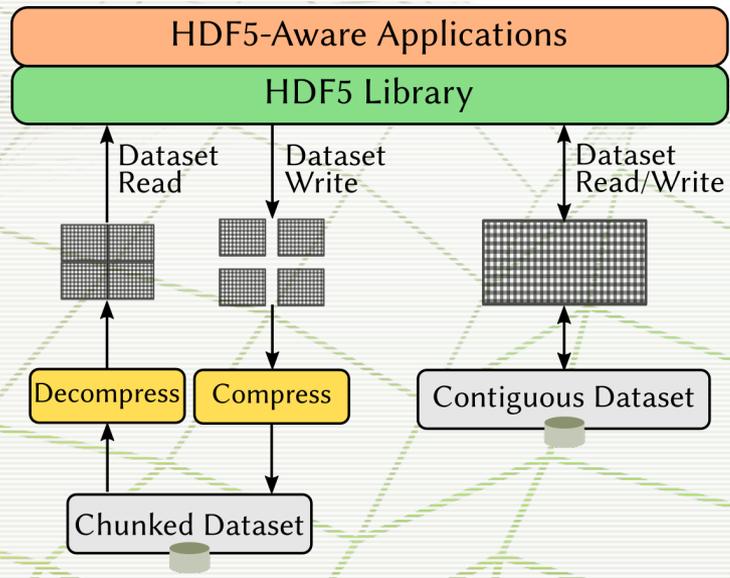




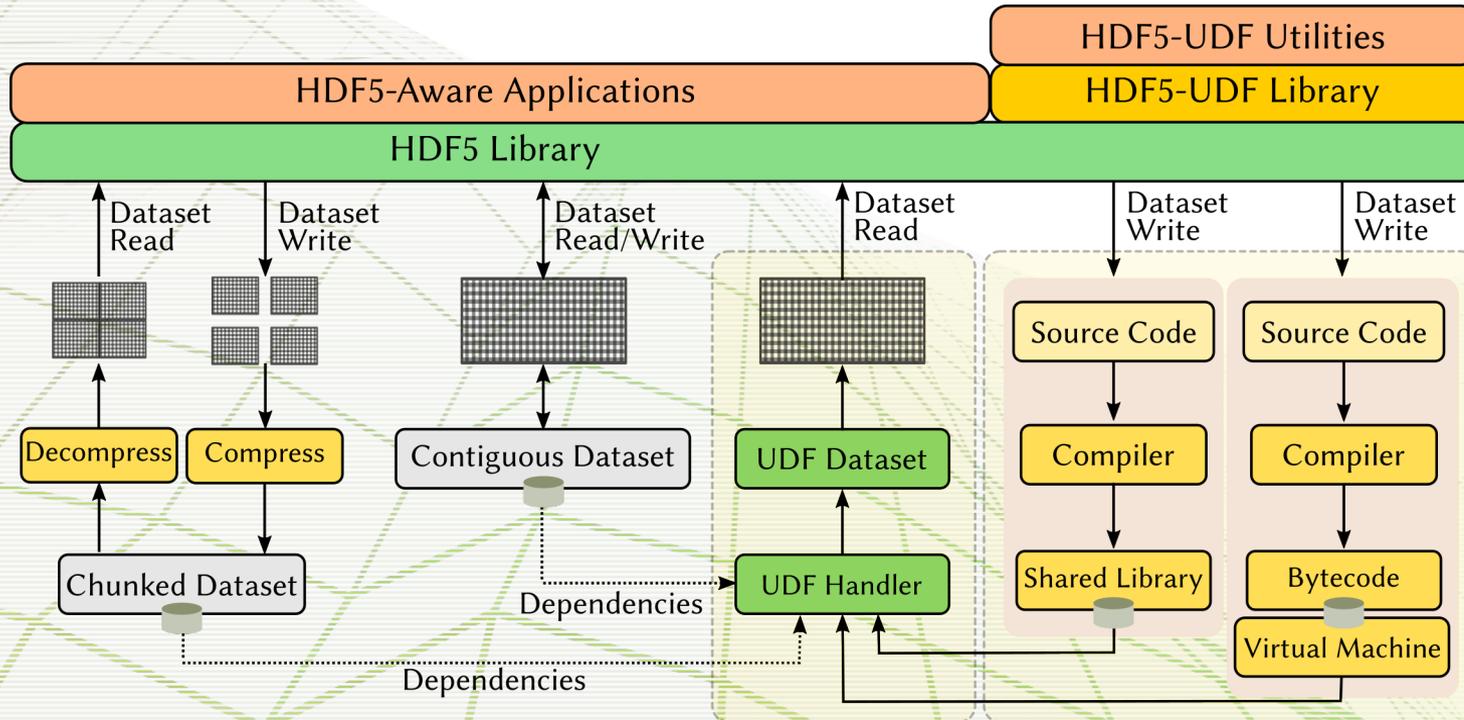
Computational Storage with HDF5-UDF

Lucas C. Villa Real
IBM Research
lucasvr@br.ibm.com

What is HDF5-UDF?



What is HDF5-UDF?



Many backends, simple API

```
1 def dynamic_dataset()  
2     a, b, c = lib.getData("A"), lib.getData("B"), lib.getData("C")  
3  
4     for i in range(lib.getDims("A")[0] * lib.getDims("A")[1]):  
5         c[i] = a[i] + b[i]
```

Python

Many backends, simple API

```
1 def dynamic_dataset()  
2     a, b, c = lib.getData("A"), lib.getData("B"), lib.getData("C")  
3  
4     for i in range(lib.getDims("A")[0] * lib.getDims("A")[1]):  
5         c[i] = a[i] + b[i]
```

Python

```
1 extern "C" void dynamic_dataset() {  
2     auto a = lib.getData<float>("A");  
3     auto b = lib.getData<float>("B");  
4     auto c = lib.getData<float>("C");  
5  
6     for (auto i=0, i < lib.getDims("A")[0] * lib.getDims("A")[1]; ++i)  
7         c[i] = a[i] + b[i];  
8 }
```

C++

Many backends, simple API

```
1 def dynamic_dataset()  
2     a, b, c = lib.getData("A"), lib.getData("B"), lib.getData("C")  
3  
4     for i in range(lib.getDims("A")[0] * lib.getDims("A")[1]):  
5         c[i] = a[i] + b[i]
```

Python

```
1 extern "C" void dynamic_dataset() {  
2     auto a = lib.getData<float>("A");  
3     auto b = lib.getData<float>("B");  
4     auto c = lib.getData<float>("C");  
5  
6     for (auto i=0, i < lib.getDims("A")[0] * lib.getDims("A")[1]; ++i)  
7         c[i] = a[i] + b[i];  
8 }
```

C++

```
1 function dynamic_dataset()  
2     local a = lib.getData("A")  
3     local b = lib.getData("B")  
4     local c = lib.getData("C")  
5  
6     for i=1, lib.getDims("A")[1] * lib.getDims("A")[2] do  
7         c[i] = a[i] + b[i]  
8     end  
9 end
```

Lua

Many backends, simple API

```
1 def dynamic_dataset()  
2     a, b, c = lib.getData("A"), lib.getData("B"), lib.getData("C")  
3  
4     for i in range(lib.getDims("A")[0] * lib.getDims("A")[1]):  
5         c[i] = a[i] + b[i]
```

Python

```
1 extern "C" void dynamic_dataset() {  
2     auto a = lib.getData<float>("A");  
3     auto b = lib.getData<float>("B");  
4     auto c = lib.getData<float>("C");  
5  
6     for (auto i=0, i < lib.getDims("A")[0] * lib.getDims("A")[1]; ++i)  
7         c[i] = a[i] + b[i];  
8 }
```

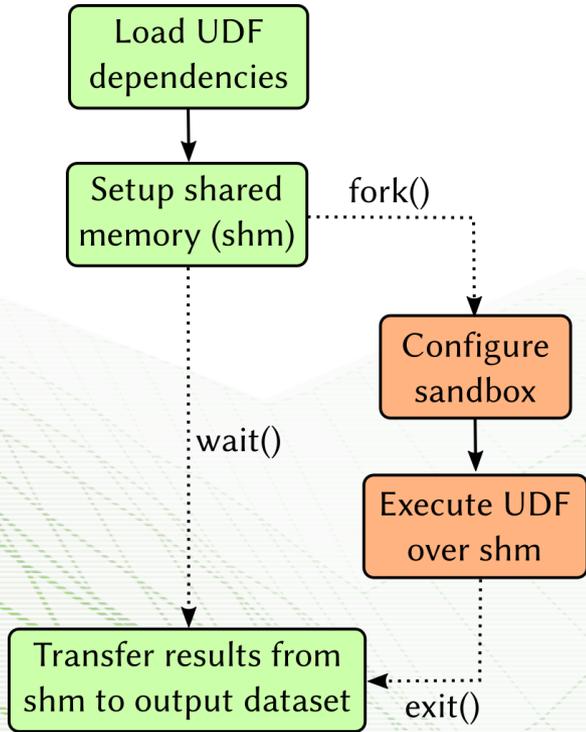
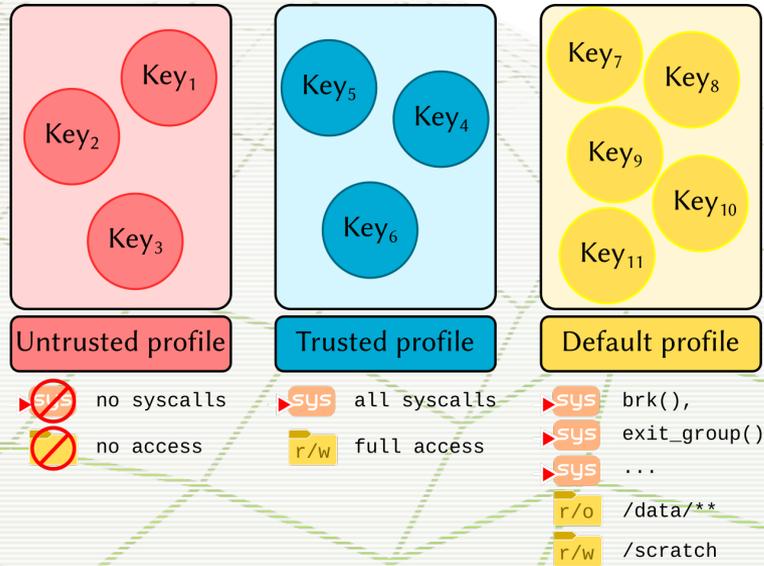
C++

```
1 function dynamic_dataset()  
2     local a = lib.getData("A")  
3     local b = lib.getData("B")  
4     local c = lib.getData("C")  
5  
6     for i=1, lib.getDims("A")[1] * lib.getDims("A")[2] do  
7         c[i] = a[i] + b[i]  
8     end  
9 end
```

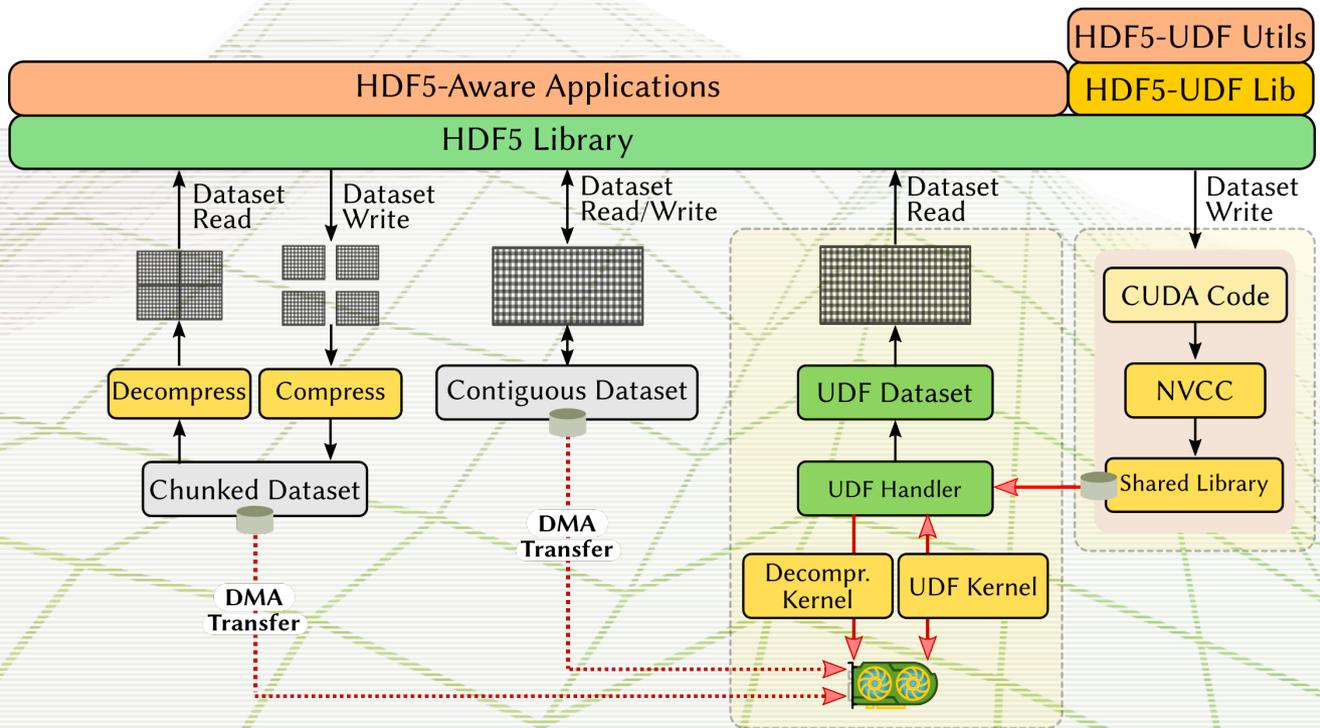
Lua

```
# hdf5-udf file.h5 udf.py C:1024x768:float
```

Built-in security model



New: Computational Storage with HDF5-UDF



```

__global__ void
kernel(int *red, int *nir, float *ndvi, size_t n)
{
    int idx = blockIdx.x * blockDim.x + threadIdx.x;
    if (i < n)
        ndvi[i] = (nir[i]-red[i]) / (nir[i]+red[i]);
}

extern "C" void dynamic_dataset()
{
    // Output dataset
    auto ndvi = lib.getData<float>("NDVI");
    auto dims = lib.getDims("NDVI");
    auto n = dims[0] * dims[1];

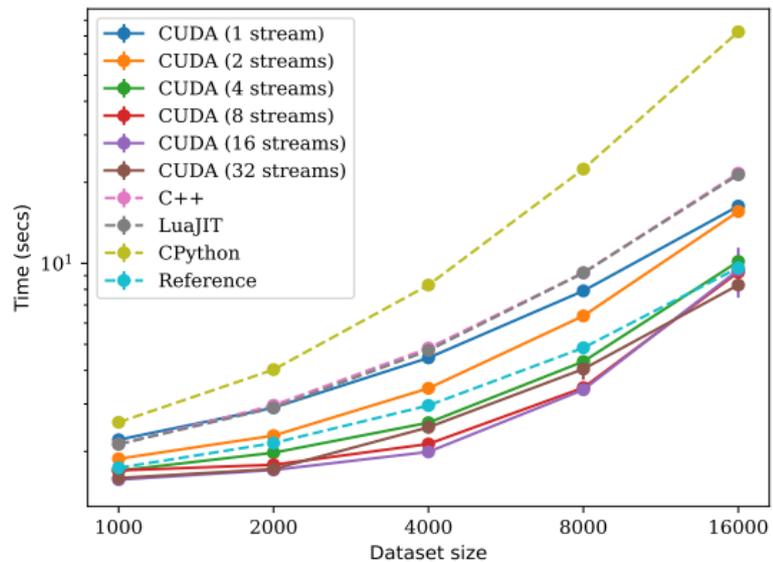
    // Input datasets
    auto red = lib.getData<int>("Red");
    auto nir = lib.getData<int>("NIR");

    // Configure and launch the kernel
    int block_dim = 1024;
    int grid_dim = ceil((float)(n*sizeof(int))/block_dim);

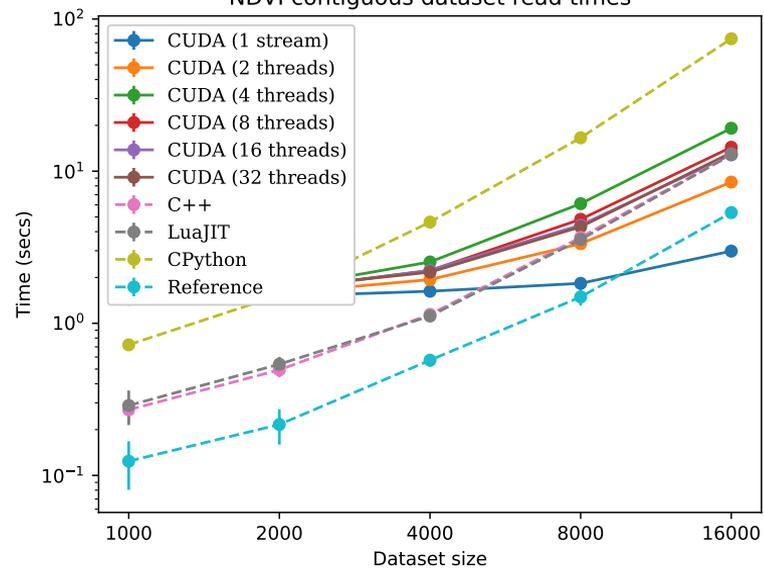
    kernel<<<grid_dim,block_dim>>>(red, nir, ndvi, n);
}
    
```

Performance numbers

NDVI chunked dataset read times



NDVI contiguous dataset read times



Have fun with HDF5-UDF!

Try it out at:

<https://github.com/lucasvr/hdf5-udf>

Check the examples at:

<https://github.com/lucasvr/user-defined-functions>