OSPP 2025

Project ID:25b970403

Triton-CPU tile operation on AArch64 SVE/SME

Daniel Antonio Martínez Sánchez
(danielantonio.martinezs@um.es)

30/9/2025

Contents

1 Project Information

1.1 Solution description

For enabling SVE/SME extensions on triton-cpu for open-euler I used the triton-shared open source project to help translate from triton IR to MLIR. I needed to convert to MLIR because I was also helped by the Huawei Team at Cambridge that provided SVE/SME optimization pipelines that worked on MLIR but not on Triton IR. Still I had to integrate and improve triton-shared to fit the already existing triton-cpu for open-euler.

1.2 Time planning

The time planning was altered as the project took a different approach as we collaborated with the team at Cambridge, all the details about the planning are documented on the gitee issues but I will make a brief summary here

- Weeks 1-2: Started integrating triton-shared into triton-cpu
- Weeks 3-4: Fully integrated triton-shared and started working on SVE/SME
- Weeks 5-6: Added SVE/SME optimization pipelines and started working on passing all test with this new backend
- Weeks 7-10: Fixed many issues with the tests adding new functionality.
- Week 11-14: Improved SVE pipeline with the help of Cambridge team and prepared project handover.

2 Project Progress

2.1 Work Completed

Even if by a different approach than the firts one proposed with the project, SVE/SME is now available on triton-cpu, it still has some issues to solve as there are few failing test remaining plus some work to do in the SVE pipeline as it could not be implemented before since the Cambridge team had to open source it.

2.2 Problems encountered and solutions

This was a long and tedious project in which I faced many challenges there is extensive documentation of this project in this link but I will again describe each one of them briefly here.

I had to see how to integrate triton-shared into the existing triton-cpu for this I used the triton plugins functionality and after fixing some compilation errors and version incompatibilities between the three projects (llvm, triton-cpu, triton-shared) I was able to compile successfully. However, I did find a bug where every triton program was crashing but I ended up solving it by fixing the driver of the triton-shared backend.

Then I started adding experimental support for SVE/SME looking at the official llvm examples, along the way I had to fix a bug with SVE/SME in the transform-schedule of MLIR as the transform-schedule is the main mechanism used for performance optimizations.

Running on both SVE and SME proved challenging as I found some more bugs along the way, all of them are reference on the link mentioned before. Here I was starting to get comfortable with the work environment and in the upcoming weeks my productivity kept improving.

Later, I started working on passing as much tests as possible and this was the point the project was more focused on, I had to add support for a lot of new ops like `tt.scan` or `tt.atomic_rmw`, to new datatypes as fp8e5m2 and fp8e4m3 and also found some more bugs and issues along the way in the llvm I was using.

All of this bug fixing and additions really sky-rocketed my technical knowledge of the LLVM/MLIR infrastructure, I spent countless hours debugging that are now experience and I believe there is a tremendous gap between my knowledge at the start of the project and my knowledge now and I'm really glad I was able to improve that much. I also learned how to communicate better in a team full of talented and knowledge people and also how to interact in the open source community which are key abilities to be able to keep improving.

As said I found a lot of bugs and had to add a lot of things so here is a list of all of my PR's to the open-euler llvm:

- https://gitee.com/openeuler/llvm-project/pulls/236
- https://gitee.com/openeuler/llvm-project/pulls/288
- https://gitee.com/openeuler/llvm-project/pulls/243
- https://gitee.com/openeuler/llvm-project/pulls/266
- https://gitee.com/openeuler/llvm-project/pulls/269
- https://gitee.com/openeuler/llvm-project/pulls/255
- https://gitee.com/openeuler/llvm-project/pulls/290
- https://gitee.com/openeuler/llvm-project/pulls/291
- https://gitee.com/openeuler/llvm-project/pulls/293

Also it may be worth to take a look at these two pieces of documentation, one contains the reproduction instructions for the project and the mentioned project log where everything I did is explained:

- Project log
- Reproduction instructions

2.3 Next steps

As mentioned earlier there are still somethings to fix around in the recently added SVE pipeline and some tests fails to fix, on top of that a new benchmark evaluation should be done with the newer optimized pipelines. With the help of the mentors and the Cambridge team we have made the smoothest handover possible. As for me even now that the OSPP is over, I hope I can keep contributing to llvm and the open source compiler community in the near future.