GEMM SUP Optimizations

Mithun Mohan K M, Nallani Bhaskar

Contents

Introduction

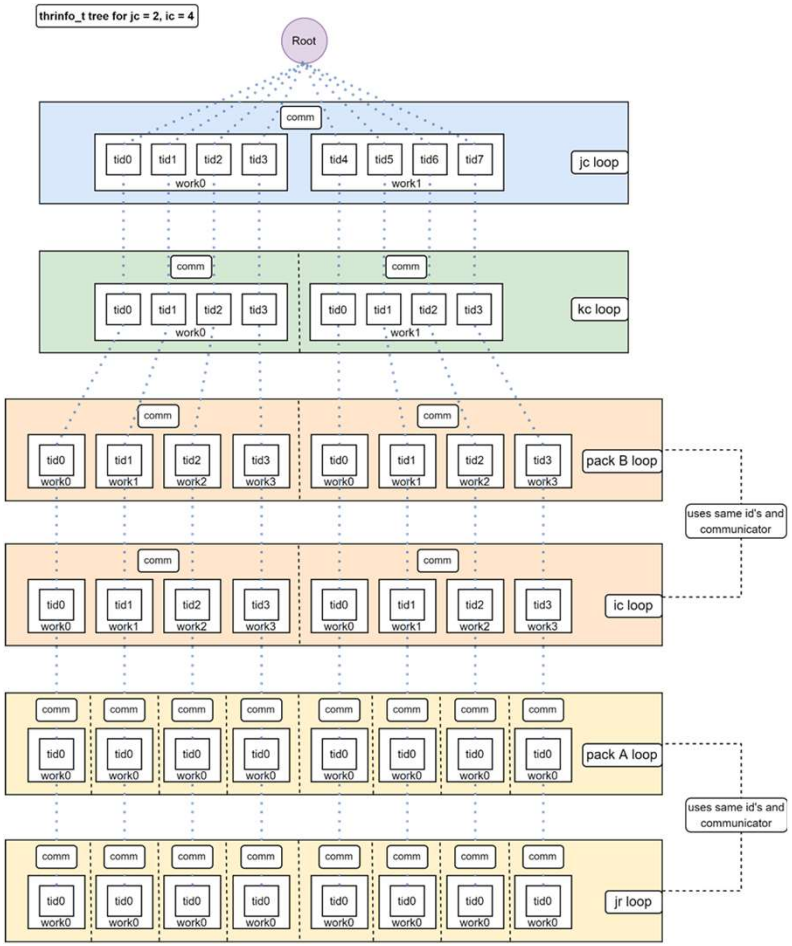
- AOCL (AMD optimizing CPU Libraries) is AMD's CPU Math Library tuned for AMD EPYC™ processor family.
  - AOCL-BLIS is a fork of BLIS library optimized as part of AOCL.

- Multi-Thread GEMM performance was not optimal:
  - SUP performance, especially for smaller matrices, was not scaling well with greater number of threads.
  - In SGEMM native path, packing costs were not amortized when per thread matrix blocks were small, thereby decreasing overall performance.
    - Packing costs consists of cost of packing the matrix + cross thread synchronization costs (required for memory allocation, pack buffer propagation).

- Solution was two phased:
  - Identification of scaling bottlenecks in SUP path and removal of the same.
  - Native vs SUP path selection based on per thread matrix blocks.

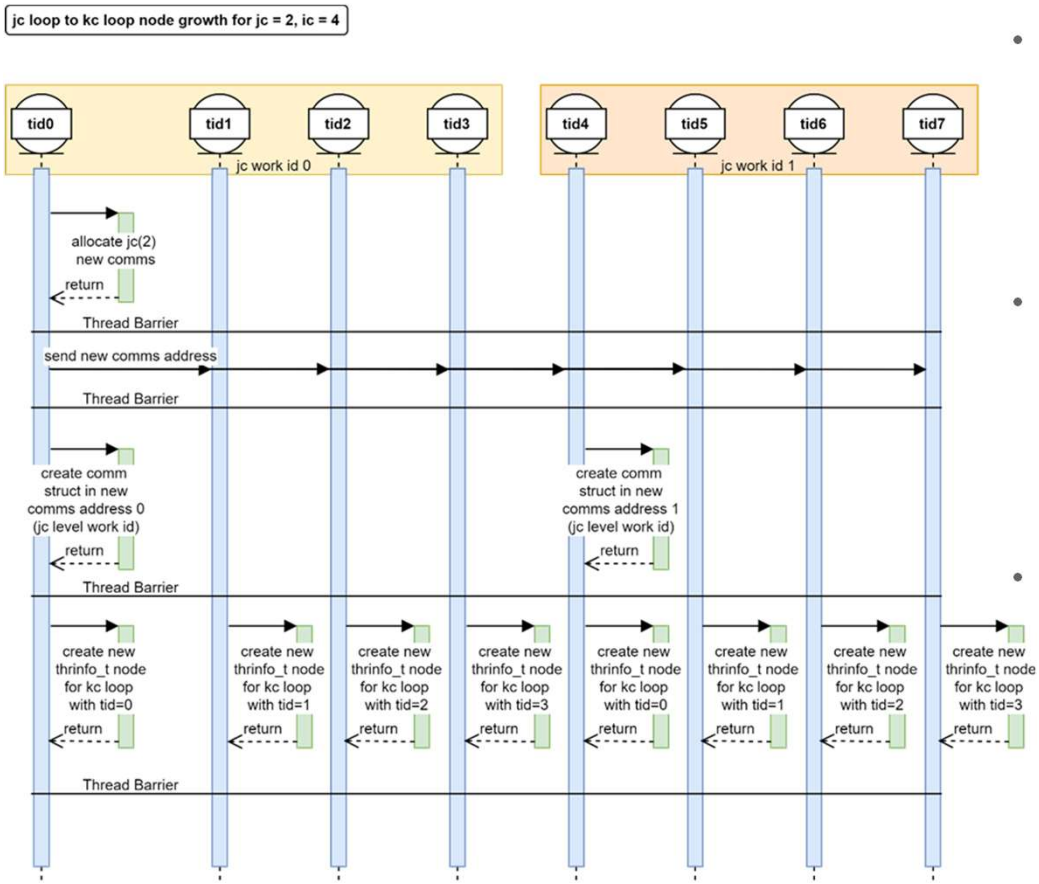
- Multi-Thread GEMM performance improved greatly for SUP inputs with smaller k dimension.

SUP thrinfo_t tree - Overview



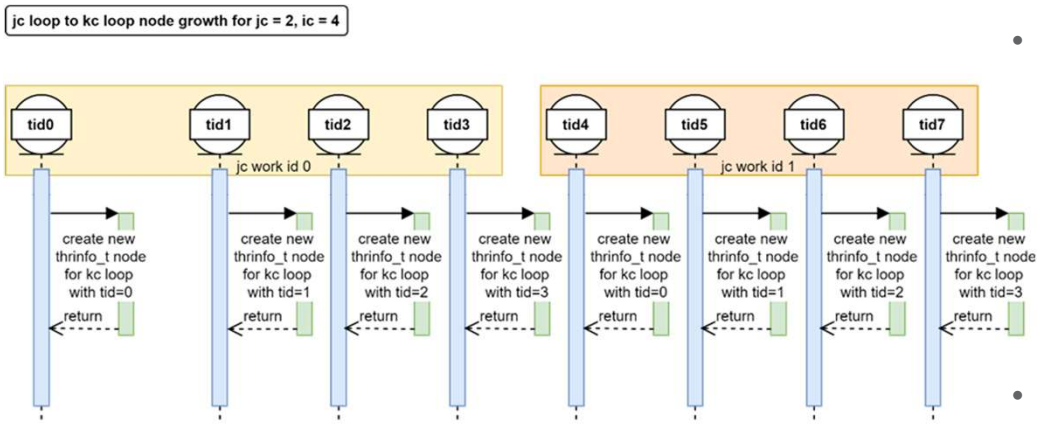
- thrinfo_t tree used to facilitate division of work among threads in multi-thread gemm.
  - 1-ary graph with multiple levels to represent either a loop of 5 loop algorithm or A/B packing.
  - Each level contains communicator for inter-thread communication.
    - Pack buffer broadcast.
    - Barriers for thread synchronization.
- Threads use tree node attributes to identify its work group and sub block/panel of matrix to work on.

SUP thrinfo_t tree - Creation



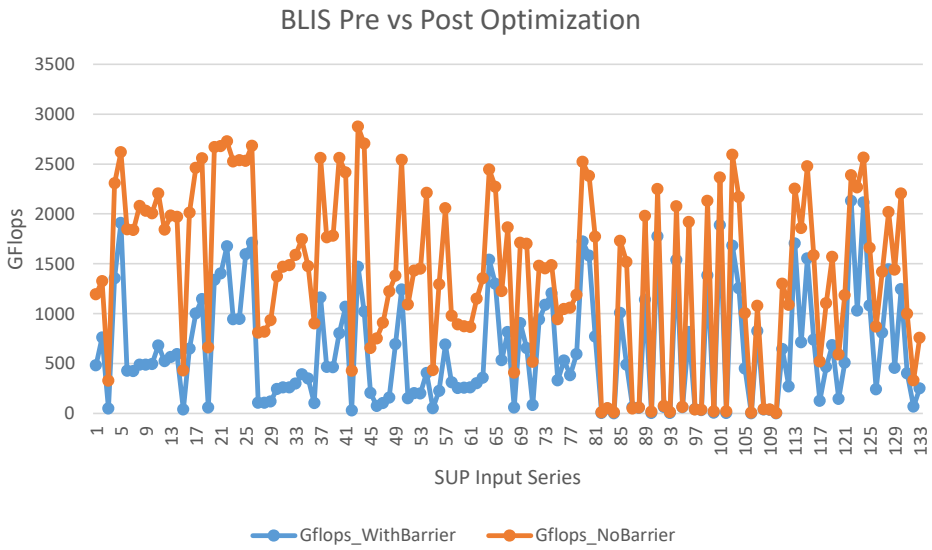
- Nodes/levels in thrinfo_t tree created in iterative fashion.
  - New level (child) created as algorithm progress through the different loops for the first time.
- Chief nodes (threads) of a level responsible for thread safe actions:
  - Memory allocation for communicators.
  - Broadcasting the communicator address.
- Barriers used for thread synchronization during communicator creation and broadcast.

SUP thrinfo_t tree optimization - Barrier removal



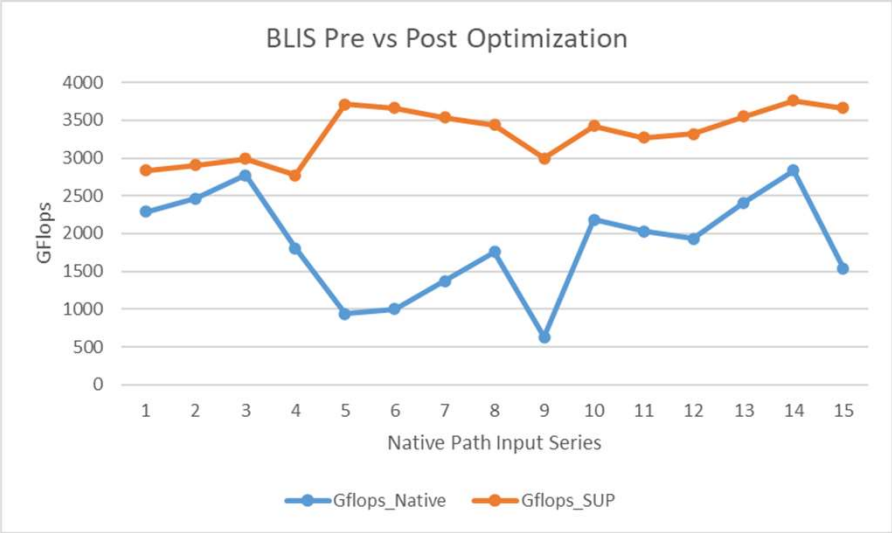
- Communicators only required when optional packing (A or B) is enabled in SUP path.
  - Pack buffer allocation by chief thread and broadcast among threads in work group.
  - Thread synchronization using barriers for thread safe packing within a work group.
- Communicator creation and associated barriers can be avoided if packing is not enabled.
  - thrinfo_t tree creation is faster and avoids multi-thread bottleneck at barriers.

SUP thrinfo_t tree optimization - Impact



- sgemm SUP Gflops doubled for smaller matrix sizes.
  - sgemmt also improved as part of it.
- Performance improvements observed for gemm across data types.

Moving native path inputs to SUP path



Example								
m	n	k	Gflops_Native	ic	jc	m_ic	n_jc	Gflops_SUP
1024	1024	845	2291.95	4	16	256	64	2836.57
1152	1024	845	2468.51	8	8	144	128	2909.5
4096	256	512	1759.13	32	2	128	128	3439.39
Original			**MT = 512**	**NT = 200**	**KT = 240**			

- Native or SUP path taken based on input dimensions.
  - Does not consider per-thread dimensions in multi-thread gemm.
    - Within SUP limits even when input dimensions are above it.
- Packing A and B matrix results in overhead when per thread dimensions are small.
- Modifying existing path selection logic to a per-thread dimension-based logic.
  - Per thread SUP limits observed to be less than original SUP limits.

B matrix packing in SUP path

Example								
m	n	k	Gflops_Native	ic	jc	m_ic	n_jc	Gflops_SUP
4096	1024	845	3365.56	8	8	512	128	3970.67
8192	1024	845	3746.91	32	2	256	512	3873.37
8192	512	1024	3464.17	16	4	512	128	3895.92
Original			**MT = 512**	**NT = 200**	**KT = 240**			

- B packing in SUP path results in performance improvement when per thread dimensions are sufficiently large.
  - B packing is multi-threaded with ic threads per B panel.
- B packing gains offsets the thread synchronization costs.

questions?

COPYRIGHT AND DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate releases, for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.


AMD