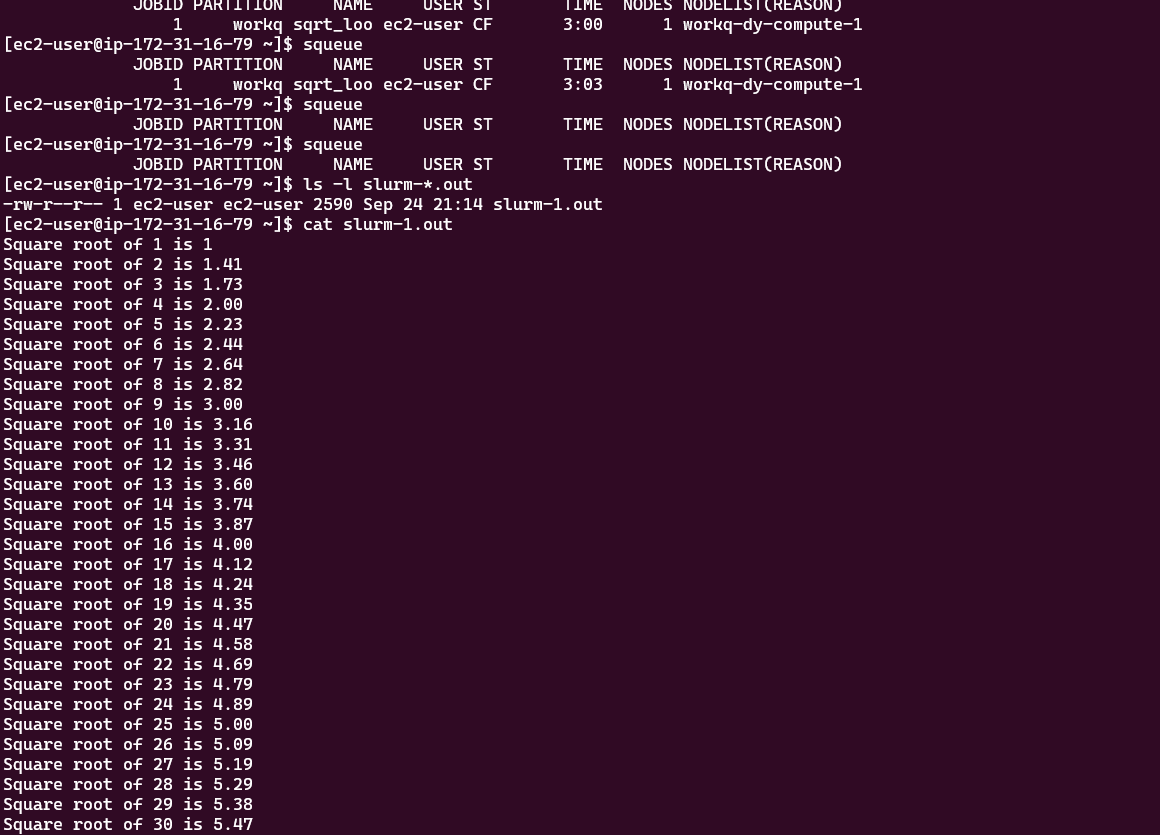
No dobrze, konfiguracja była męką, ale w końcu udało mi się. Okazało się, że problemem były permission policies na AWS.

Tworzę prosty skrypt obliczeniowy, dam kod tutaj, żeby już nie komplikować i ułatwić sprawdzanie:

Zrobię skrypt który liczy pierwiastki od 1 do 10000 z dokładnością do 2 miejsc, max 3 minuty

Sama konfiguracja zajęła mi bardzo długo, a output już bardzo prosty:



Kod bardzo prosty:

#!/bin/bash

#SBATCH --nodes=1

#SBATCH --ntasks-per-node=1

#SBATCH --time=00:05:00

#SBATCH --job-name=sqrt\_loop

for i in {1..100}

do

echo "Square root of $i is $(echo "scale=2; sqrt($i)" | bc)"

done

potem porównałem 1 node z 2 nodami, dostałem taki output

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez AI może być niepoprawna.

Użycie 2 nodes dało znaczące przyspieszenie

To kod do 2 węzłów:

cat <<'EOF' > sqrt\_parallel.sh

#!/bin/bash

#SBATCH --nodes=2

#SBATCH --ntasks-per-node=2

#SBATCH --time=00:10:00

#SBATCH --job-name=sqrt\_parallel

START=$(( (SLURM\_PROCID \* 5000) + 1 ))

END=$(( START + 4999 ))

for i in $(seq $START $END)

do

echo "Task $SLURM\_PROCID: sqrt($i) = $(echo "scale=2; sqrt($i)" | bc)"

done

EOF

Dobra, teraz użyłem c5.large

Porównamy czasy dla node 1 vs node 2 dla c5.large

Obraz zawierający tekst, zrzut ekranu

Zawartość wygenerowana przez AI może być niepoprawna.

c5.large (1 node), start: 22:18:48, finish: 22:19:37, 49 s

c5.large (2 nodes) , start: 22:18:48, finish: 22:19:00, 12 s

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez AI może być niepoprawna.

Wniosek jest prosty: jest szybciej dla c5.large

Porównuję koszty:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez AI może być niepoprawna.

Czyli najtaniej to t3 small 1 node

Najszybciej c5 large 2 nodes

Balans to t3 small 2 nodes – drożej niż 1 node tylko troche, a 2 x szybciej

Wizualizacja

Obraz zawierający zrzut ekranu, tekst, linia, Równolegle

Zawartość wygenerowana przez AI może być niepoprawna.

Obliczenia efektywności

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez AI może być niepoprawna.

Kluczowe koncepty:

Klastry składają się w AWS z HeadNode i Compute Nodes, zarządzanie zadaniami odbywa się w kolejkach, mamy łatwą skalowalność – można zmienić instancję i liczbę węzłów, dopasowując środowisko do potrzeb, minusem może być czasem lawinowo rosnący koszt, warto o tym pamiętać w trakcie konfiguracji

HPC można stosować

W symulacjach matematyczno-fizycznych

W bioinformatyce– machine learning I szukanie ukrytych wzorców w wielowątkowych danych omicznych, ale też analiza dużych kohort z RNAseq, jak i mapowanie oraz analiza single-cell RNAseq

Szkoelenie dużych modeli

T3 small jest tani, ale wolny, c5 large znacznie szybszy, ale bardziej kosztowny