Porting IFS Dwarfs to SX-AURORA TSUBASA

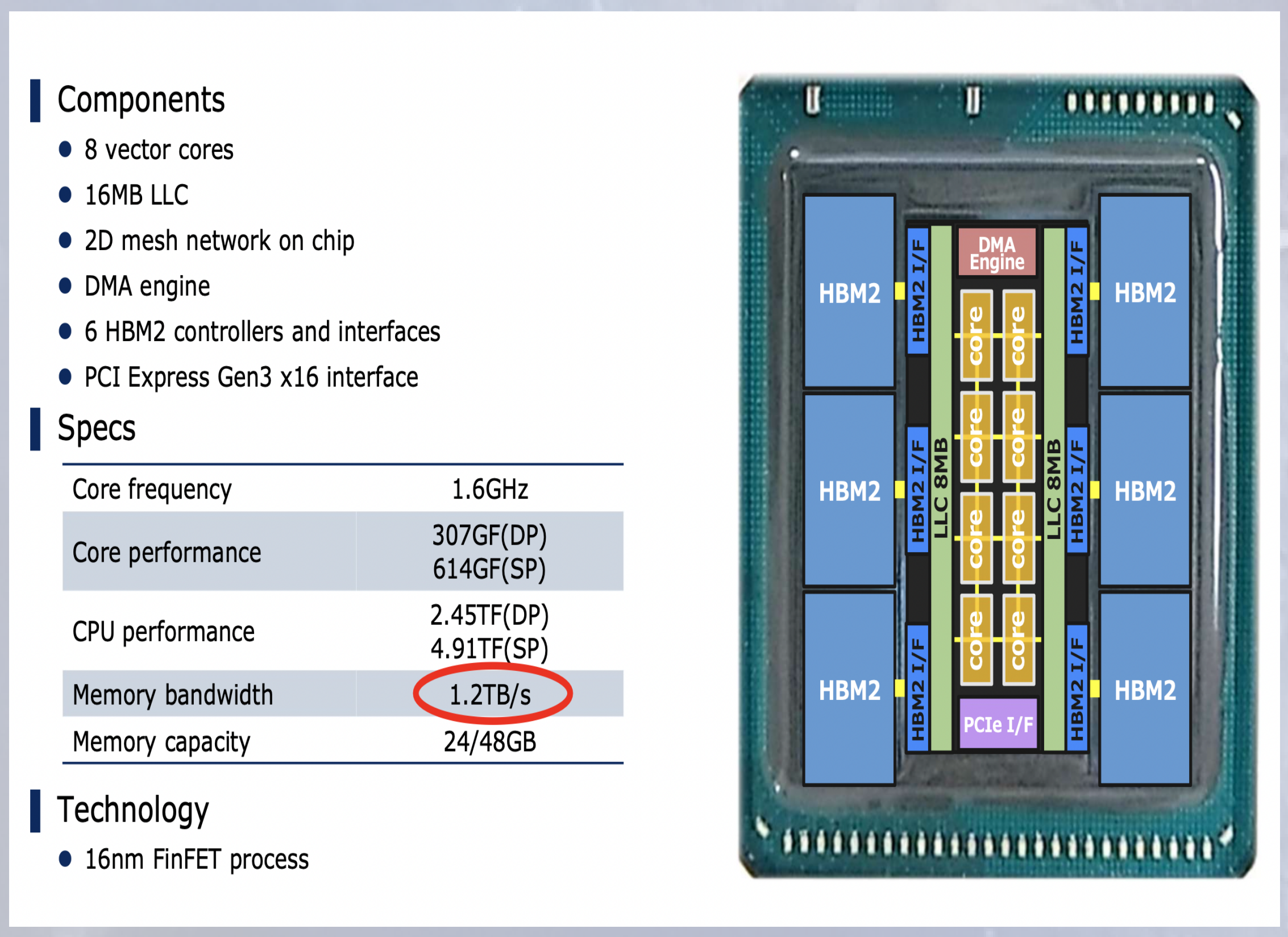
Fatemeh Pouyan, Ioan Hadade, Olivier Marsden
European Centre for Medium-Range Weather Forecasts (ECMWF)

Motivation

- Investigating further vectorisation and optimisation opportunities for the IFS application
- Identifying the root causes of the potential percolumnwince bottlenecks
- Discovering potential application optimization options
- Exploring various architectural decisions that are well-suited for computational demands of the IFS application

IFS dwarfs are standalone mini applications that consist of set of algorithms to present the key functional blocks of various parts of IFS application such as cloud microphysics scheme (CLOUDSC) or radiation scheme (ECRAD).

NEC Architecture Overview


Components

- 8 vector cores
- 16MB LLC
- 2D mesh network on chip
- DMA engine
- 6 HBM2 controllers and interfaces
- PCI Express Gen3 x16 interface

Specs

Core frequency	1.6GHz
Core performance	307GF(DP) 614GF(SP)
CPU performance	2.45TF(DP) 4.91TF(SP)
Memory bandwidth	1.2TB/s
Memory capacity	24/48GB

Technology

- 16nm FinFET process

Fig. 1: Vector Engine Processor Overview [1]

Method Overview

The NEC compiler can vectorize many loops automatically, however, in some cases, benchmarking and profiling are essential to identify bottlenecks and tune the performance.

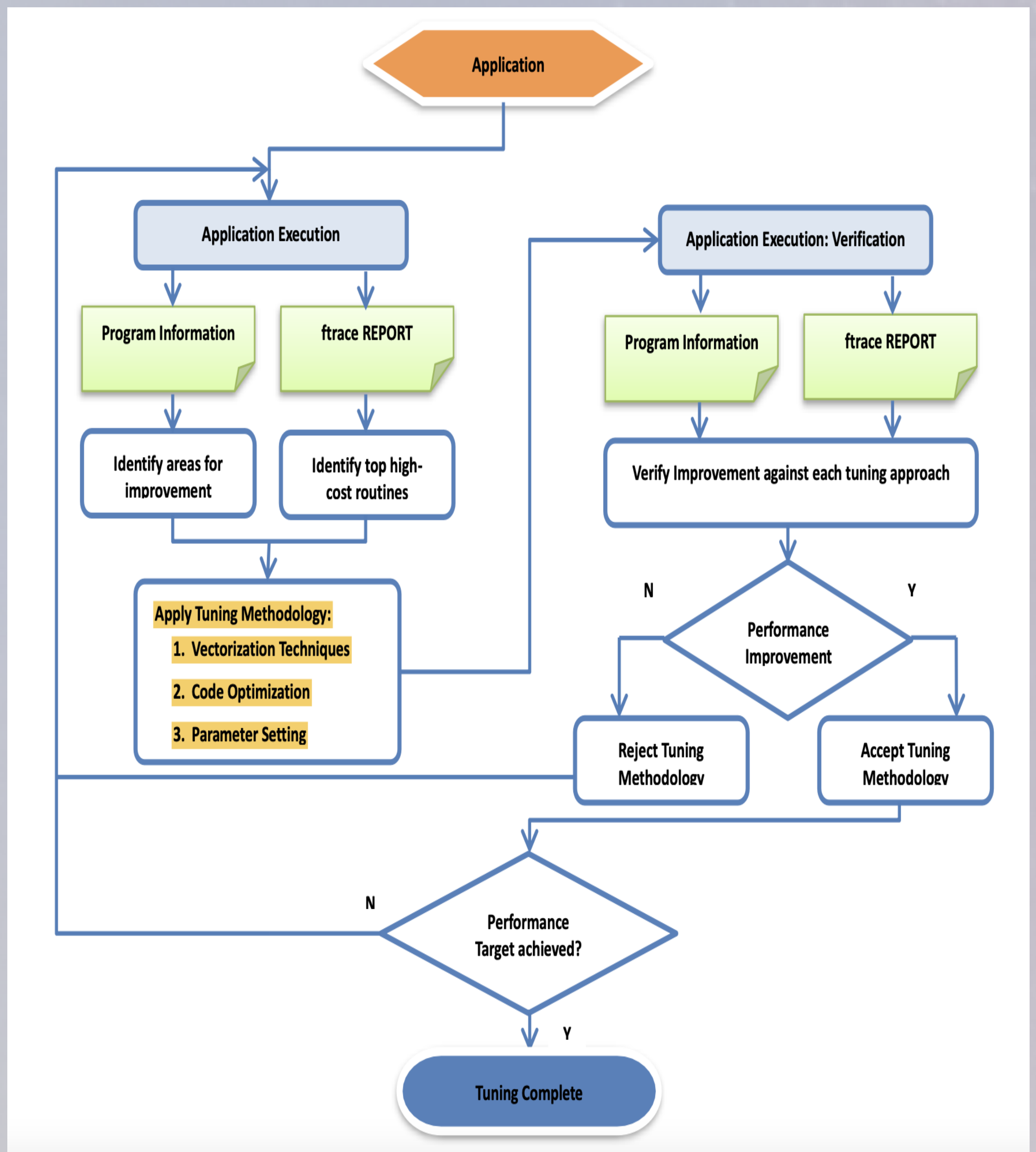
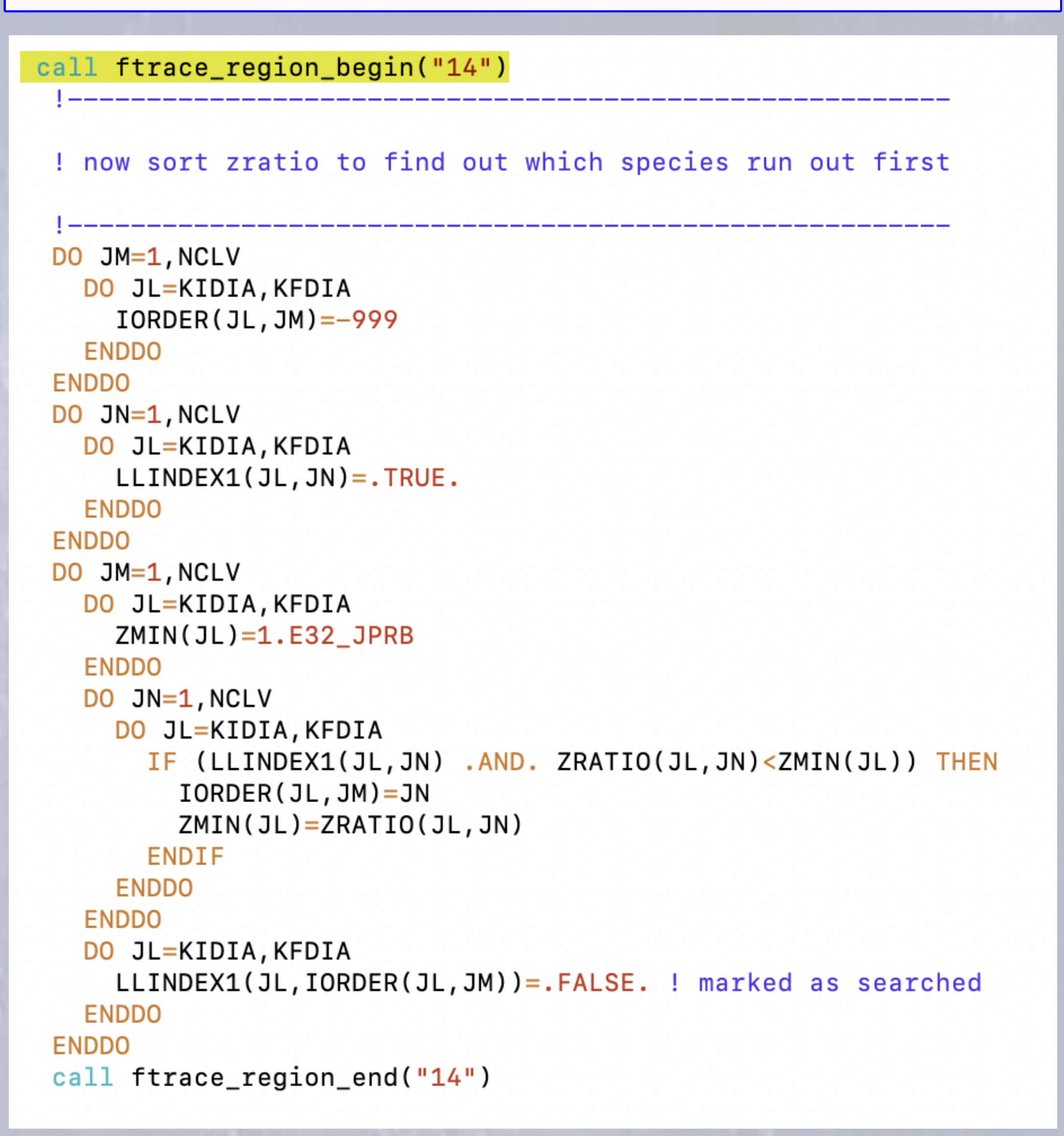


Fig. 2: Benchmarking, Profiling and Performance Tuning [2]

CLOUDSC on NEC

1 Identifying the hot-spot regions :

Instrumenting Ftrace regions inside various code sections and taking time of those sections.


```
call ftrace_region_begin("14")
  !----------------------------------------------------------

  ! now sort zratio to find out which species run out first

  !----------------------------------------------------------
  DO JM=1,NCLV
    DO JL=KIDIA,KFDIA
      IORDER(JL,JM)=-999
    ENDDO
  ENDDO
  DO JN=1,NCLV
    DO JL=KIDIA,KFDIA
      LLINDEX1(JL,JN)=.TRUE.
    ENDDO
  ENDDO
  DO JM=1,NCLV
    DO JL=KIDIA,KFDIA
      ZMIN(JL)=1.E32_JPRB
    ENDDO
    DO JN=1,NCLV
      DO JL=KIDIA,KFDIA
        IF (LLINDEX1(JL,JN) .AND. ZRATIO(JL,JN)<ZMIN(JL)) THEN
          IORDER(JL,JM)=JN
          ZMIN(JL)=ZRATIO(JL,JN)
        ENDIF
      ENDDO
    ENDDO
    DO JL=KIDIA,KFDIA
      LLINDEX1(JL,IORDER(JL,JM))=.FALSE. ! marked as searched
    ENDDO
  ENDDO
  call ftrace_region_end("14")
```

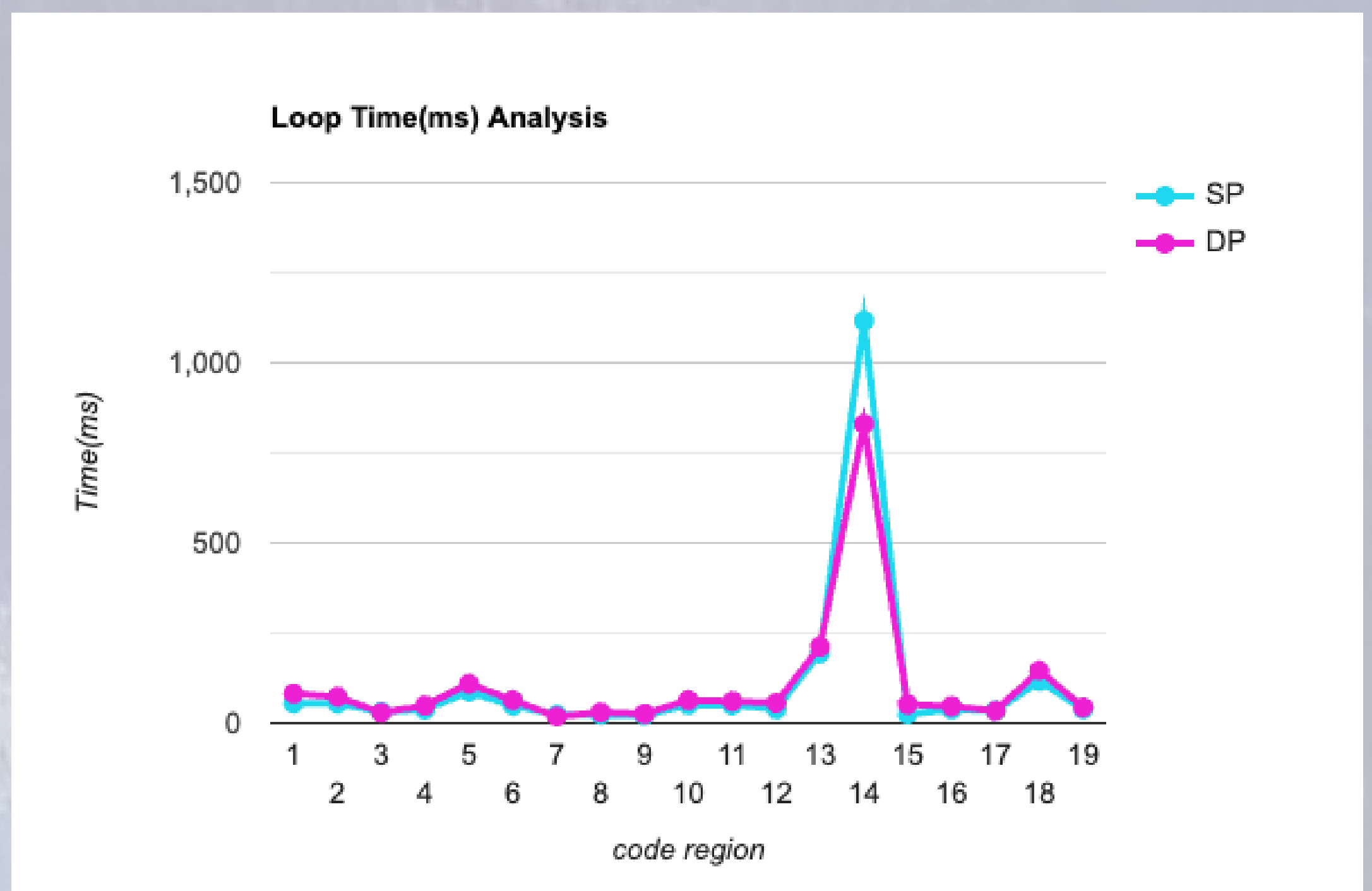


Fig. 4: Time analysis of different regions

2 Improving performance by modifying the hot-spot region :

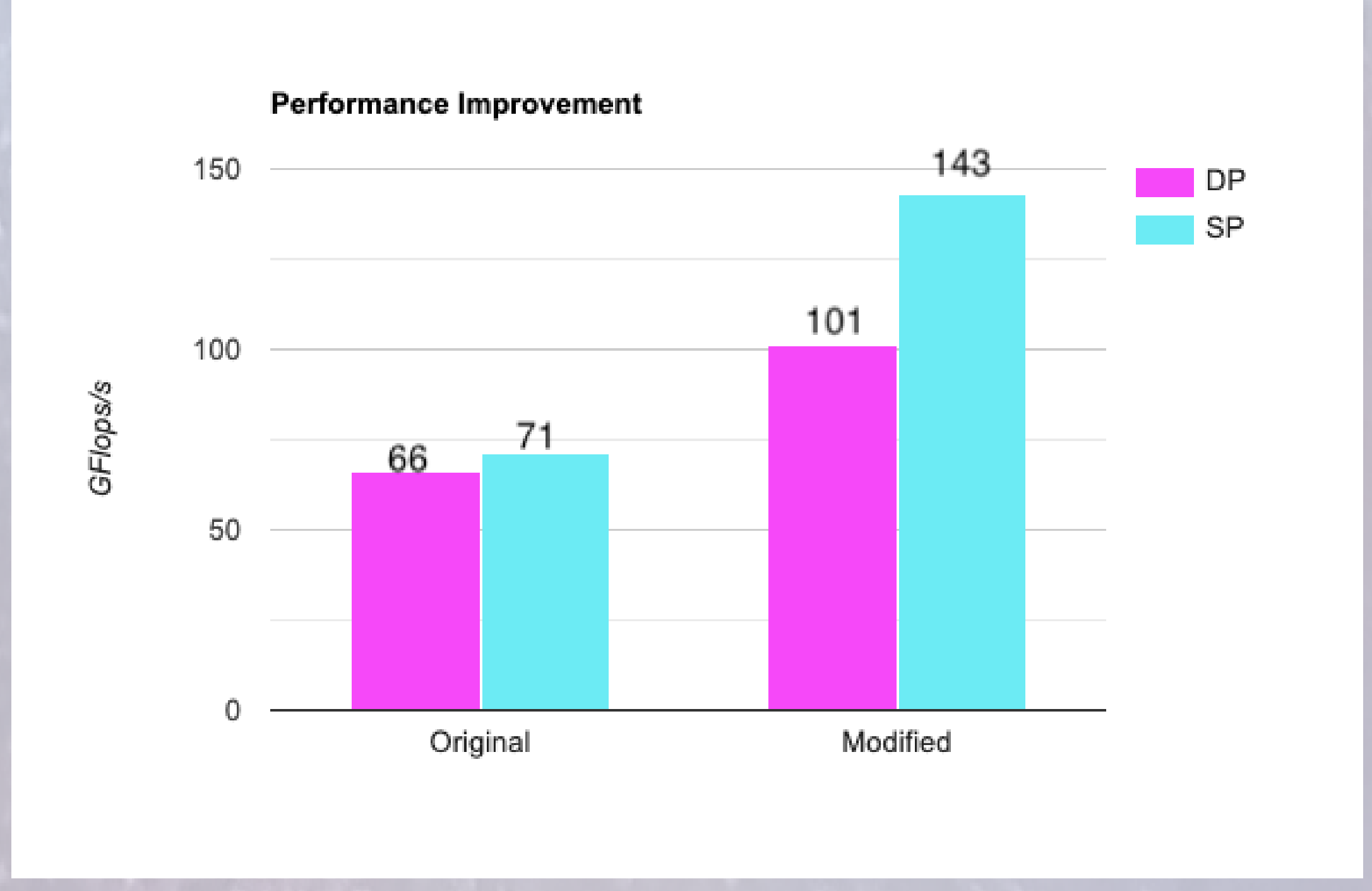


Fig. 5: Original Performance vs Optimised Performance

CLOUDSC on NEC vs AMD-EPYC

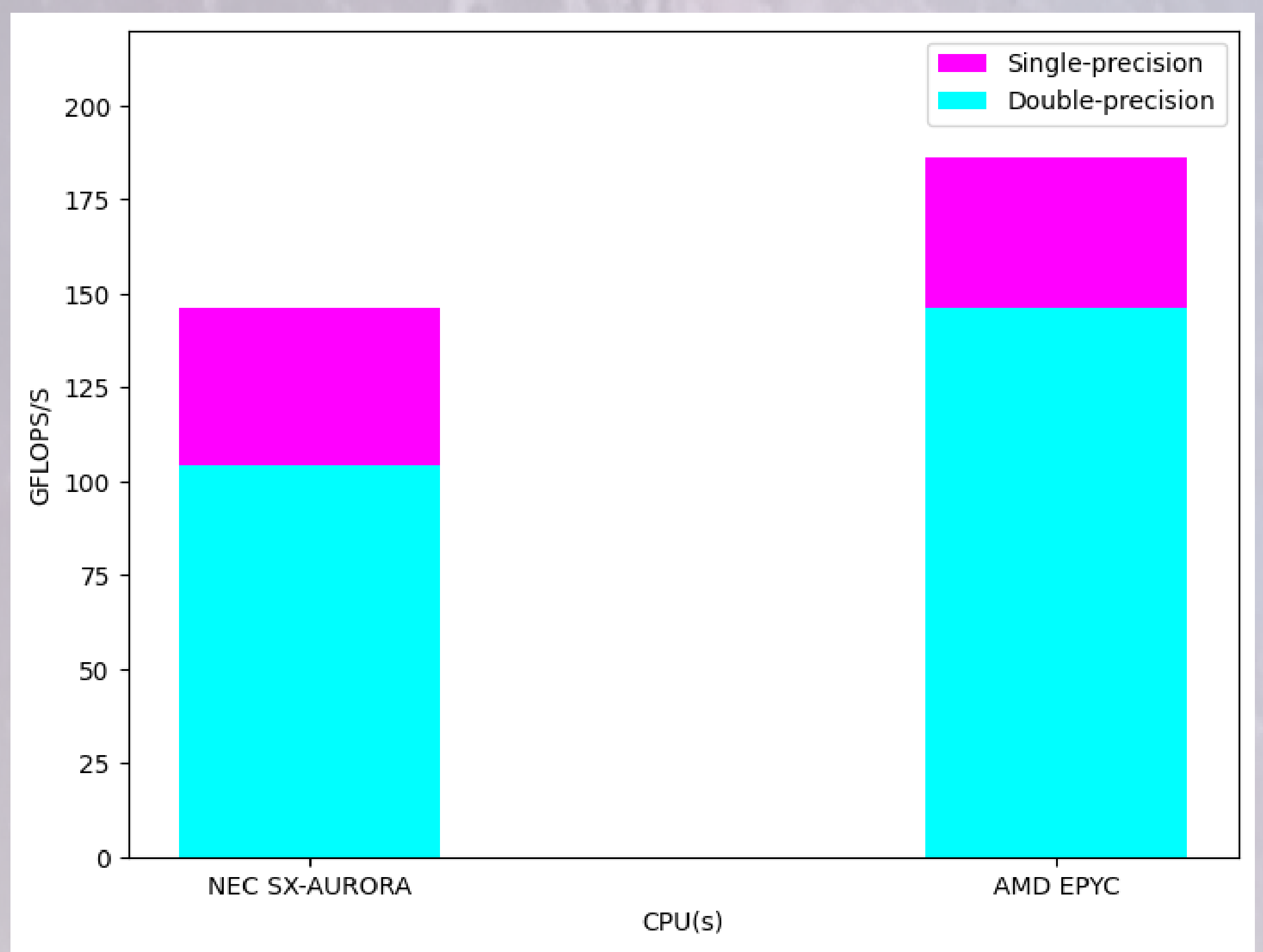


Fig. 6: Identifying hot-spot regions

Is CLOUDSC a CPU-Dominant Mini-Application?

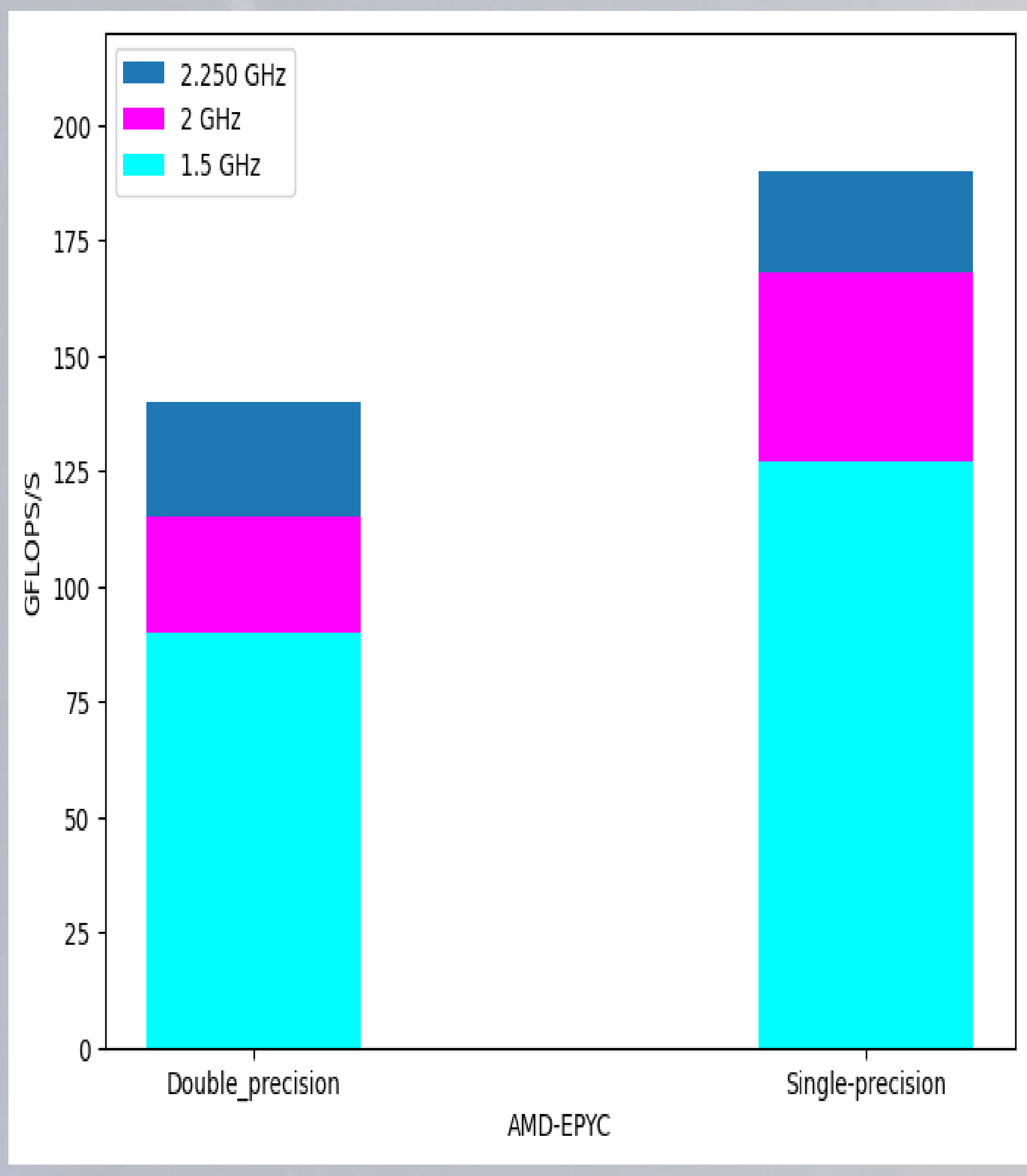


Fig. 7: Impact of CPU Frequency on Cloudsc Performance

ECRAD on NEC

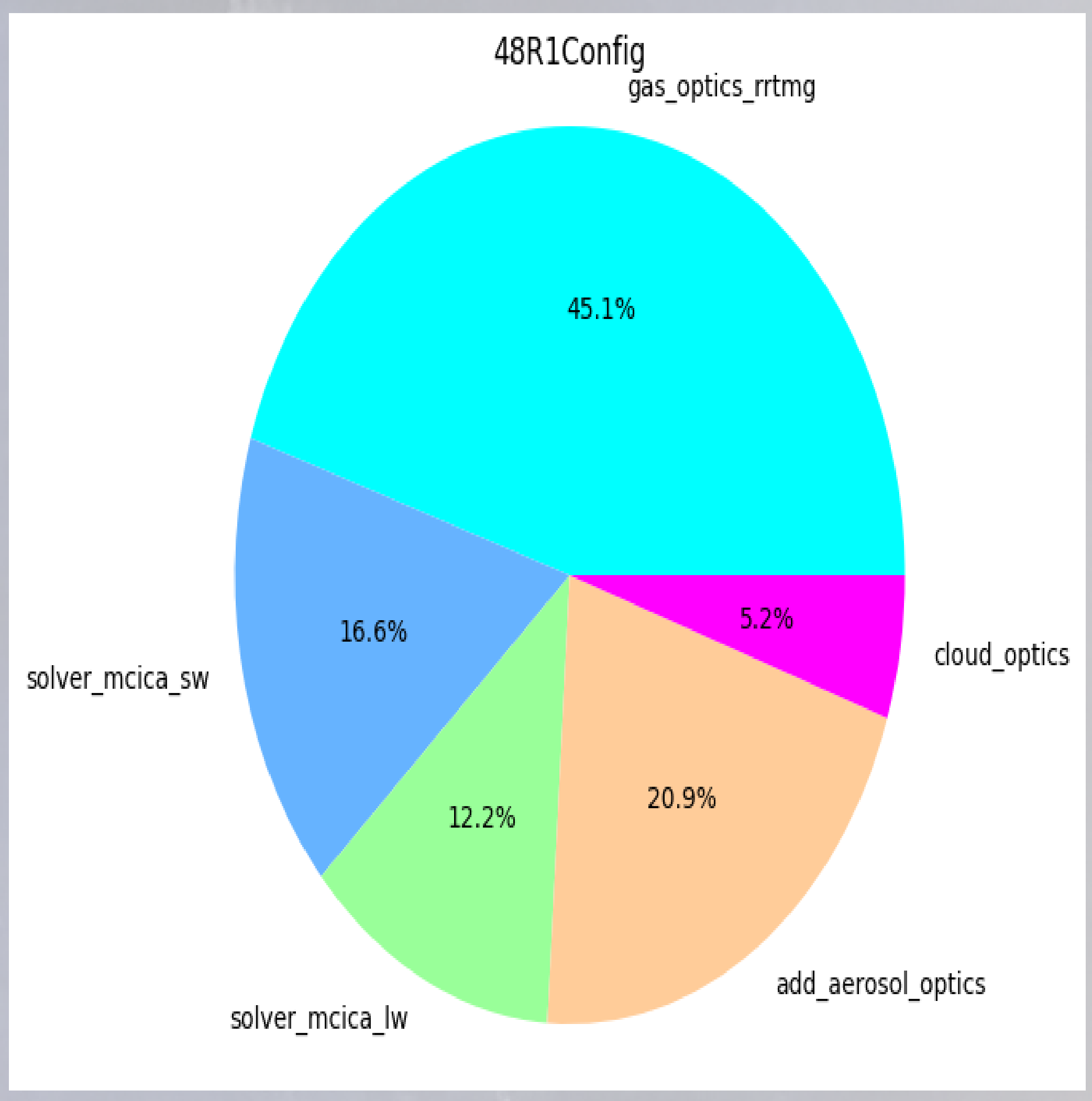


Fig. 8: Hot-spot subroutines for 48R1 Config - NPRONA = 80)

ECRAD on NEC vs AMD-EPYC

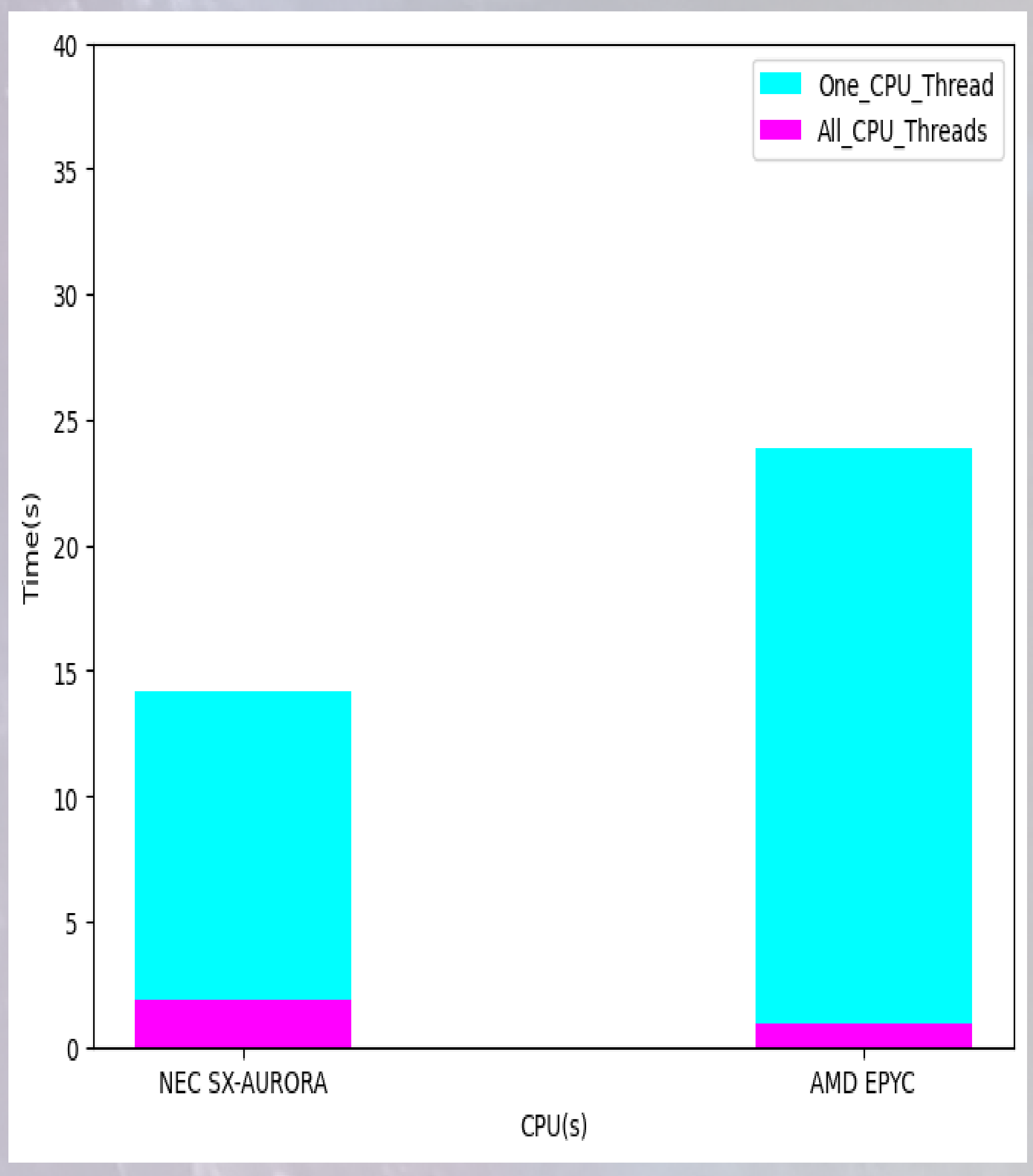


Fig. 9: Execution Time(s) of ECRAD on NEC vs AMD-EPYC

References

[1] Erich Focht. “Introduction to the NEC SX-Aurora Vector Engine”. In: ().
[2] “SX-Aurora TSUBASA Performance Tuning Guide”. In: ().