An Approach for Large-scale Distributed FFT Framework on GPUs

Yichang Hu[1], BinBin Zhou[1] and Lu Lu[2]


Abstract & Introduction
Multi-nodes heterogeneous systems aspire to meet the increasing demands from parallel computing FFT and we present a highly efficient GPU-based distributed FFT framework by adapting the Cooley-Tukey recursive FFT algorithm in this work.

Typical approaches used by parallel libraries are pencil and slab decomposition. Our work employs slab decomposition (Fig.1) and consists of four steps, batched 2D FFTs, all-to-all communication (Fig.2), local transpose, batched 1D FFTs.

Design and methodology
Dynamic low-dimensional FFT kernels generation.
- Group small sequence butterflies.
- Non-strided data access mode.
- Runtime compilation by hipRTC APIs.

Hybrid OpenMP-MPI programming model.
- Multiple threads for intra-node communication to control multiple devices.
- GPU-aware MPI for inter-node communication to avoid the high latency through CPU.
- Efficient asynchronous management of packing and communication (shown in Fig.4, which demonstrates the classical packing process and a novel approach in our design, overlapping packing/unpacking with MPI communication). Specific high performance transpose kernel.
- A 3D point on (i,j,k) can be mapped into the (iXY+Z+jX+Z+k)th element, and the destination location after transposition is (D is the number of devices):

\[ X \times Y \times D \times Z \times (j \times Y \div D) + i \times H \times X + j \mod Y \div D \times X + k \]

Evaluation
The performance of our work is compared with heFFTe[1] and FFTW on a heterogeneous system, which are the state-of-art distributed FFT libraries. Fig. 5 is case study on AMD MI100 and MI250 GPUs, and scalability comparison is in Table 2. It is clear that our implementations consistently outperform heffte, while not compromising on communication overhead.

Table 2 Scalability on MI250 (GFlops)

<table>
<thead>
<tr>
<th>GPUs</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>498</td>
<td>818</td>
<td>1538</td>
<td>2581</td>
<td>4312</td>
<td>6942</td>
</tr>
<tr>
<td>heFFTe</td>
<td>160</td>
<td>320</td>
<td>591</td>
<td>1200</td>
<td>2010</td>
<td>3694</td>
</tr>
</tbody>
</table>

Conclusion
This work proposes an apporach for large-scale distributed FFT framework on GPUs that makes following three primary novel contributions:
- Novel data reshuffling and memory transfer pattern.
- Asynchronous management for packing and communication.
- Unit-stride data layout and specific transposne kernel for high efficient memory operations.

Our work achieves more than 40x speedup over CPU FFT libraries and about 2x speedup over heFFTe.

References