

# Hands-on: Understanding data locality in NUMA via STREAM

---

- Choose either C/C++ (`c` and `ctrad`) or Fortran (`fortran`). Both of them are fine, as well.

## C/C++

### How to execute

#### 1. Edit a job script

- Before trying this hands-on, you need to do `00_stream` to compile STREAM. Moreover, we suggest that you confirm the results in `01_bandwidth`.
- We have two working directories, `c/fj_zfill.default` and `c/fj_zfill.pdp`. You can find:
  - `run.sh`: a script to execute STREAM
  - `task.sh`: a job script to run STREAM with different kinds of settings
- Edit `BINDIR` variable in `run.sh` before the execution. You need to write your installed location of STREAM binary (e.g., `stream.exe`) there.

#### 2. Run program

- You can run the program either:

```
## We show the case of using fj_zfill.pdp
## To run as a batch job
$ cd c/fj_zfill.pdp
$ pjsub task.sh
## Or, to run in an interactive job
$ cd c/fj_zfill.pdp
$ bash task.sh
```

- Each of the cases in the Exercises will be completed within 3-4 minutes.
  - For safety, we set the job elapsed time in the job scripts is 6 minutes.

### Exercises A

- E1: Check whether `stream.c` is satisfied with first-touch policy in Linux or not.
- E2: Try `c/fj_zfill.pdp`. You can find that the paging policy in A64FX is set by `prepage:demand:prepage`. Please check `XOS_MMM_L_PAGING_POLICY` in `run.sh`. How do the behaviors of memory bandwidth change from those in `01_bandwidth/c/fj_zfill`? How is the paging policy set in `01_bandwidth`? In particular, check the cases of `thread=12` and `thread=48`.
- E3: Try `c/fj_zfill.default`. You can find there is no explicit setting of the paging policy in `run.sh`. Are the behaviors of the memory bandwidth similar to `01_bandwidth/c/fj_zfill`?

### Exercises B

- E4: Change `stream.c` so that the first-touch policy **is not** satisfied. Then, compare the results for Triad to those in `01_bandwidth`.
  - A simple way is to comment out all `#pragma omp parallel for` before measuring Triad.

## Fortran

### How to execute

#### 1. Edit a job script

- Before trying this hands-on, you need to do `00_stream` to compile STREAM. Moreover, we suggest that you confirm the results in `01_bandwidth`.
- We have two working directories, `fortran/fj_zfill.default` and `fortran/fj_zfill.pdp`. You can find:
  - `run.sh`: a script to execute STREAM
  - `task.sh`: a job script to run STREAM with different kinds of settings
- Edit `BINDIR` variable in `run.sh` before the execution. You need to write your installed location of STREAM binary (e.g., `stream.exe`) there.

#### 2. Run program

- You can run the program either:

```
## We show the case of using fj_zfill.pdp
## To run as a batch job
$ cd fortran/fj_zfill.pdp
$ pjsub task.sh
## Or, to run in an interactive job
$ cd fortran/fj_zfill.pdp
$ bash task.sh
```

- Each of the cases in the Exercises will be completed within 3-4 minutes.
  - For safety, we set the job elapsed time in the job scripts is 6 minutes.

### Exercises A

- E1: Check whether `stream.f` is satisfied with first-touch policy in Linux or not.
- E2: Try `fortran/fj_zfill.pdp`. You can find that the paging policy in A64FX is set by `prepage:demand:prepage`. Please check `XOS_MMM_L_PAGING_POLICY` in `run.sh`. How do the behaviors of memory bandwidth change from those in `01_bandwidth/fortran/fj_zfill`? How is the paging policy set in `01_bandwidth`? In particular, check the cases of `thread=12` and `thread=48`.
- E3: Try `fortran/fj_zfill.default`. You can find there is no explicit setting of the paging policy in `run.sh`. Are the behaviors of the memory bandwidth similar to `01_bandwidth/fortran/fj_zfill`?

### Exercises B

- E4: Change `stream.f` so that the first-touch policy **is not** satisfied. Then, compare the results for Triad to those in `01_bandwidth`.

- A simple way is to comment out all `!$OMP PARALLEL DO` before measuring Triad.