Accelerating Sparse Linear Solvers for O&G Reservoir Simulations Using NVIDIA Grace

Felipe Portella
IT Advisor, Petrobras

João Paulo Navarro
Senior Solutions Architect, NVIDIA

March/2024

Session: S62529

Speakers Biography

Felipe Portella

PETROBRAS

HPC Advisor

M.Sc. PUC-Rio, Brazil

Ph.D. Candidate
BSC/UPC, Spain

Expertise:
- Petroleum reservoir simulation
- HPC for Research & Development

João Paulo Navarro

NVIDIA

Senior Solutions Architect

M.Sc., UFJF, Brazil

Expertise:
- Energy segment
- HPC for seismic processing, machine learning, and data analytics in GPU architectures

Team

Fabiane Vieira Castro
CESAR

Felipe Albuquerque Portella
Petrobras / BSC / UPC

João Paulo Navarro
NVIDIA

José Roberto Pereira Rodrigues
Petrobras

Leonardo da Silva Gasparini
Petrobras

Luigi Fernando Marques da Luz
CESAR

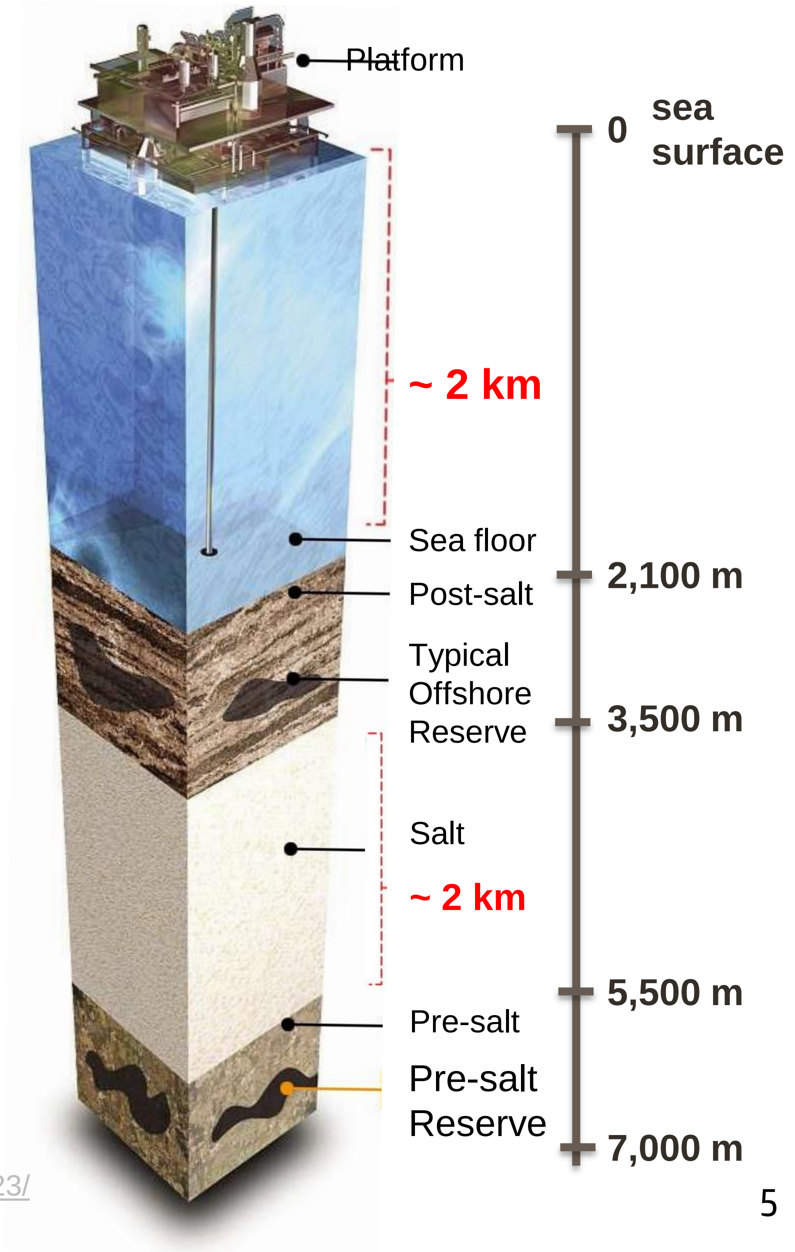
Vitor Barros Aquino
CESAR

Agenda

- Background
  - Petrobras & O&G Workloads
  - SolverBR
- The Journey
- Results and Discussion
- Closing Remarks

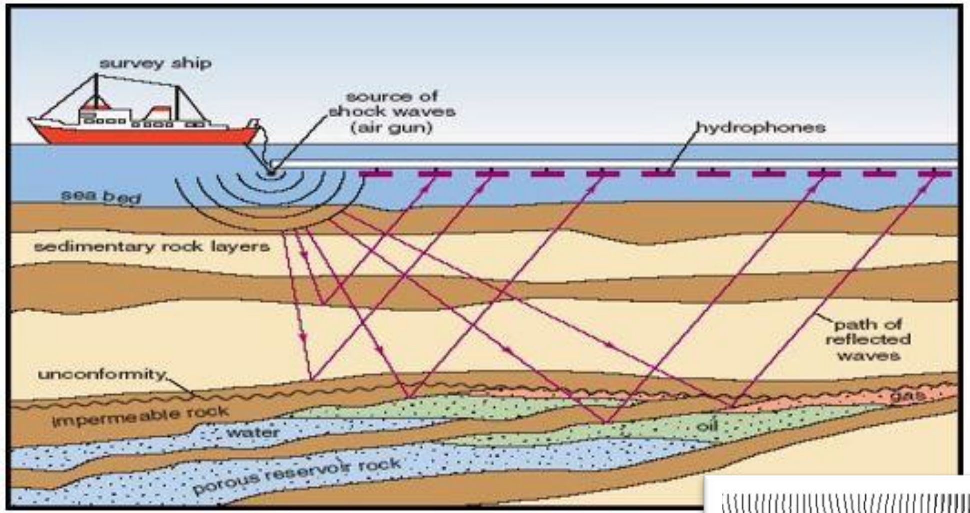
Petrobras Quick Facts

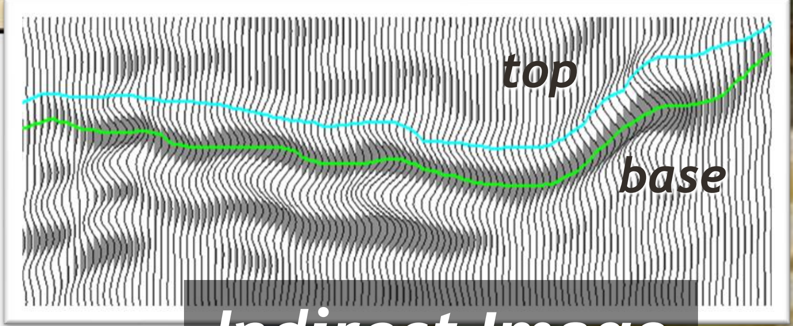
- Brazilian Energy Company
- Oil and Gas Exploration and Production as core business
- Reference in ultra-deepwater exploration
- Why HPC at Petrobras?
  - Each deep-water well typically costs between US$70 and US$100 million (1)
  - Petrobras will drill more than 300 offshore wells in the next 5 years (2)
  - Latin America #1 HPC: TOP500 and Green500 lists
  - Two main disciplines/workloads:
    - Geophysics – Seismic Processing
    - Reservoir Engineering – Reservoir Simulation



(1) https://www.agenciapetrobras.com.br/pt/inovacao/petrobras-monta-supercomputador-para-desenvolver-tecnologias-18-01-2023/
(2) Petrobras Strategic Plan 2023-2027

Seismic Processing





Indirect Image

Helps to answer the “WHERE”

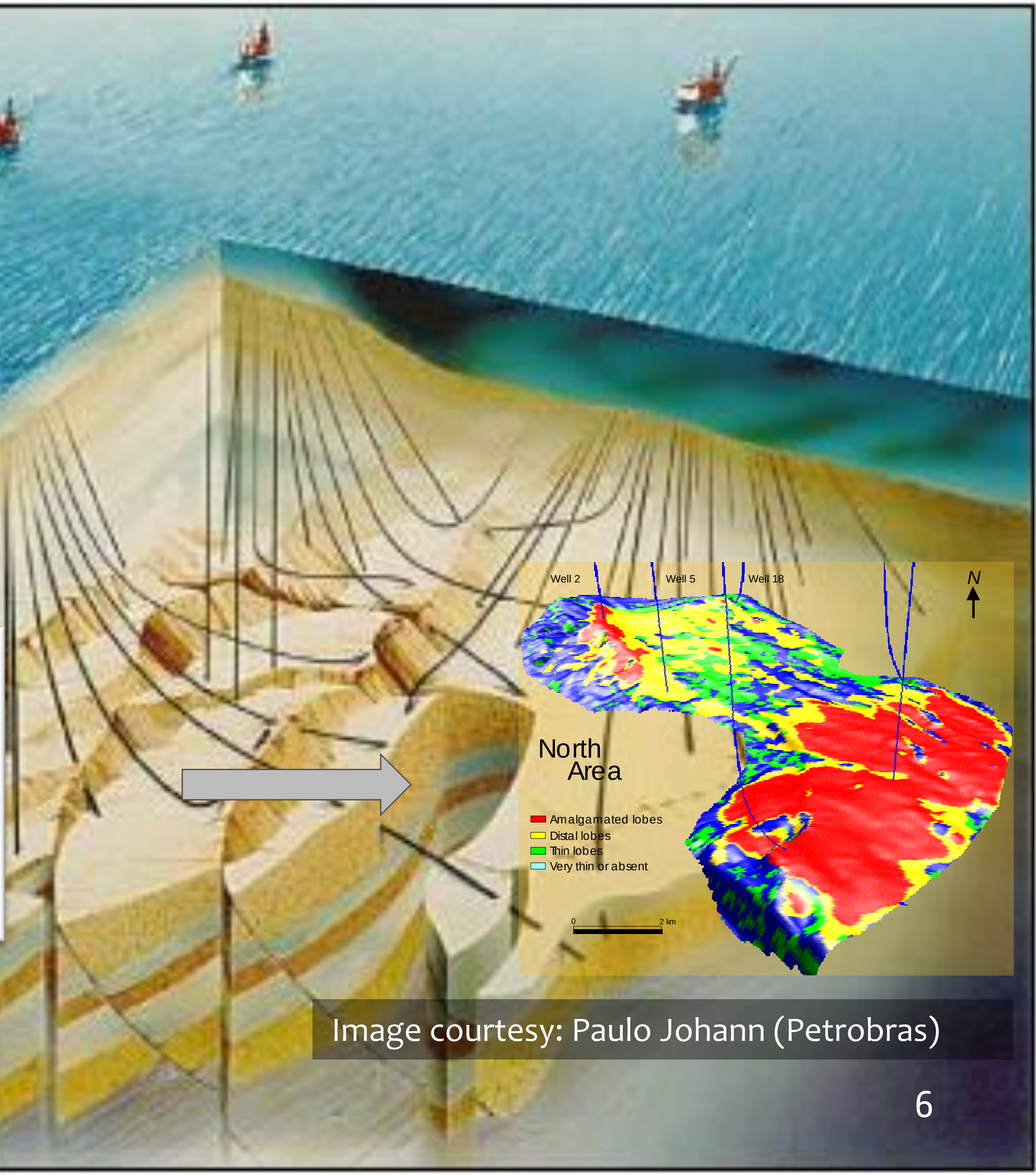


Image courtesy: Paulo Johann (Petrobras)

Reservoir Simulation

- **Wells**
  - How many?
  - Where should they be placed?
  - What type (vertical, horizontal, ...)?
  - Etc.
- **Recovery methods**
  - Primary depletion?
  - Injection of water or gas?
  - How to control the wells?
  - Etc.
- **Forecast behavior**
  - What are the final recovery volumes?
  - How about the cash flow?
  - Etc.
- **Etc.**

Helps to answer the "HOW"

NVIDIA C.E.S.A.R PETROBRAS

Extraction Strategy

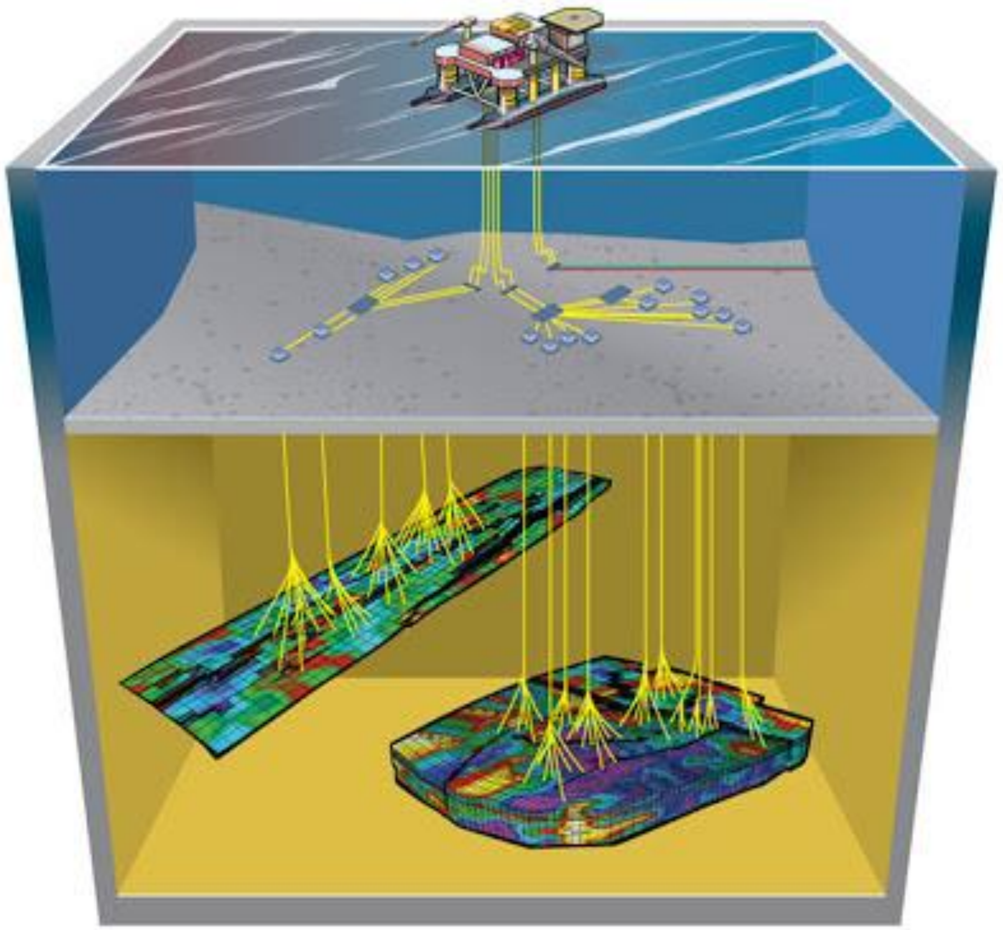


Image source: PetroBlogger.com

Main HPC Workloads

- Seismic Processing & Imaging
  - Single precision (FP32)
  - Expressively large datasets (TBs+)
  - Mix of CPU and **GPU** Jobs (GPU)
  - **GPU** jobs (typically) with InfiniBand
  - Key compute-intensive imaging algorithms are developed in-house and optimized for **GPU** utilization
  - Processing jobs can extend over **several weeks**
  - Some GPU jobs may use **hundreds of GPUs** (even thousands)

Main HPC Workloads

- Reservoir Simulation
  - Double precision (FP64)
  - Datasets of a smaller scale compared to seismic processing
  - Majority of **CPU** workloads (GPU acceleration still an exception)
  - Predominantly uses commercial applications
  - Relies on shared memory architectures
  - Jobs typically take hours
  - **Intensive ensemble workflows such as optimization or inverse problems**
    - batches of dozens or **hundreds of jobs** concurrently
- Other Relevant Workloads:
  - Machine Learning, Petrophysics, Geomechanics, Multiphysics, etc.

SolverBR Motivation

- Born from the need to accelerate reservoir studies
  - Faster simulations potentially enhances predictions and reduce uncertainties
  - Simulation time is an increasingly limiting factor
  - The **linear solver** is often the **most computationally expensive kernel**
  - Reservoir problems have specific characteristics that can be explored to achieve significantly higher performance than commercial (off-the-shelf) solvers

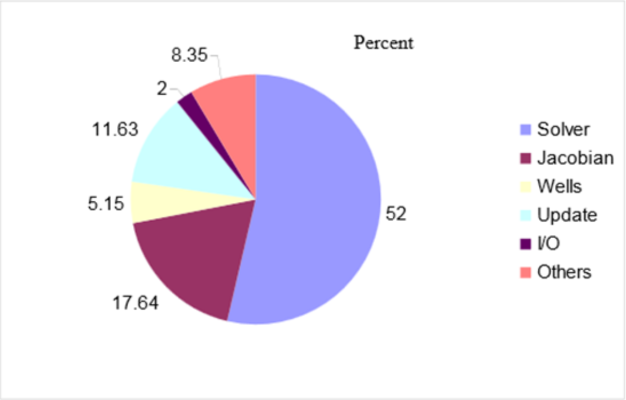


Figure 16- Distribution of Computational Work in a Next Generation Parallel Simulator

Source: Dogru et al., From Mega-Cell to Giga-Cell Reservoir Simulation, paper SPE 116675, 2008

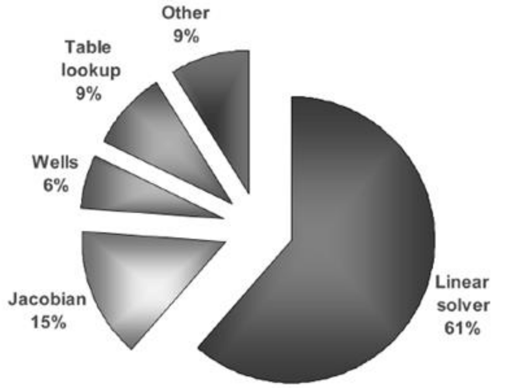


Fig. 3—Pie chart for computer time; 1.2 million-cell real reservoir, 26 years of history, 100 wells (Field Problem 1).

Source: Dogru et al., A Parallel Reservoir Simulator for Large-Scale Reservoir Simulation, SPE Reservoir Evaluation & Engineering, Feb 2002

SolverBR

- High-performance **sparse linear solver** specialized for reservoir simulation applications developed in collaboration with academic institutions

- It aims to combine excellence in parallel computing with the most advanced algorithms of sparse linear algebra for flow and geomechanics problems

Journal of Computational Science 51 (2021) 101330

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Hybrid parallel iterative sparse linear solver framework for reservoir geomechanical and flow simulation

Leonardo Gasparini [a], José R.P. Rodrigues [a], Douglas A. Augusto [b], Luiz M. Carvalho [c,*], Cesar Conopoima [d], Paulo Goldfeld [d], Jairo Panetta [e], João P. Ramirez [d], Michael Souza [f], Mateus O. Figueiredo [a], Victor M.D.M. Leite [g]

SolverBR

- Implemented in C++, on top of Linux, targeting dedicated reservoir simulation HPC clusters
- It efficiently supports various parallelism models: shared memory, distributed memory, and hybrid models
- Continuous build processes run weekly tests with different software stack and options (*e.g.*: different C++ compilers, with and without MPI, OpenMP, etc.)

Q2/2017
- Beginning of Development

Q2/2018
- Basic Solver (flow and geomechanics)

Q4/2018
- GPU Version (geomechanics)

Q2/2020
- Adoption in production for geomechanics

Q2/2021
- 2φ2k support

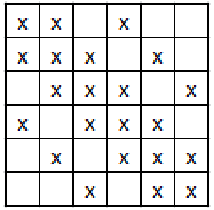
Q2/2022
- Distributed GEM+SBR prototype

Q1/2023
- Adoption in production for flow simulation

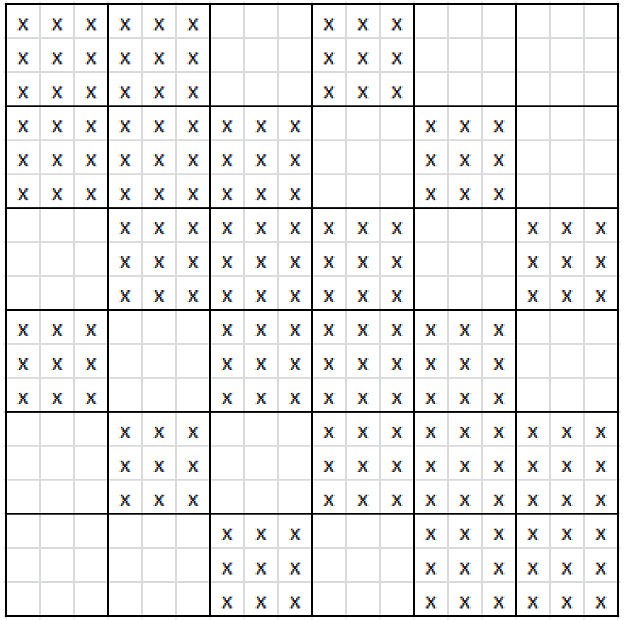
SolverBR Algorithms

- Several matrix storage formats (*e.g.*, scalar, blocked AIM)
- Several Krylov solvers (*e.g.*, CG, GMRES, BiCGStab)
- Several preconditioners (*e.g.*, ILU, AINV, CPR, Multiscale)
- Flexibility to combine and nest solvers and preconditioners

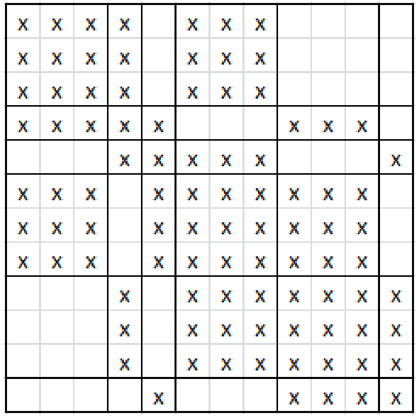
Scalar



Blocked (uniform block)



AIM (variable block)



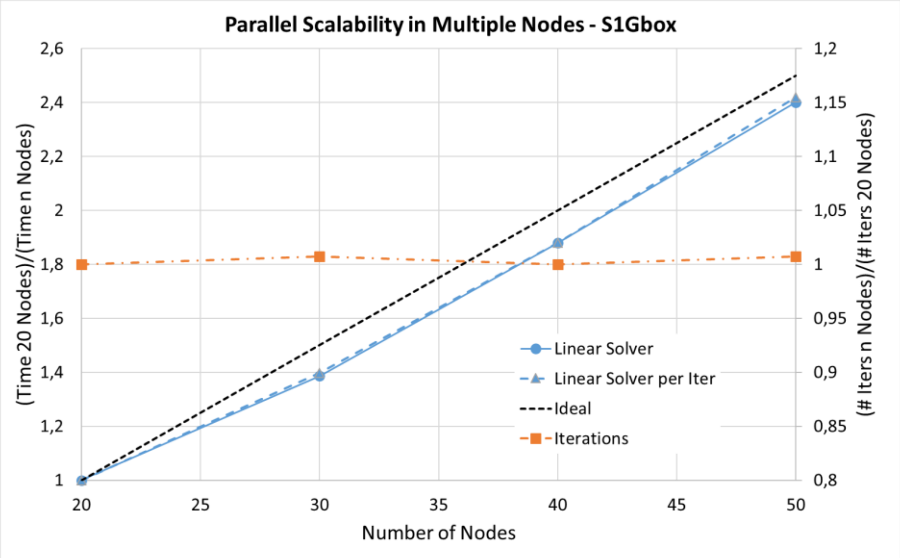
Integration of SolverBR with GeomecBR

- In 2020, we incorporated new features into SolverBR to enable the implementation of the multiscale method
- This method is used as a preconditioner, leveraging a coarse system to accelerate the convergence of the original system
- Reduction in the number of iterations can reach up to 80%


ELSEVIER
Upstream Oil and Gas Technology
Volume 11, September 2023, 100095

A parallel viscoplastic multiscale reservoir geomechanics simulator

Mateus O. de Figueiredo [a], Luis Carlos de Sousa Junior [a], Jose R.P. Rodrigues [a], Leandro B. dos Santos [a], Leonardo S. Gasparini [a], Ricardo F. do Amaral [b], Rafael J. de Moraes [b]



Integration of SolverBR with CMG

- Computer Modelling Group (CMG) is a software company specialized on reservoir simulation

- CMG's software suite includes industry-leading reservoir simulation tools such as IMEX, GEM and STARS

  - IMEX - Black Oil Reservoir Simulator
  - GEM – Compositional & Unconventional Simulator
  - STARS - Thermal & Advanced Processes Simulator

- Petrobras has a partnership with CMG for over 40 years!

Other Computer Architectures

- Petrobras initiated a project with UFF University in 2023 to reduce Cloud HPC costs, specifically targeting reservoir simulation
  - It is worth considering alternative instances, such as ARM, which has demonstrated better price-performance for various applications

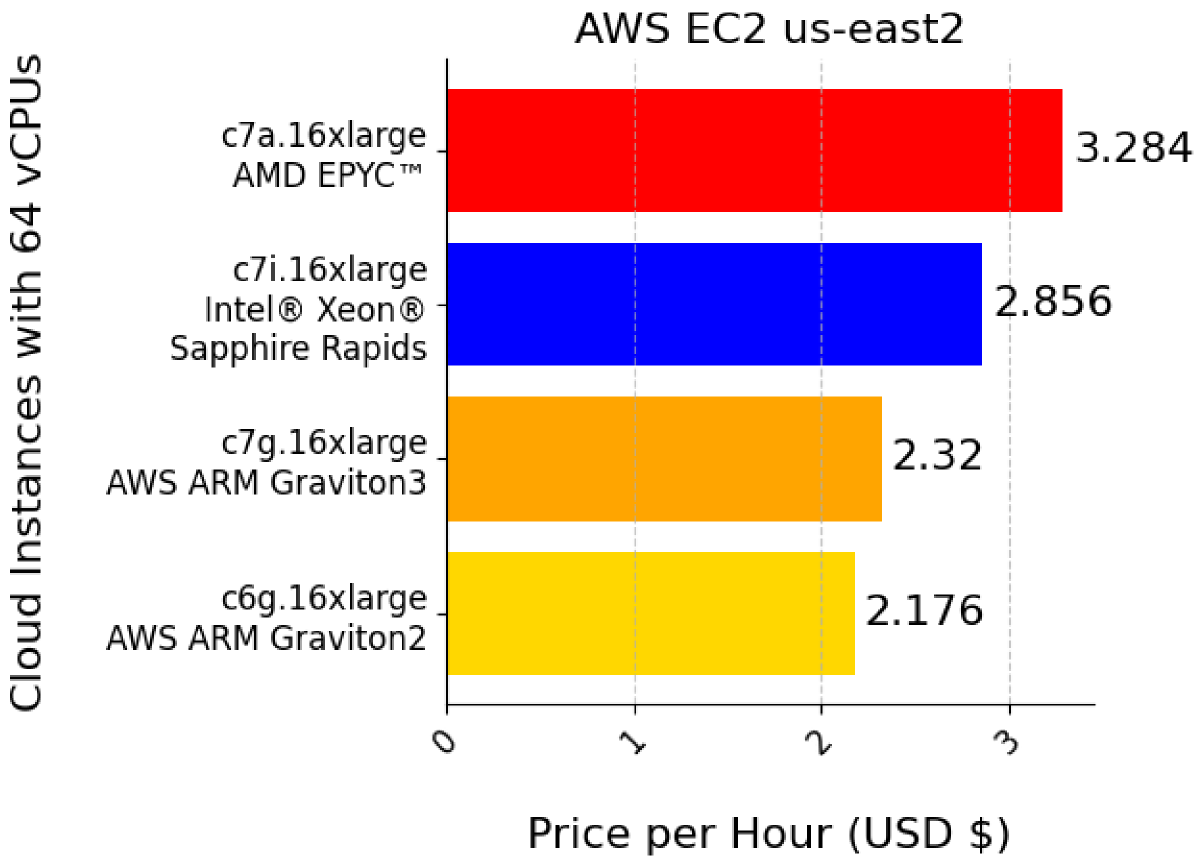
MScheduler: Leveraging Spot Instances for High-Performance Reservoir Simulation in the Cloud

Felipe A. Portella[*†‡], Paulo J. B. Estrela[*], Renzo Q. Malini[*], Luan Teylo[§¶], Josep L. Berral[‡†], Lúcia M. A. Drummond[¶]

- Petrobras started the “blue sky” project in collaboration with Instituto CESAR to port the SolverBR code to the ARM platform

Opportunity

- ARM architecture has gained attention in recent years
- To port SolverBR aiming to investigate numerical accuracy of ARM-based CPUs versus x86 for reservoir simulation
- Understanding the ARM price-performance in practice, both on the cloud and on-prem



The Journey

The NVIDIA Grace CPU

The building block of the superchip

High Performance Power Efficient Cores

72 flagship Arm Neoverse V2 Cores with
SVE2 4x128b SIMD per core

Fast On-Chip Fabric

3.2 TB/s of bisection bandwidth connects
CPU cores, NVLink-C2C, memory, and system IO

High-Bandwidth Low-Power Memory

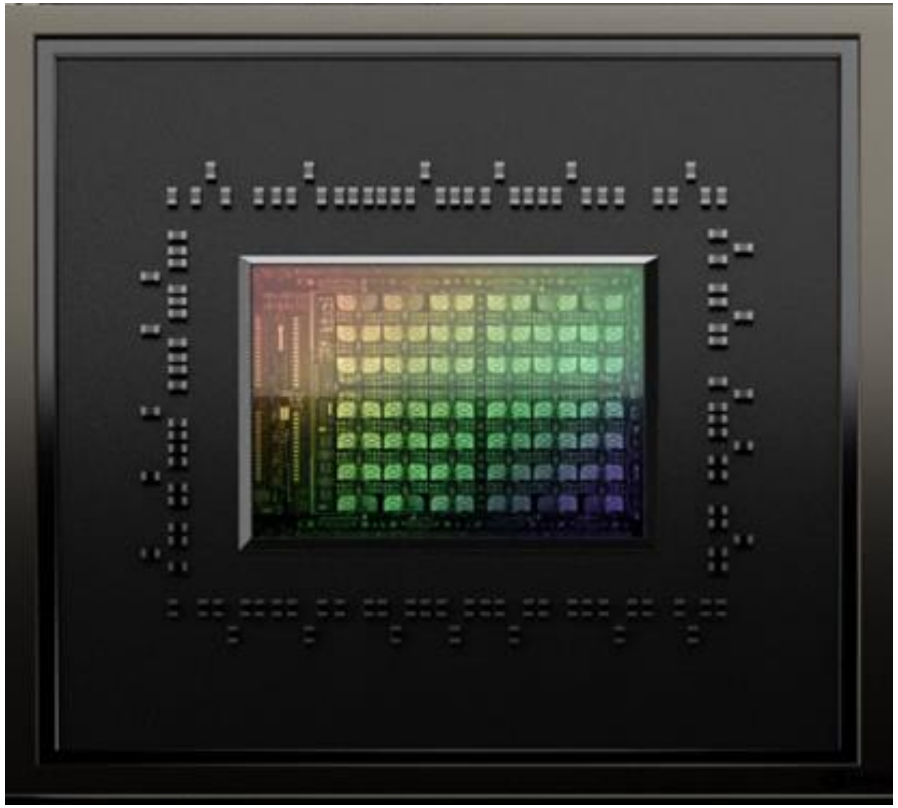
Up to 480 GB of data center enhanced LPDDR5X Memory that
delivers up to 500 GB/s of memory bandwidth

Coherent Chip-to-Chip Connections

NVLink-C2C with 900 GB/s bandwidth for coherent
connection to CPU or GPU

Industry Leading Performance Per Watt

Up to 2X perf / W over today's leading servers

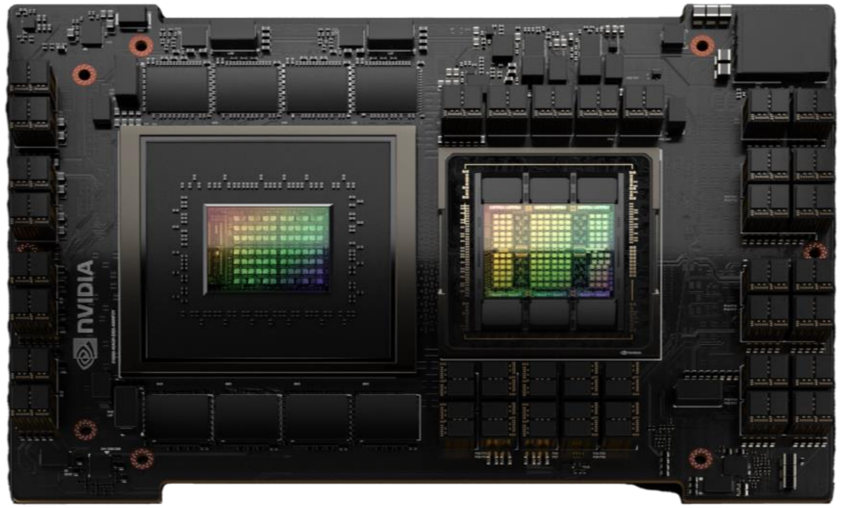


https://www.nvidia.com/en-us/on-demand/session/gtcfall22-a41129/

One Powerful CPU – Two Superchip Configurations

Grace Hopper Superchip (GH200)

More than "Grace + Hopper"



Grace CPU Superchip

More than "2 x Grace"

Use Standards-compliant Multi-platform Compilers

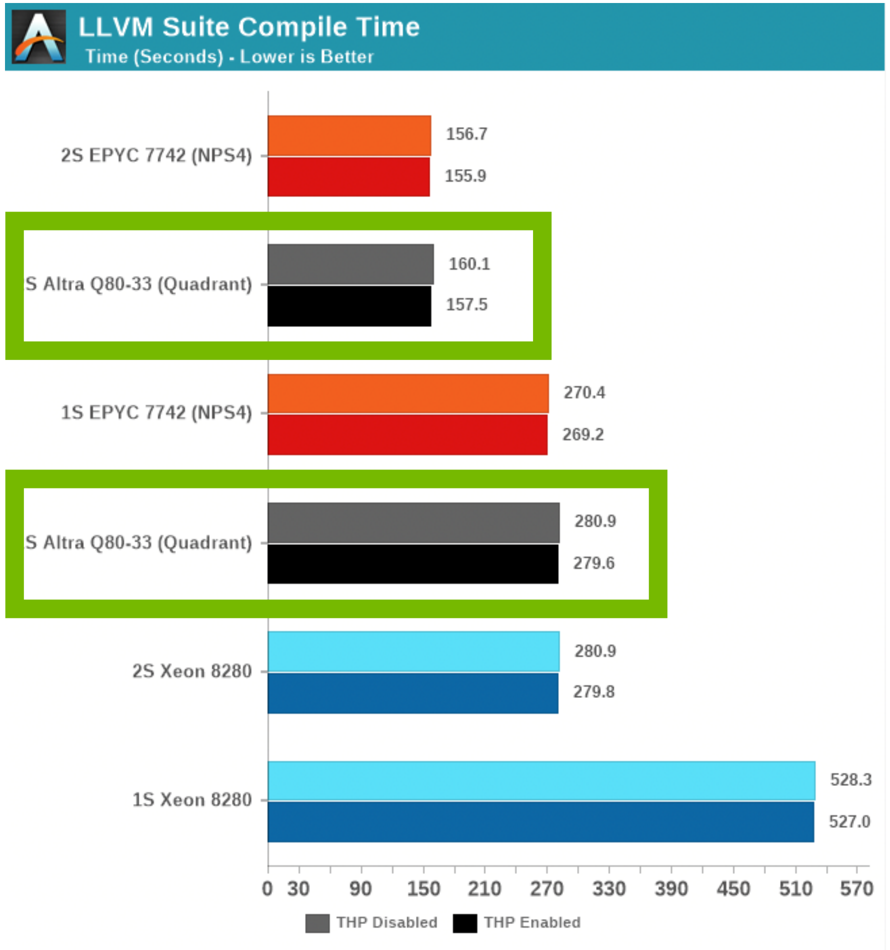
You're not porting to Arm. You're porting *away* from ifort, xlf, etc.

- Use any portable multi-platform compiler: NVIDIA, GCC, LLVM, etc.
- Use the most recent compiler possible. **GCC 12+** is strongly recommended.
- Beware of non-standard build systems
  - `icc`, `ifort`, `xlf`, etc. may be hard-coded into the build system
  - Be explicit about which compiler to use. Don't let the build system make assumptions
- Beware of non-standard default compilers
  - Check default compiler commands (`cc`, `fc`, `gcc`, etc.) invoke a recent compiler
  - Use `` `mpicc -show` `` to verify that MPI compiler wrappers invoke the right compiler
- Log the build, then check the log afterward

No Cross Compiling! Just Don't.

All popular build systems are supported – and *performant* – on Arm

- GCC and LLVM are excellent Arm compilers
  - Auto-vectorizing, auto-parallelizing, tested, in production
  - Arm & partners are the majority of GCC contributors
- All major build systems and tools work on Arm
  - CMake, Make, GNUMake, EasyBuild, Spack etc.
- Compiler & build system performance is excellent
  - Ampere Altra compilation performance is on is on-par with AMD EPYC 7742 – **you do not need to cross compile**



https://www.anandtech.com/show/16315/the-ampere-altra-review/8

Selecting GNU and LLVM Compiler Flags for Grace

Similar flags have different meanings across compilers and across platforms

- Remove all architecture-specific flags: -mavx, -mavx2, etc.
- Remove -march and -mtune flags
  - These flags have a different meaning on aarch64 [details]
- Use **-Ofast -mcpu=native**
  - If fast math optimizations are not acceptable, use -O3 –ffp-contract=fast
  - For even more accuracy, use -ffp-contract=off to disable floating point operation contraction (e.g. FMA)
  - Can also use -mcpu=neoverse-v2, but -mcpu=native will “port forward”
- Use –flto to enable link-time optimization
  - The benefits of link-time optimization vary from code to code, but can be significant [details]
- Apps may need -fsigned-char or -funsigned-char depending on the developer’s assumption.

ARM Journey

Building System Adjustments

- Added a new compile target for aarch64 architecture
  - Mapped compatible compile variables
- Initially used gcc for a “fair” and easier comparison
  - icc and nvc+ as next steps

Component	Version
Base Image	Ubuntu 22.04
Compiler	GCC 12.3
C++ libraries	Boost C++ v1.63.0
MPI	OpenMPI v4.1.4
BLAS	OpenBlas v0.3.20
LAPACK	LAPACK Lib v3.10.0

Architecture	Compiler Flags
x86_64	-std=c++17 -O3 -lrt -fPIC -m64 -march=native -mtune=native -fopenmp-simd -fopenmp
aarch64	-std=c++17 -O3 -lrt –fPIC -mcpu=native -fopenmp-simd -fopen

ARM Journey

Code Porting Adjustment

- Removed Intel Intrinsics
    - Evaluated SIMDe and SSE2NEON as replacements

- Addressed synchronization issues that led to precision errors in floating-point calculation

- Other minor fixes and best practices adjustments

```
#ifndef __aarch64__
#include <xmmintrin.h>
#else
#include <utils/simd/sse2neon.h>
#endif
```

```
ALWAYS_INLINE void P2PNotify( const int task, volatile int * taskFinished )
{
#ifdef __aarch64__
    OPENMP ( omp flush)
    OPENMP ( omp atomic write)
#endif
    taskFinished[ task ] = 1;
}
```

```
ALWAYS_INLINE void P2PWait( const int threadTask, const int * parentIndex,
    const int * parents, const volatile int * taskFinished )
{
    for ( int tIdx = parentIndex[ threadTask ]; tIdx < parentIndex[ threadTask + 1 ]; ++tIdx )
    {
#ifdef __aarch64__
        const int t = parents[ tIdx ];
        int finished;
        do
        {
            OPENMP ( omp flush)
            OPENMP ( omp atomic read)
            finished = taskFinished[ t ];
        } while ( finished == 0 );
#else
        const int t = parents[ tIdx ];
        while ( taskFinished[ t ] == 0 ) PAUSE;
#endif
    }
}
```

ARM Journey

Benchmarking Environment

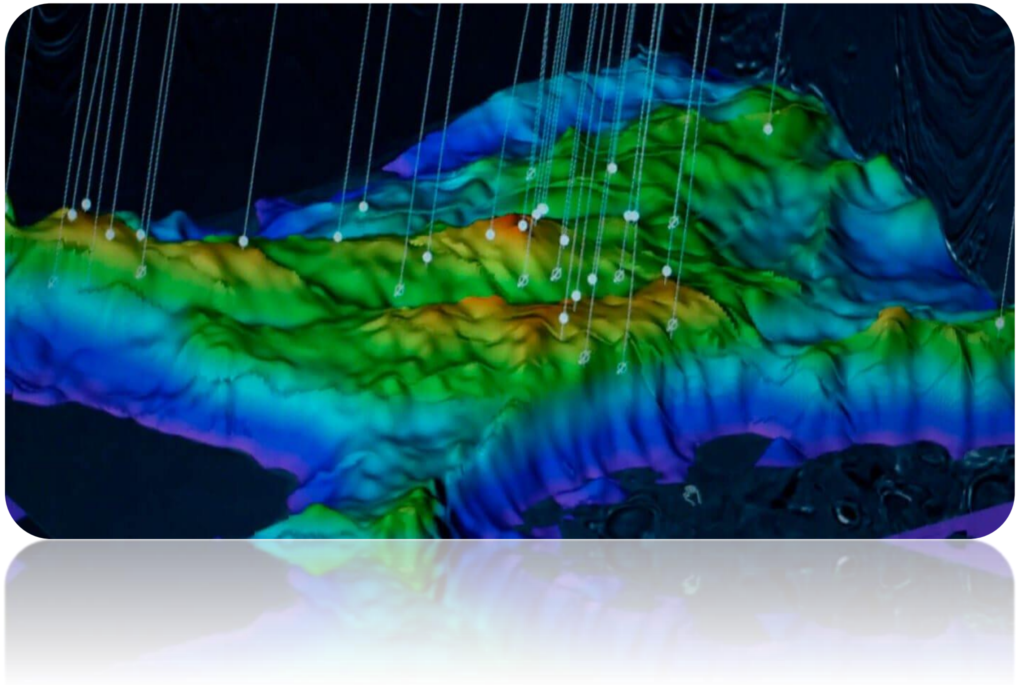
- **Singularity** as the container engine.
- Single definition file, multiple containers.
- One “.sif” file for each CPU architecture.

```
Bootstrap: library
From: ubuntu:22.04

%post
    apt-get update -y
    DEBIAN_FRONTEND=noninteractive apt-get install -y
    --no-install-recommends \
        build-essential \
        ca-certificates \
        cmake \
        cxxtest \
        git \
        libglib2.0-0 \
        htop \
        liblapacke-dev \
        libopenblas-dev \
        libssl-dev \
        locales-all \
        vim \
        wget \
        zlib1g \
        zlib1g-dev
    rm -rf /var/lib/apt/lists/*

# GNU compiler
%post
    apt-get update -y
    DEBIAN_FRONTEND=noninteractive apt-get install -y
    --no-install-recommends \
        g++-12 \
        gcc-12 \
        gfortran-12
    rm -rf /var/lib/apt/lists/*

    ...
```

Results and Discussion

Experiments Cases

- Matrices of 5 reservoirs with 3 different timesteps
  - SPE10
    - 10th SPE Comparative Solution Project
  - Búzios
    - Biggest ultra deep-water oil field of the world
  - Sapinhoá
    - PreSalt reservoir of Santos Basin
  - Proxy 100
    - Semi Synthetic Model with 100x100m cells
      - 6,245,051 active cells
  - Proxy 200
    - Semi Synthetic Model with 200x200m cells
      - 765,620 active cells

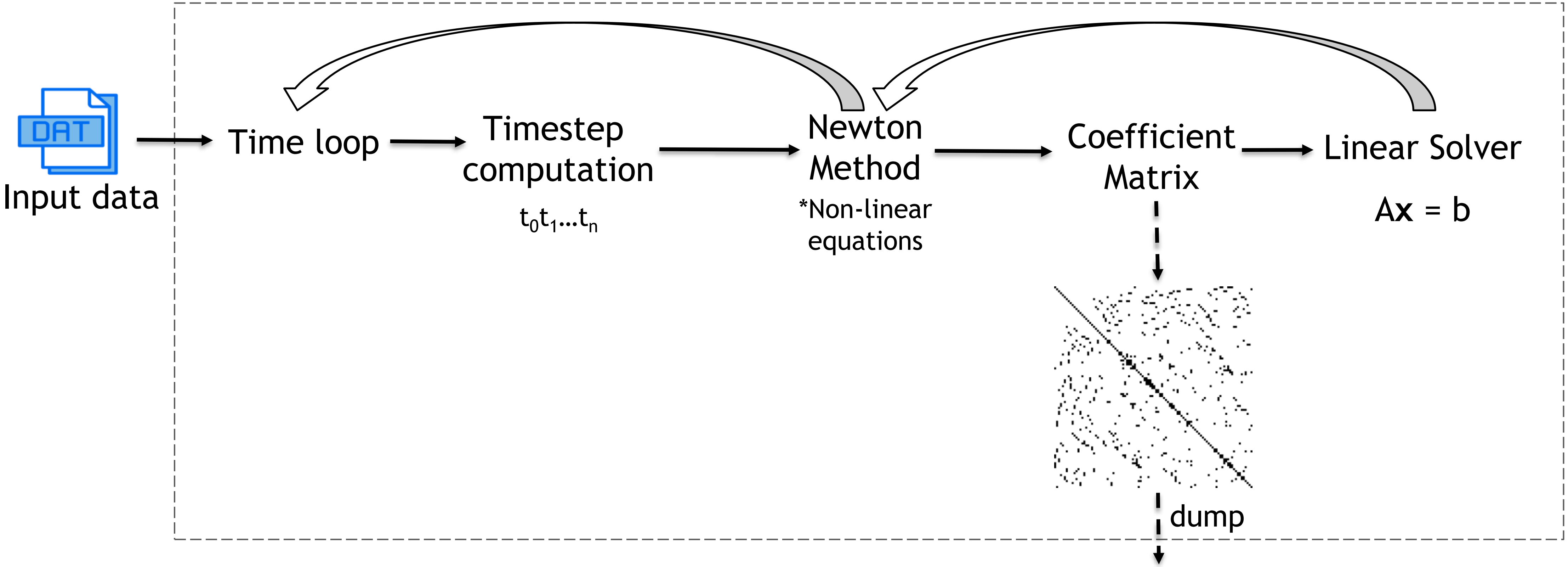


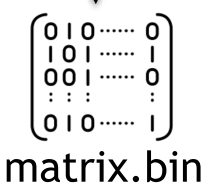
Matrices dumps

Reservoir simulation overview

Compositional & Unconventional Simulator GEM

Black Oil & Unconventional Simulator IMEX



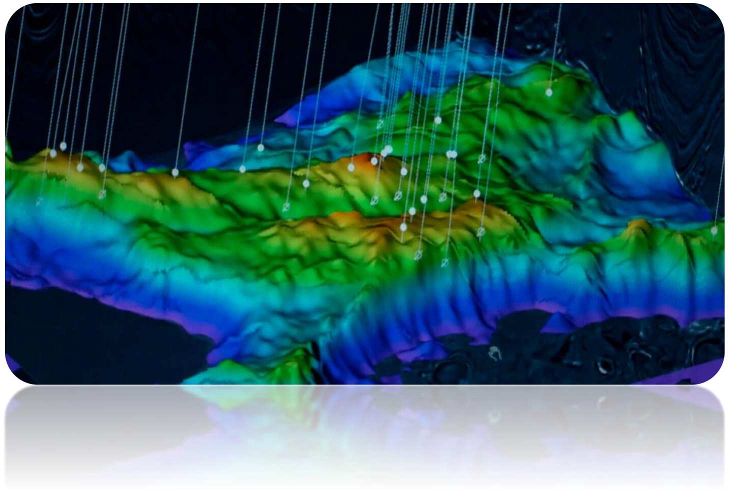


We have used **dumps** at **three arbitrary timesteps**

Matrices characterization

Matrix	# Well Eq.	# IMPES Eq.	# FIM Eq.
Buzios	95	1.32M	51,640
Proxy100	43	6.24M	4,842
Proxy200	0	0.76M	127
SPE10	1	0.90M	187,303
Sapinhoa	19	0.51M	21,195

A high “# FIM Eq." indicates that the matrix has a higher number of unknowns, increasing its complexity

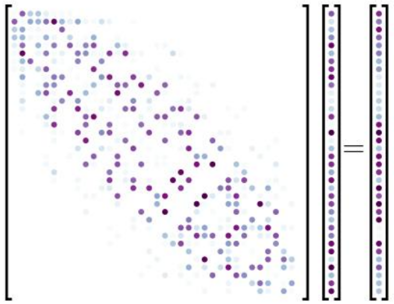


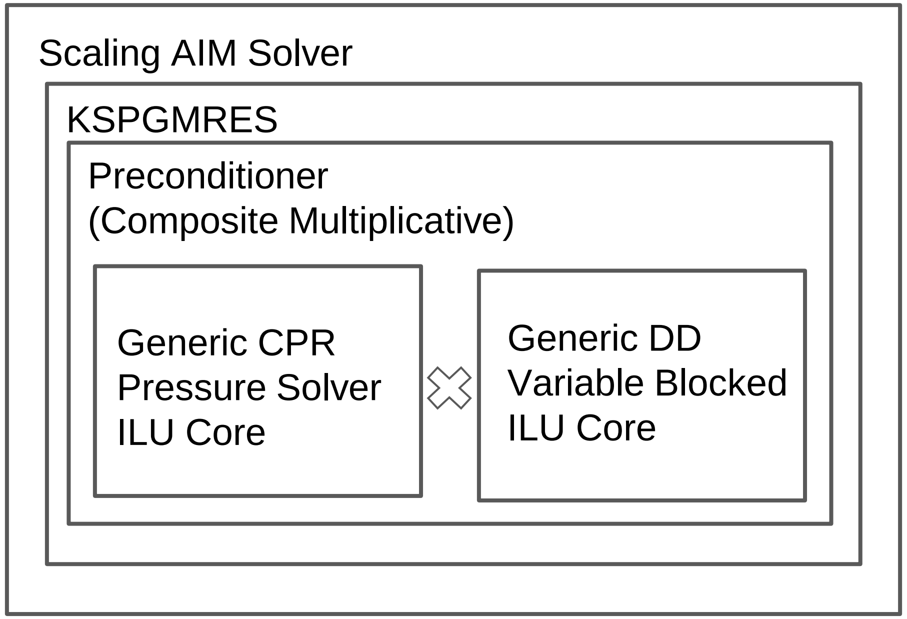
IMPES: Implicit Pressure (1 unknowns)

FIM: Full Implicit (1 + NC unknowns), where NC is the number of components of the simulated fluid

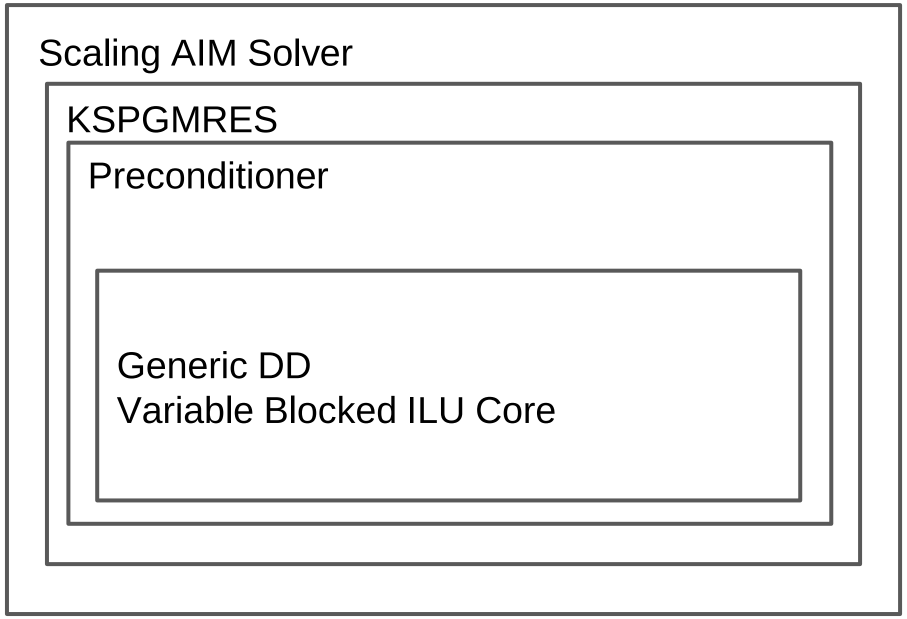
Solver Algorithms

$Ax = b$





SPE10 Matrices



Buzios, Proxy100, Proxy200, and Sapinhoa Matrices

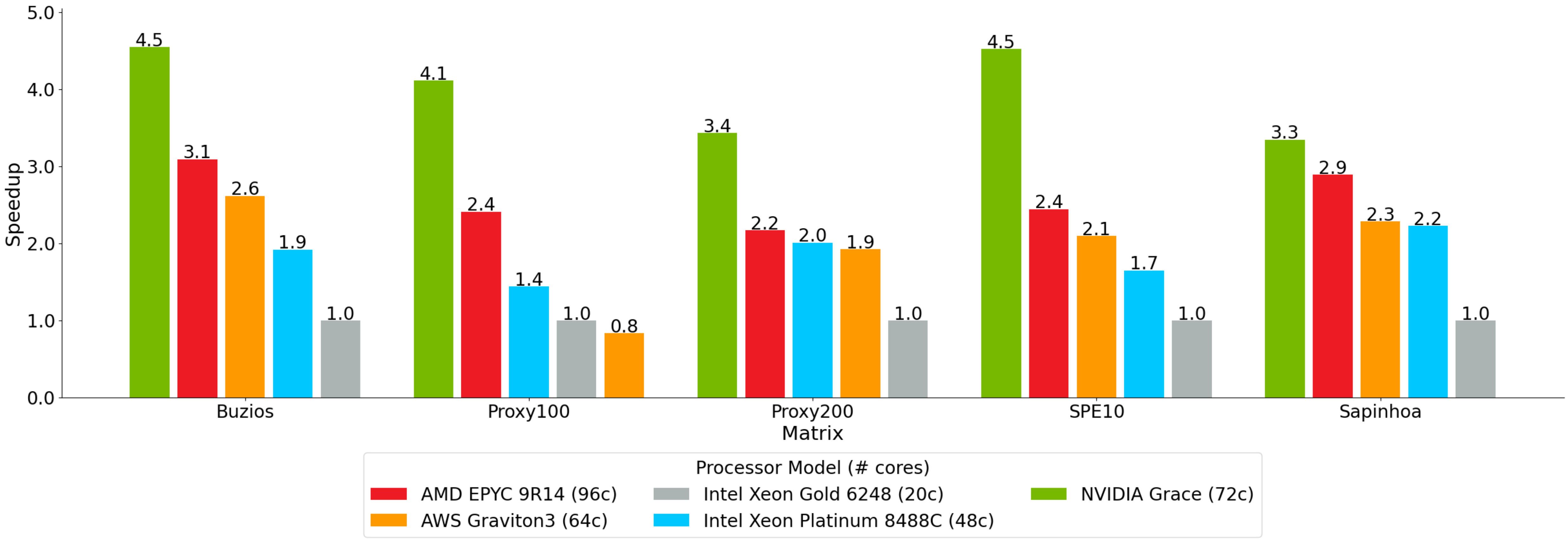
AIM: Adaptive Implicit Method
KSPGMRES: Krylov Subspace Projection Gneralized Minimal Residual
CPR: Constrained Pressure Residual
DD: Domain Decomposition
ILU: Incomplete LU Factorization

Processors Specifications

Environment	Processor	Architecture	Physical Cores
On-premises	**Intel Xeon Gold 6248***	**x86_64**	**20**
AWS EC2 R7g	AWS Graviton3	ARMv8	64
On-premises	NVIDIA Grace	ARMv9	72
AWS EC2 R7i	Intel Xeon Platinum 8488C (Sapphire Rapids)	x86_64	48
AWS EC2 R7a	AMD EPYC 9684X (Genoa)	x86_64	96

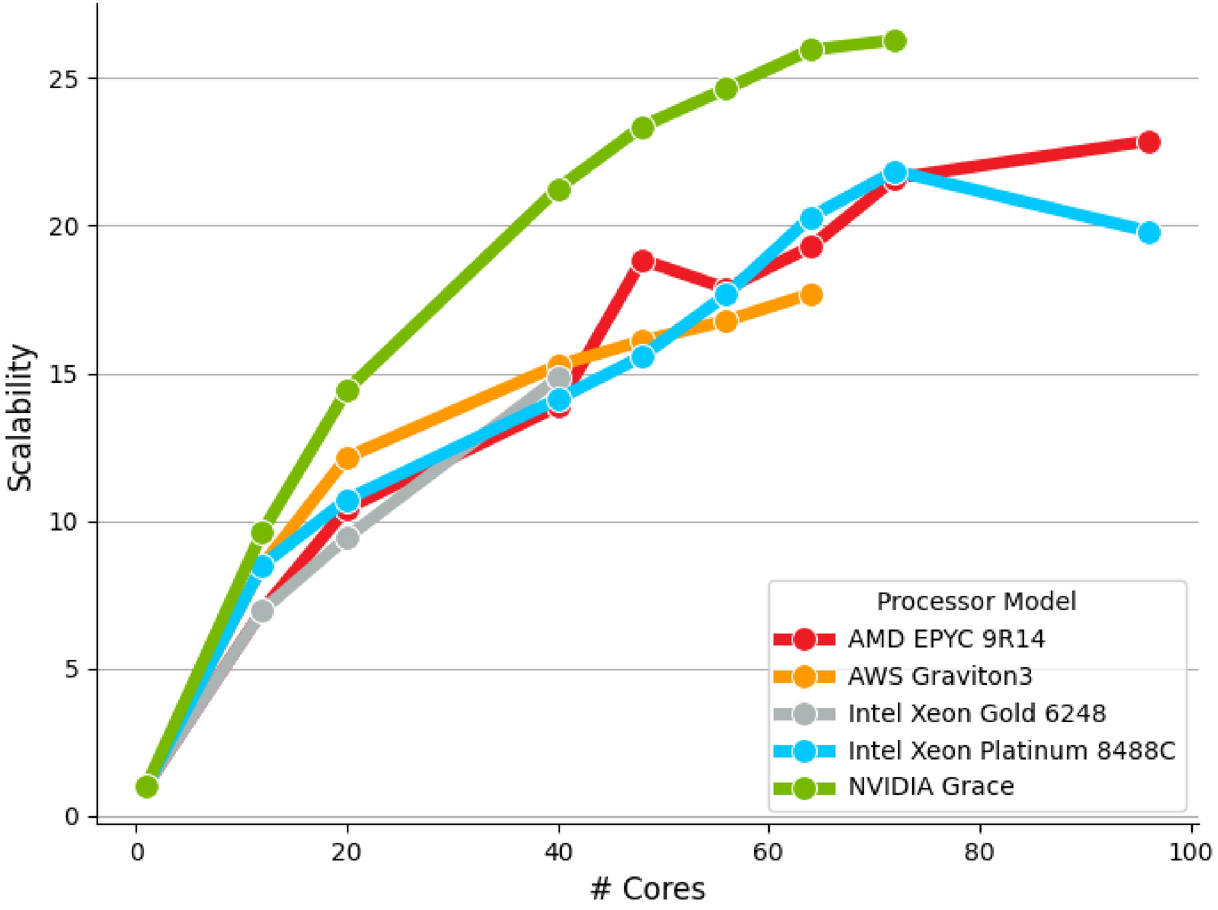
* *Intel Xeon Gold 6248 is currently the main on-premises CPU cluster Petrobras Research Center (CENPES).*

Single-socket Speedups (max core-count)



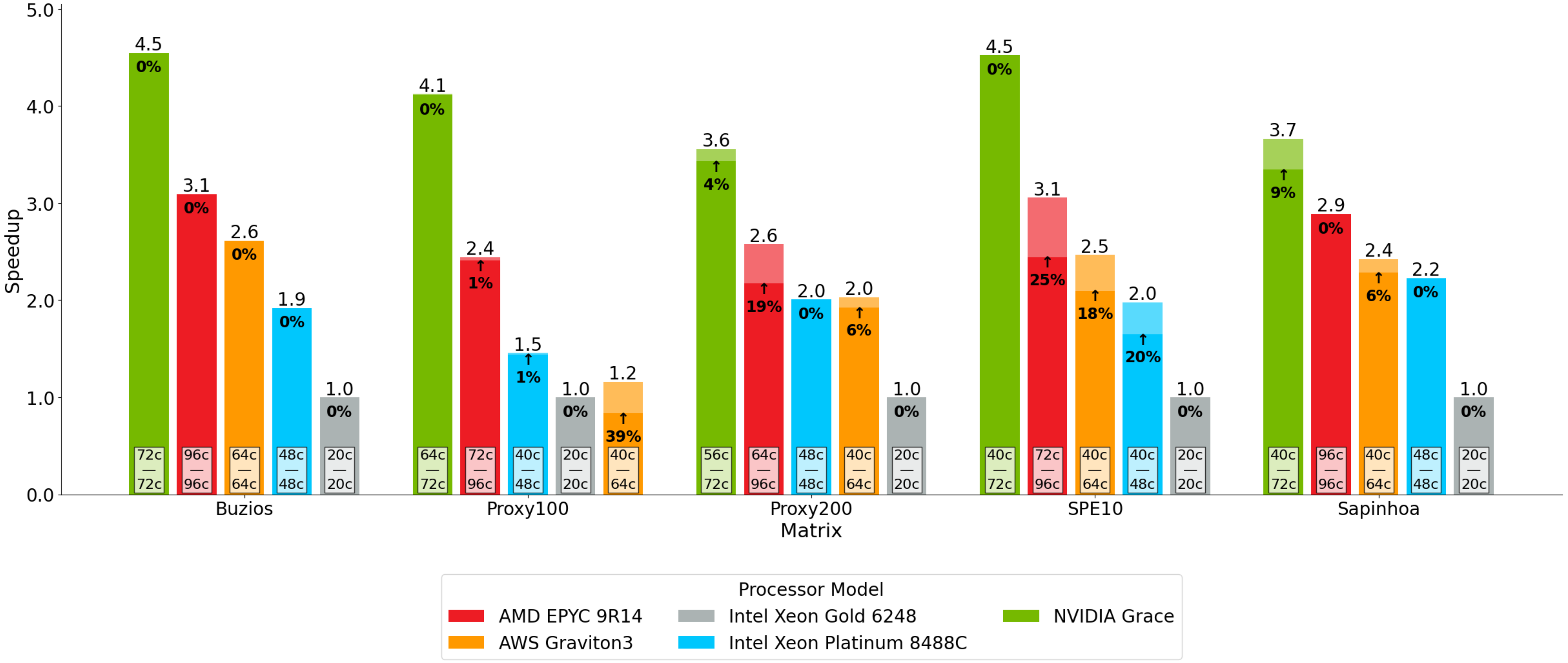
Average of 150 executions per matrix (considering 3 different timesteps)

Single-socket Solver Scalability – Búzios



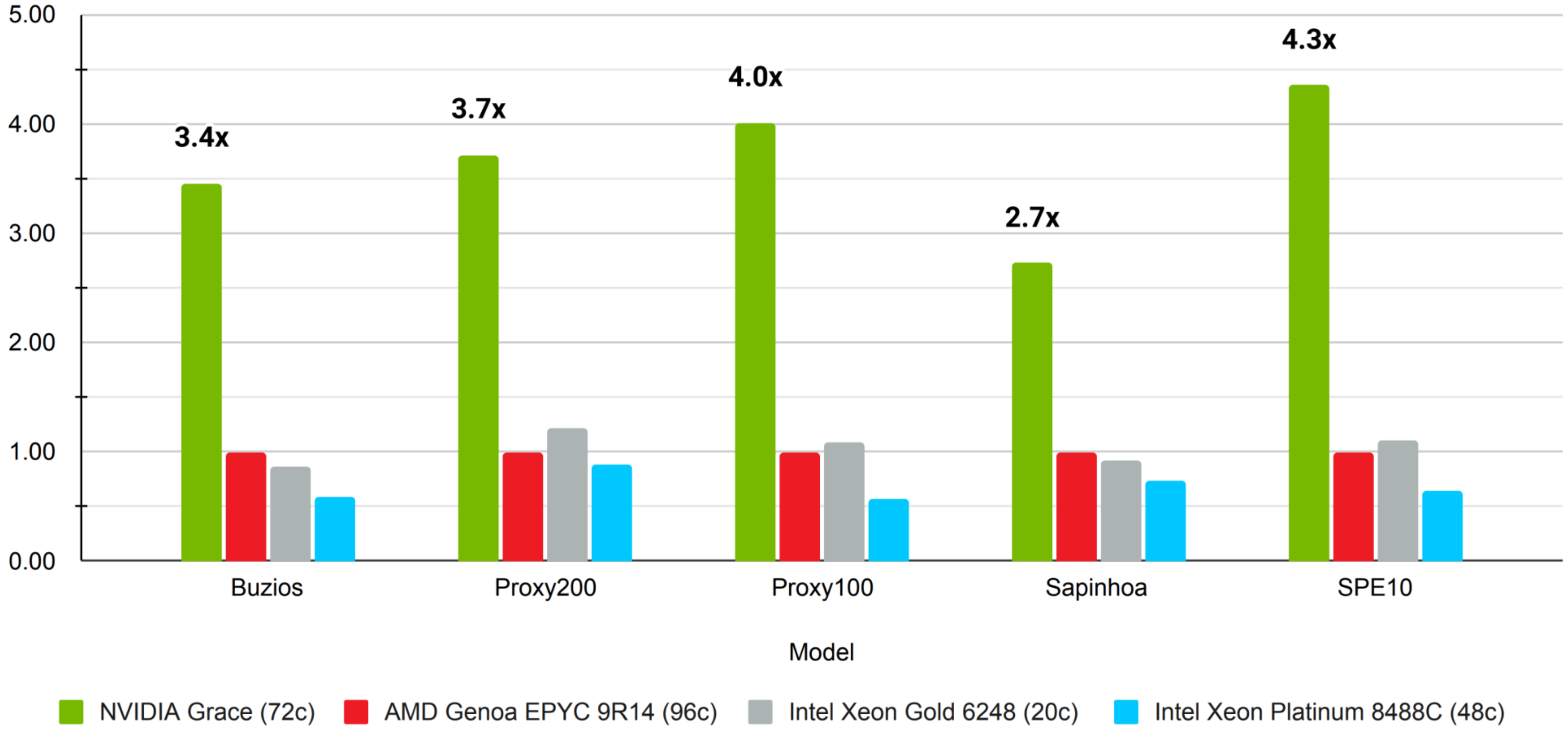
The variation in the number of iterations ranged from 0 to 15% among all executions of the same reservoir model on a given processor.

Best Core-count Performance



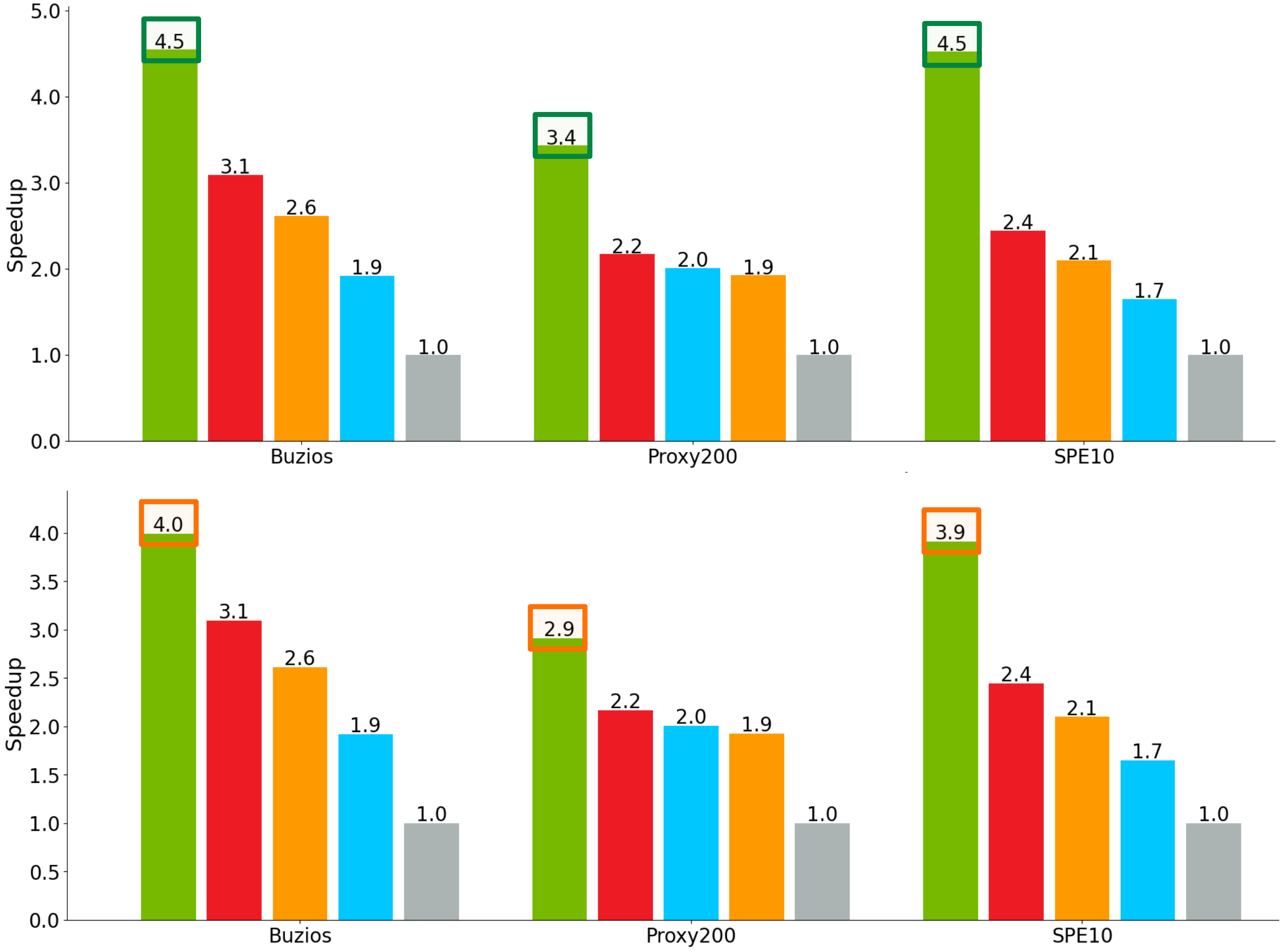
**Average of 150 executions per matrix (considered 3 different timesteps)*

Estimated Energy Efficiency at Max Load



Estimated are computed considering CPU max TDP at full load plus estimated memory consumption based on capacity and technology. We assume an average of 3W per 8 GB for DDR4 and 4.75W per 16 GB for DDR5. For Grace SoC, CPU plus memory is ~250W. Speed-ups are computed using AMD EPYC 9R14 as baseline.

Same Soc, Different Memory Capacity



Superchip	Capacity (GB)	OMP_NUM_THREADS	Expected TRIAD Bandwidth
Grace-Hopper	120	72	450+
Grace-Hopper	480	72	340+
Grace CPU	240	144	900+
Grace CPU	480	144	900+
Grace CPU	960	144	680+

https://docs.nvidia.com/grace-performance-tuning-guide.pdf

Conclusions

Best ranked processors for the evaluated reservoir models

Processor	Búzios	Proxy100	Proxy200	SPE10	Sapinhoá
NVIDIA Grace	1	1	1	1	1
AMD EPYC 9R14	2	2	2	2	2
AWS Graviton3	3	4	3	3	3
Intel Xeon Platinum 8488C	4	3	4	4	4
Intel Xeon Gold 6248	5	5	5	5	5

*This ranking considers the best performing processors for each evaluated matrix

Conclusions

Best ranked processors for the evaluated reservoir models

Processor	Ranking
NVIDIA Grace	1
AMD EPYC 9R14	2
AWS Graviton3	3
Intel Xeon Platinum 8488C	4
Intel Xeon Gold 6248	5

*This ranking considers the best performing processors for each evaluated matrix

Conclusions & Future Works

- ARM surprised with better performance, even without code optimization
- NVIDIA Grace gives the best out-of-the-box results among all tested processors

Next Steps

- Port the end-to-end simulators
  - GeomecBR porting in process
- Evaluate code optimizations for ARM
  - Proofing and Tuning such as done for Intel® processors
- Evaluate other computer architectures, such as Grace Hopper
- Explore half-precision mixed calculation

Broader Team Effort

Thank you, Cesarianos!

- **Luigi Fernando**
  - *Developer*
- **Vitor Aquino**
  - *Developer*
- **Fabiane Castro**
  - *Agile Master*

Thank you, NVIDIANs!

- **Filippo Spiga**
  - *Technical Product Manager, Accelerated Compute Workloads and Performance*
- **John Linford**
  - *Principal Technical Product Manager, Datacenter CPU Software*
- **Giri Chukkapalli**
  - *NVIDIA Distinguished Engineer*
- **Daniel Ruiz**
  - *DevTech Engineer*

References

- FIGUEIREDO, Mateus O., SOUSA JUNIOR, Luis Carlos, RODRIGUES, Jose Roberto P., SANTOS Leandro B., GASPARINI, Leonardo da Silva, AMARAL, Ricardo F., MORAES e Rafael J. “**A parallel viscoplastic multiscale reservoir reservoir geomechanics simulator.**” Upstream Oil & Gas Technology, Elsevier, Volume 11, Setembro, 2023.

- GASPARINI, Leonardo da Silva, RODRIGUES, Jose Roberto P., CONOPOIMA, Cesar, AUGUSTO, Douglas A., SOUZA, Michael, CARVALHO, Luiz M., GOLDFELD, Paulo, RAMIREZ, João P., PANETTA, Jairo. “**A linear solver framework for flow and geomechanical reservoir simulation.**” In: 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), IEEE Computer Society, 2019.

- PORTELLA, Felipe A., ESTRELA, Paulo J. B., MALINI, Renzo Q., TEYLO, Luan, BERRAL, Josep L. Berral, and DRUMMOND, Lúcia. “**MScheduler : Leveraging Spot Instances for High-Performance Reservoir Simulation in the Cloud.**” In: IEEE CloudCom (2023). doi: 10.1109/CloudCom59040.2023.00028.

Questions / Contact

Felipe Portella

felipeportella@petrobras.com.br

linkedin.com/in/felipeportella

João Paulo Navarro

jpnavarro@nvidia.com

linkedin.com/in/jppnavarro