

NEC SX-Aurora Tsubasa: Instrukcja

Ostatnia aktualizacja: 2 kwietnia 2020 by M. Hermanowicz <m.hermanowicz@icm.edu.pl>

Dokument opisuje podstawy użytkowania komputera wektorowego NEC SX-Aurora Tsubasa, dostępnego w centrum obliczeniowym ICM UW. Treść jest oparta na cytowanej w tekście dokumentacji i ma na celu zebranie podstawowych informacji i ich źródeł na potrzeby użytkowników ICM.

Spis treści

| | | |
|----------|---|----------|
| 1 | Wprowadzenie | 1 |
| 2 | Podstawy użytkowania | 1 |
| 3 | SOL: Transparent Neural Network Acceleration | 3 |

1 Wprowadzenie

NEC SX-Aurora Tsubasa to zapowiedziany w 2017 roku procesor wektorowy (ang. *vector engine*, VE) o architekturze należącej do serii SX – rozwijanej przez firmę NEC od połowy lat osiemdziesiątych [1]. W przeciwieństwie do swoich poprzedników, urządzenie nie jest samodzielną jednostką obliczeniową, ale zostało zaprojektowane jako karta PCIe, która pracuje w ramach i pod kontrolą serwera o standardowej architekturze x86_64 (ang. *vector host*, VH). Host jest zarządzany przez system operacyjny GNU/Linux i dostarcza kompletnego środowiska programistycznego niezbędnego do pracy z podłączonymi kartami VE. Jednym z kluczowych komponentów oprogramowania jest VEOS (ang. *Vector Engine Operating System*), który pracując po stronie VH pełni funkcję systemu operacyjnego dla programów wykonywanych na VE [2].

ICM UW dostarcza użytkownikom instalację NEC SX-Aurora Tsubasa jako część klastra obliczeniowego Rysy (partycja `ve`) – patrz Tabela 1.

| | Vector Host (VH) | Vector Engine (VE) |
|-----------------|----------------------|------------------------------|
| Model CPU | Intel Xeon Gold 6126 | NEC SX-Aurora Tsubasa A300-8 |
| Rdzenie CPU | 2 × 12 | 8 × 8 |
| Pamięć RAM [GB] | 192 | 8 × 48 |

Tabela 1: NEC SX-Aurora Tsubasa w ICM UW: partycja `ve` klastra Rysy

2 Podstawy użytkowania

Użycie komputera NEC wymaga zalogowania się do klastra Rysy przez SSH [3] za pośrednictwem systemu dostępowego `hpc.icm.edu.pl`. Można wykonać tę czynność w dwóch krokach, jak pokazano na Listingu 1, lub w jednym: wywołując instrukcję `ssh` z opcją `-J` (ang. *jump host*), która pozwala określić host pośredniczący w połączeniu z systemem docelowym (szczegóły: `man ssh`).

```
$ ssh username@hpc.icm.edu.pl
$ ssh rysy
```

Listing 1: Uzyskiwanie dostępu do systemu NEC SX-Aurora Tsubasa – klastr Rysy, ICM UW

Obciążeniem klastra i zadaniami użytkowników zarządza Slurm [4], natomiast oprogramowanie zorganizowane jest w formie modułów [5]. Pojedynczy węzeł obliczeniowy partycji `ve` (PBaran) może być używany interaktywnie – patrz Listing 2 – lub wsadowo (patrz kolejne akapity).

```
$ srun -A GRANT_ID -p ve --gres=ve:1 --pty bash -l
```

Listing 2: Uruchamianie sesji interaktywnej Slurm na klastrze Rysy

| Opcja | Opis |
|---------------------|---|
| -c | tworzy plik obiektowy |
| -o | nazwa pliku wyjściowego |
| -I/path/to/include | włączenie plików nagłówkowych |
| -L/path/to/lib | włączenie bibliotek |
| -g | symbole dla debuggera |
| -Wall | ostrzeżenia o składni |
| -Werror | traktuj ostrzeżenia jako błędy |
| -O[0-4] | poziomy optymalizacji |
| -ftrace | użycie <i>profilera</i> |
| -proginf | włącz analizę wykonania programu |
| -report-all | raportuj diagnostykę |
| -traceback | włącz informacje zwrotne nt. wykonania programu |
| -fdiag-vector=[0-3] | poziom szczegółów diagnostyki wektorowej |

Tabela 2: Podstawowe opcje kompilatorów NEC

Po uruchomieniu sesji interaktywnej zostaje automatycznie zdefiniowana nowa zmienna środowiskowa `$VE_NODE_NUMBER`, która określa która z kart VE zostanie użyta przez oprogramowanie użytkownika. Wartość tej zmiennej można sprawdzić i zdefiniować manualnie za pomocą instrukcji, odpowiednio, `echo` [6], `export` [7]. Oprogramowanie służące obsłudze kart VE znajduje się w katalogu `/opt/nec/ve`. Jego efektywne wykorzystanie wymaga modyfikacji niektórych zmiennych środowiskowych [8], takich jak `$PATH`, `$LD_LIBRARY_PATH` i innych, czego można w wygodny sposób dokonać za pomocą instrukcji `source` [9]:

```
$ source /opt/nec/ve/mpi/2.2.0/bin/necmpivars.sh
```

Listing 3: Modyfikacja zmiennych środowiskowych, potrzebnych do pracy z VE

Ustawienie zmiennych środowiskowych (Listing 3) sprawia, że narzędzia VE stają się dostępne dla użytkownika – wśród nich kompilatory NEC języków C, C++ i Fortran, które można wywołać instrukcjami, odpowiednio, `ncc`, `nc++` i `nfort` lub ich odpowiednimi wariantami MPI: `mpincc`, `mpinc++` i `mpinfort`. Należy pamiętać, że w systemie dostępnych jest kilka wersji kompilatorów, dlatego zająć trzeba uwzględnienia numeru wersji przy ich wywołaniu, np. `ncc-2.5.1`. Ogólny schemat użycia kompilatorów NEC jest zgodny ze standardem znanym z GNU GCC: `<kompilator> <opcje> <plik Źródłowy>`. Tabela 2 listuje zbiór kilkunastu podstawowych opcji kompilatorów NEC. Ostatnie cztery z nich, oznaczone czerwoną czcionką, pozwalają na analizę wydajności i szczegółów wektoryzacji programów. Ponadto, niektóre z nich wymagają dodatkowo zdefiniowania odpowiednich zmiennych środowiskowych. Pełna lista opcji wraz z opisem generowanych danych wyjściowych znajduje się w `PROGINF/FTRACE User's Guide` [10] oraz w dokumentacji poszczególnych kompilatorów [11, 12].

Programy mogą być uruchamiane bezpośrednio, poprzez podanie ścieżki dostępu do nich, lub pośrednio poprzez użycie programu ładującego VE (`ve_exec`) – przykłady z uwzględnieniem MPI zestawiono na Listingu 4. Listę opcji `mpirun` można uzyskać na stronie podręcznika systemowego `mpirun` [13] lub wydając polecenie `mpirun -h`.

```
$ ./program
$ ve_exec ./program
$ mpirun ./program
$ mpirun -v -np 2 -ve 0-1 ./program # używa kart VE numer 0 i 1
```

Listing 4: Uruchamianie sekwencyjnych i równoległych programów na karcie VE

Pełna dokumentacja dotycząca wszystkich komponentów SX-Aurora Tsubasa (sprzętowych i programowych) znajduje się na stronie internetowej NEC [15]. Przystępne dla użytkownika wprowadzenie znajduje się także na dedykowany blogu [16].

Wsadowy tryb pracy VE wymaga stworzenia skryptu i zlecenia go do kolejki Slurm. Przykładowa treść takiego pliku znajduje się na Listingu 5. Zawarte w nim opcje specyfikują nazwę zadania (`-J`), żadaną

liczbę węzłów obliczeniowych (`-N`), rdzeni CPU (`--ntasks-per-node`), pamięć (`-mem`; tutaj w meba-bajtach), limit czasowy (`--time`), identyfikator grantu (`-A`), partition (`-p`), domyślne zasoby (`--gres`), plik wyjściowy (`--output`), a także właściwe instrukcje, które zostaną wykonane w ramach zadania po przyznaniu zasobów. Wyczerpująca lista dostępnych znajduje się w dokumentacji [14].

```
#!/bin/bash -l
#SBATCH -J name
#SBATCH -N 1
#SBATCH --ntasks-per-node 1
#SBATCH --mem 1000
#SBATCH --time=1:00:00
#SBATCH -A <Grant ID>
#SBATCH -p ve
#SBATCH --gres=ve:1
#SBATCH --output=out

./program
```

Listing 5: Przykładowy skrypt Slurm

Listing 6 przedstawia podstawowe polecenia służące pracy z zadaniami Slurm: zlecenie zadania do kolejki (`sbatch`), które zwraca nadany identyfikator, wyświetla wszystkie zadania użytkownika i ich status (`squeue`), zwraca szczegóły dotyczące wybranego zadania (`scontrol`), anuluje wykonanie zadania (`scancel`). Pełna lista dostępnych opcji znajduje się w dokumentacji Slurm [14].

```
$ sbatch job.sl # zleca zadanie do kolejki
$ squeue -u $USER # listuje wszystkie zadania uzytkownika
$ scontrol show job <ID> # zwraca szczegoly zadania o danym identyfikatorze
$ scancel <ID> # anuluje zadanie o danym identyfikatorze
```

Listing 6: Przykładowe polecenia Slurm

Klaster Rysy nie posiada dedykowanego systemu plików na potrzeby obliczeń. Oznacza to, że wszystkie obliczenia należy prowadzić w swoim katalogu domowym (`$HOME`). Partycja `ve` (węzeł PBaran) jest ponadto przeznaczony do zadań wykorzystujących karty VE i nie powinien być używany do zadań, które znacznie obciążają CPU.

3 SOL: Transparent Neural Network Acceleration

Projekt SOL ma na celu optymalizację pracy z sieciami neuronowymi [17]. Jest to oprogramowanie pośredniczące (ang. *middleware*), które integruje się z takimi narzędziami, jak PyTorch, TensorFlow i MxNet. SOL posiada wsparcie dla architektury NEC SX-Aurora Tsubasa, ale może też pracować na standardowych platformach CPU i GPU (x86, ARM64, NVIDIA) [18].

Oprogramowanie wraz z kopią dokumentacji znajduje się w katalogu `/apps/nec/sol`. SOL dostępny jest w formie pakietu Python Wheel (nie jest dostępny w systemie modułów) i tym samym każdy użytkownik musi wykonać własną, lokalną instalację za pomocą narzędzia `pip` – patrz Listing 7. Przykłady użycia, publikacje i prezentacje dostępne są na stronie internetowej projektu [17].

```
$ pip3 install --user /apps/nec/sol/sol-0.1.8-py3-none-any.whl # instaluje SOL
$ pip3 install --user torch torchvision numpy # instaluje inne pakiety
```

Listing 7: Instalacja SOL

Literatura

- [1] NEC SX Vector Supercomputer
<https://www.nec.com/en/global/solutions/hpc/sx/index.html>

- [2] VEOS: Vector Engine Operating System
<https://github.com/veos-sxarr-NEC/veos>
- [3] SSH: Secure Shell
https://en.wikipedia.org/wiki/Secure_Shell
- [4] Slurm Workload Manager
<https://slurm.schedmd.com/overview.html>
- [5] Environment Modules
<https://modules.readthedocs.io/en/latest>
- [6] echo (command)
[https://en.wikipedia.org/wiki/Echo_\(command\)](https://en.wikipedia.org/wiki/Echo_(command))
- [7] export command
<https://ss64.com/bash/export.html>
- [8] Environment variable
https://en.wikipedia.org/wiki/Environment_variable
- [9] source command
<https://ss64.com/bash/source.html>
- [10] PROGINF/FTRACE User's Guide
https://www.hpc.nec/documents/sdk/pdfs/g2at03e-PROGINF_FTRACE_User_Guide_en.pdf
- [11] NEC C/C++ Compiler User's Guide
<https://www.hpc.nec/documents/sdk/pdfs/g2af01e-C++UsersGuide-016.pdf>
- [12] NEC Fortran Compiler User's Guide
<https://www.hpc.nec/documents/sdk/pdfs/g2af02e-FortranUsersGuide-016.pdf>
- [13] mpirun command
<https://www.open-mpi.org/doc/v4.0/man1/mpirun.1.php>
- [14] Slurm Workload Manager: Documentation
<https://slurm.schedmd.com/documentation.html>
- [15] NEC SX-Aurora Tsubasa Documentation
<https://www.hpc.nec/documents/>
- [16] NEC Blog: First Steps with the SX-Aurora Tsubasa vector engine
<https://sx-aurora.github.io/posts/VE-first-steps>
- [17] SOL: Transparent Neural Network Acceleration
<http://sysml.neclab.eu/projects/sol>
- [18] SOL: Talks/Publications
<http://sysml.neclab.eu/projects/sol/talks>