Feature # 1393 (Implemented): Redesign of GPUManager to utilize concurrent kernel execution and stream callbacks

**GPUManager API change**

03/03/2017 12:06 PM - Jaemin Choi

<table>
<thead>
<tr>
<th>Status:</th>
<th>Feedback</th>
<th>Start date:</th>
<th>03/03/2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td>Jaemin Choi</td>
<td>% Done:</td>
<td>0%</td>
</tr>
<tr>
<td>Category:</td>
<td>GPU Support</td>
<td>Estimated time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Target version:</td>
<td>6.8.1</td>
<td>Spent time:</td>
<td>0.00 hour</td>
</tr>
</tbody>
</table>

**Description**

Making changes to current GPUManager API to provide a more uniform & segregated API (function calls now start with hapi_) and better usability to the user.

This also eliminates memory leaks arising from creating a workRequest itself and data structures inside it.

**[API comparison]**

**Old**

```c
workRequest *matmul = new workRequest;
matmul->dimGrid = dim3(ceil((float)matrixSize / BLOCK_SIZE), ceil((float)matrixSize / BLOCK_SIZE));
matmul->dimBlock = dim3(BLOCK_SIZE, BLOCK_SIZE);
matmul->smemSize = 0;
matmul->nBuffers = 3;
matmul->bufferInfo = new DataInfo[matmul->nBuffers];
AInfo = &(matmul->bufferInfo[0]);
AInfo->transferToDevice = YES;
AInfo->transferFromDevice = NO;
AInfo->freeBuffer = YES;
AInfo->hostBuffer = h_A;
AInfo->size = size;
BInfo = &(matmul->bufferInfo[1]);
BInfo->transferToDevice = YES;
BInfo->transferFromDevice = NO;
BInfo->freeBuffer = YES;
BInfo->hostBuffer = h_B;
BInfo->size = size;
CInfo = &(matmul->bufferInfo[2]);
CInfo->transferToDevice = NO;
CInfo->transferFromDevice = YES;
CInfo->freeBuffer = YES;
CInfo->hostBuffer = h_C;
CInfo->size = size;
matmul->callbackFn = cb;
if (useCublas) {
    matmul->traceName = "blas";
    matmul->runkernel = run_BLAS_KERNEL;
}
else {
    matmul->traceName = "matmul";
    matmul->runkernel = run_MATMUL_KERNEL;
}
matmul->userData = new int(matrixSize);
enqueue(matmul);
```

03/31/2017 1/2
New

```c
workRequest *matmul = hapi_createWorkRequest();
dim3 dimGrid(ceil((float)matrixSize / BLOCK_SIZE), ceil((float)matrixSize / BLOCK_SIZE));
matmul->setExecParams(dimGrid, dimBlock);
matmul->addBufferInfo(-1, h_A, size, cudaMemcpyHostToDevice, 1);
matmul->addBufferInfo(-1, h_B, size, cudaMemcpyHostToDevice, 1);
matmul->addBufferInfo(-1, h_C, size, cudaMemcpyDeviceToHost, 1);
matmul->setCallback(cb);
if (useCublas) {
    matmul->setTraceName("blas");
    matmul->setRunKernel(run_BLAS_KERNEL);
} else {
    matmul->setTraceName("matmul");
    matmul->setRunKernel(run_MATMUL_KERNEL);
}
matmul->setUserData(&matrixSize, sizeof(int));
hapi_enqueue(matmul);
```

Subtasks:
Support #1456: Add more stream callbacks for use after HTod transfer and kernel execution

History

#1 - 03/03/2017 12:48 PM - Jaemin Choi
- Status changed from In Progress to Feedback

Change pushed to gerrit for review.
[https://charm.cs.illinois.edu/gerrit/#/c/2283/]

#2 - 03/03/2017 02:34 PM - Michael Robson
Buffer ID (-1) should be last param and set to -1 by default
Also, is there a way to mark copy both ways?
ints should be change bools