contact : joe@fluidnumerics.com

https://fluidnumerics.com