FUJITSU

shaping tomorrow with you

Reservoir kernel code evaluation on A64FX

September 16th, 2021

Simone Rinco - Soham Sheth
Schlumberger

Fabrice Dupros
Arm ltd.

Takayuki Hoshiya
Fujitsu ltd.

Agenda

FUJITSU

- Reservoir simulation
  - Fundamental equations
  - Discretization
  - Evaluation model
  - kernel code structure
- Performance on A64FX
  - Tuning tips
- Take away/Future works

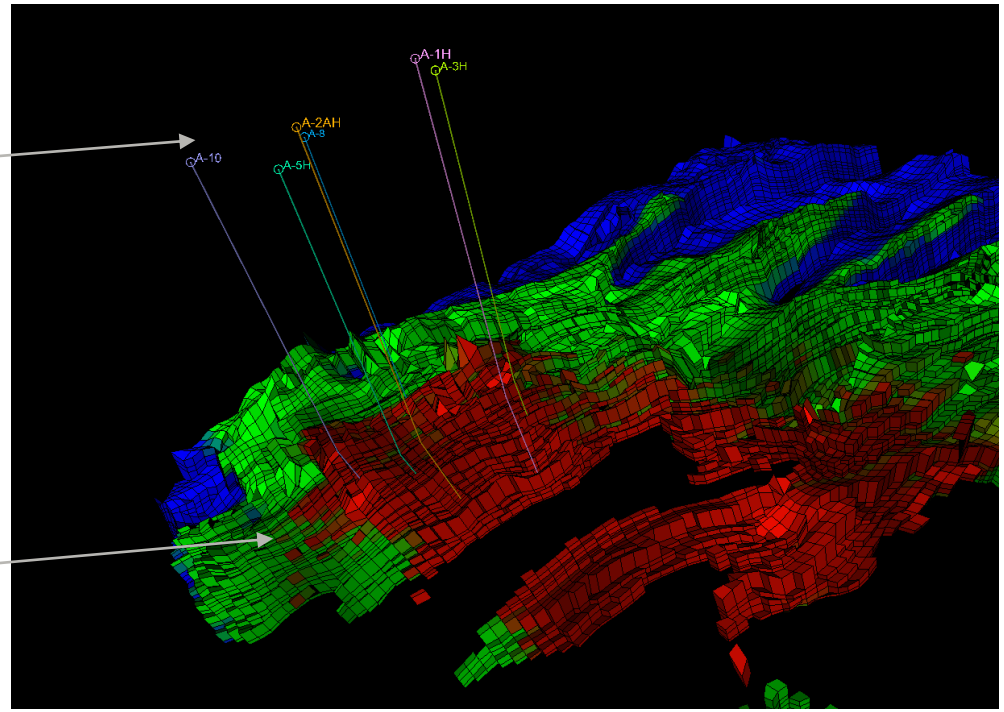
Reservoir Simulation

FUJITSU

Wells
(producers/injectors)

Reservoir

Gas
Oil
Water



Petrel

Reservoir Simulation: predict the behaviour of the **reservoir** and the **wells** according to a specific strategy (well control).

Schlumberger

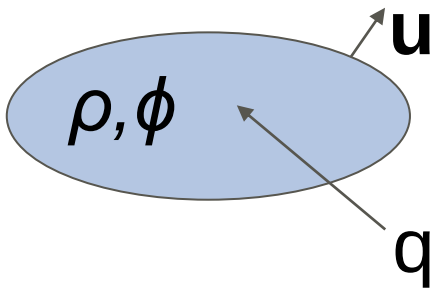
ECL ECLIPSE

IX INTERSECT

Reservoir Simulation

FUJITSU

- Two fundamental equations
  - Conservation of mass:



Control volume is arbitrary

Fluid density

Rock porosity

Fluid speed

$$\frac{\partial}{\partial t}(\rho\phi) + div\,(\rho\mathbf{u}) = q \qquad \left[\frac{Kg}{m^3 s}\right]$$

Mass injected (wells)
Negative for producers

  - Darcy's law:

Rock permeability

$$\mathbf{u} = -\frac{\mathbf{K}}{\mu}(\nabla p - \rho g \nabla D) \qquad \left[\frac{m}{s}\right]$$

Fluid viscosity

Fluid pressure

Depth

Schlumberger

Reservoir Simulation

FUJITSU

- Combining the two equations:

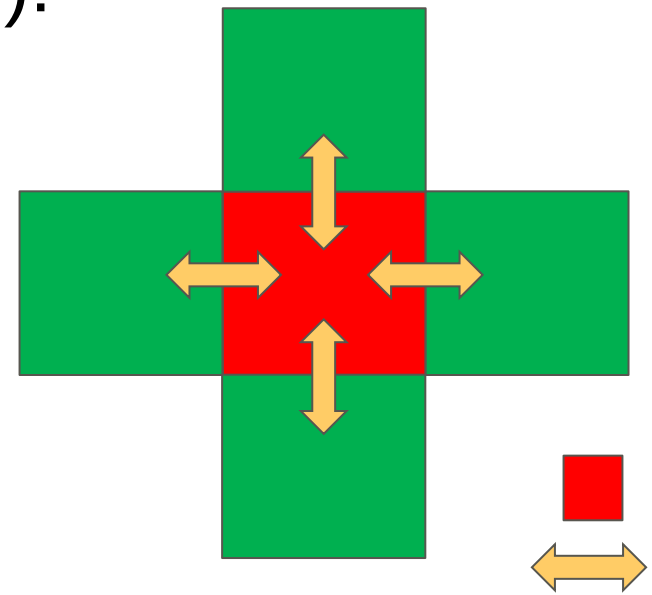
$$\underbrace{\frac{\partial}{\partial t}(\rho\phi)}_{\text{Mass accumulation}} - \underbrace{div\left(\rho\frac{\mathbf{K}}{\mu}(\nabla p - \rho g \nabla D)\right)}_{\text{Mass flux}} = \underbrace{q}_{\text{Well(s) contribution}}$$

One equation per component:
gas, oil, water (3 equations).
1 extra equation for thermal models

- Discretisation for the control volume (cell):
  - Mass accumulation:

    depends only on control volume
  - Mass flux:

    depends on control volume and neighbours
  - Wells: [ignored]

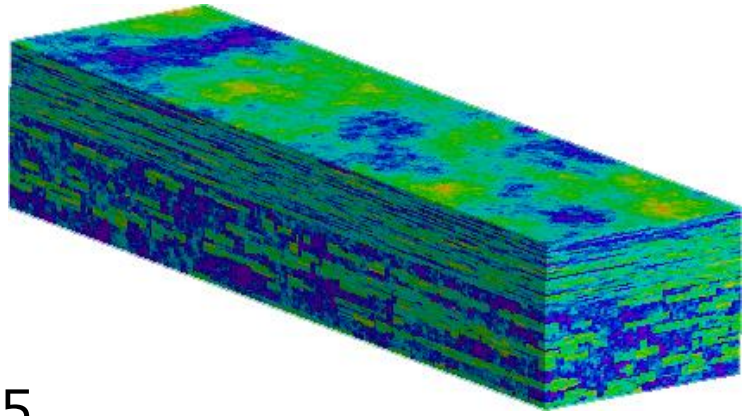


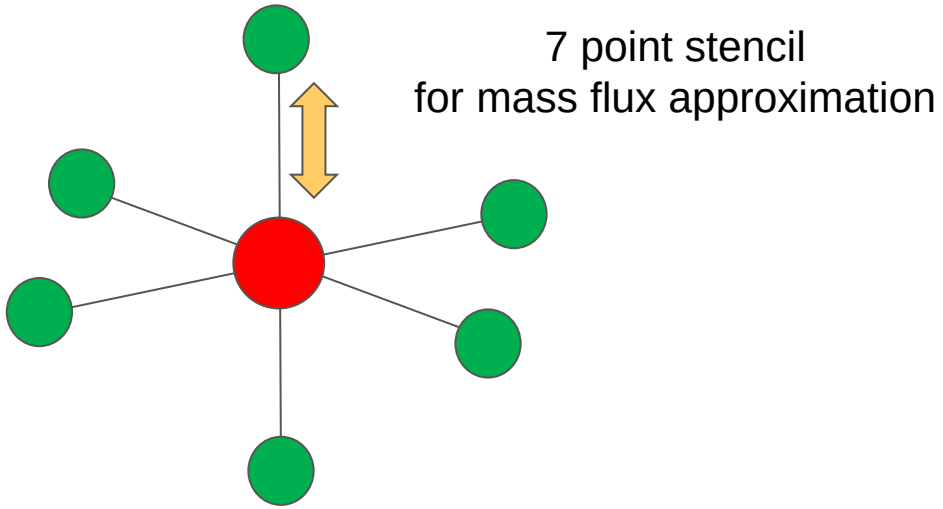
Schlumberger

Evaluation model

FUJITSU

- SPE10 model 2
  https://www.spe.org/web/csp/datasets/set02.htm
- 60 x 220 x 85 cells
- Only the topology (connections) data is used
- Original model is oil-water (2 equations). The mock code assumes gas is present as well (3 equations)
- For each time step, mass accumulation and flux are calculated 5 times. This simulates an execution of 5 Newton iterations.



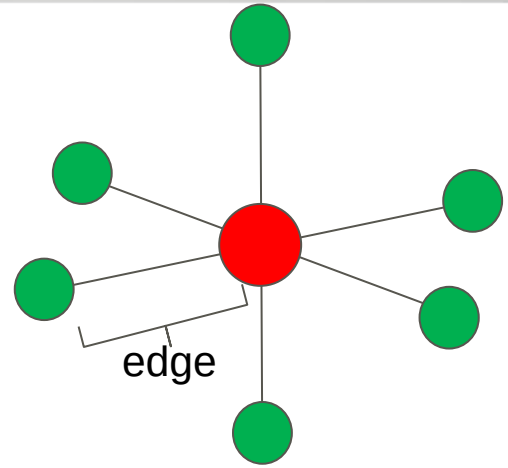


Schlumberger

kernel code structure (mass flux)

FUJITSU

```
For each time step:
{
  For each Newton iteration (hardcoded 5 iterations):
  {
    For each edge (Parallelized by OpenMP):
    {
      - get source (src) and target (tgt) cell
      - get pressure in src and tgt cells and calculate pressure difference
      - get other properties in src and/or tgt cell
      For each equation (3 or 4):
      {
        calculate mass flux and update mass balance equation in
        src and tgt cell
      }
    }
  }
}
```



Schlumberger

Kernel loop performance (summary)

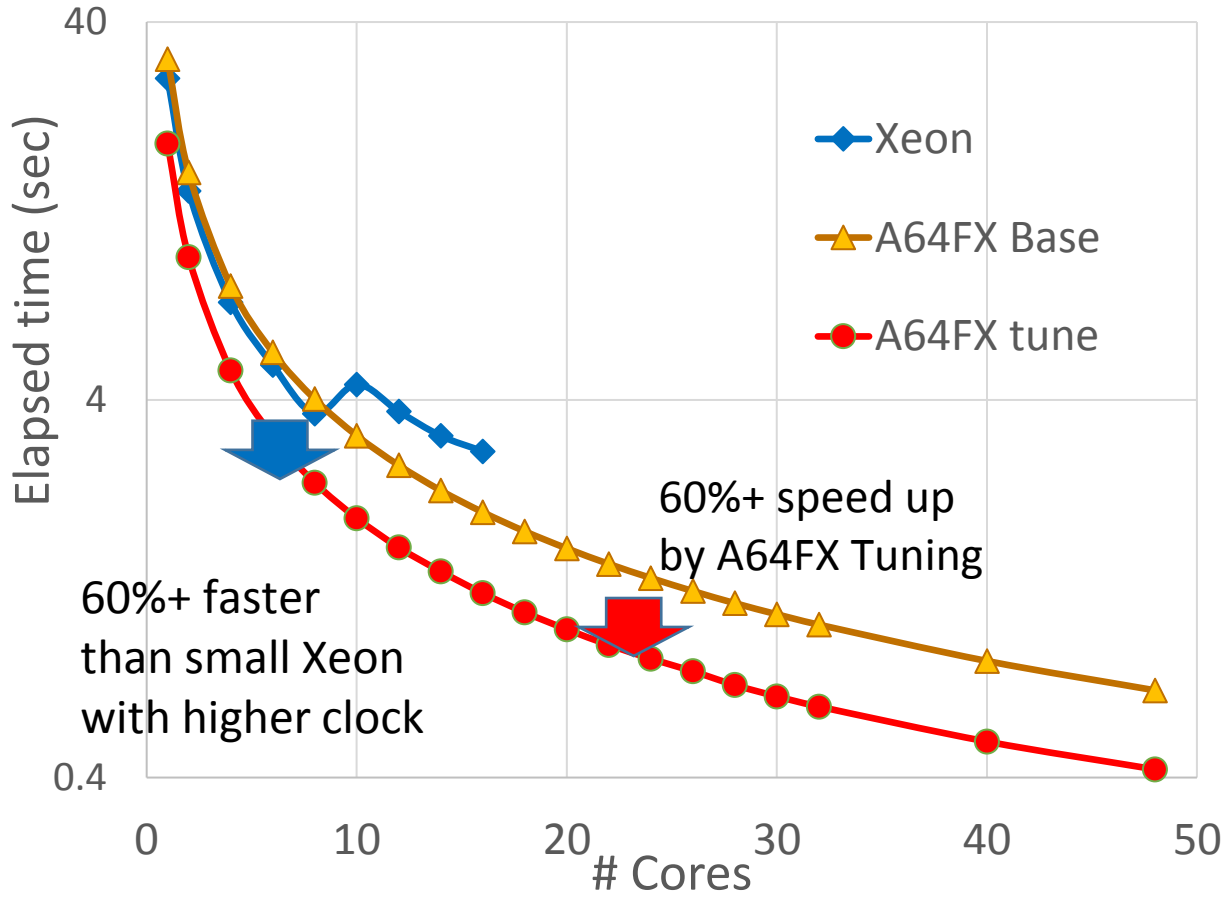
FUJITSU

Conditions

- A64FX 1 node = 48cores (2.2GHz) 32GB(HBM2), Fujitsu compiler, OpenMP, numactl (interleave)
- Reference Xeon Cascade Lake 4cores/8threads x 2 sockets (3.8GHz), 128GB(DIMMx4), gcc 10.2, OpenMP, numactl(interleave)

Performance summary

- A64FX shows similar performance with 1-8cores Xeon even though A64FX has 60% lower clock rate. A64FX performance scales up to 48 cores
  - A64FX module used SVE for small iteration in inner loop(3).
  - Gcc generated SSE code (not AVX) for this small iteration.
- By tuning A64FX assembly code, A64FX speed up 60%+.
  (60% faster than small Xeon CPU even 60% lower clock)

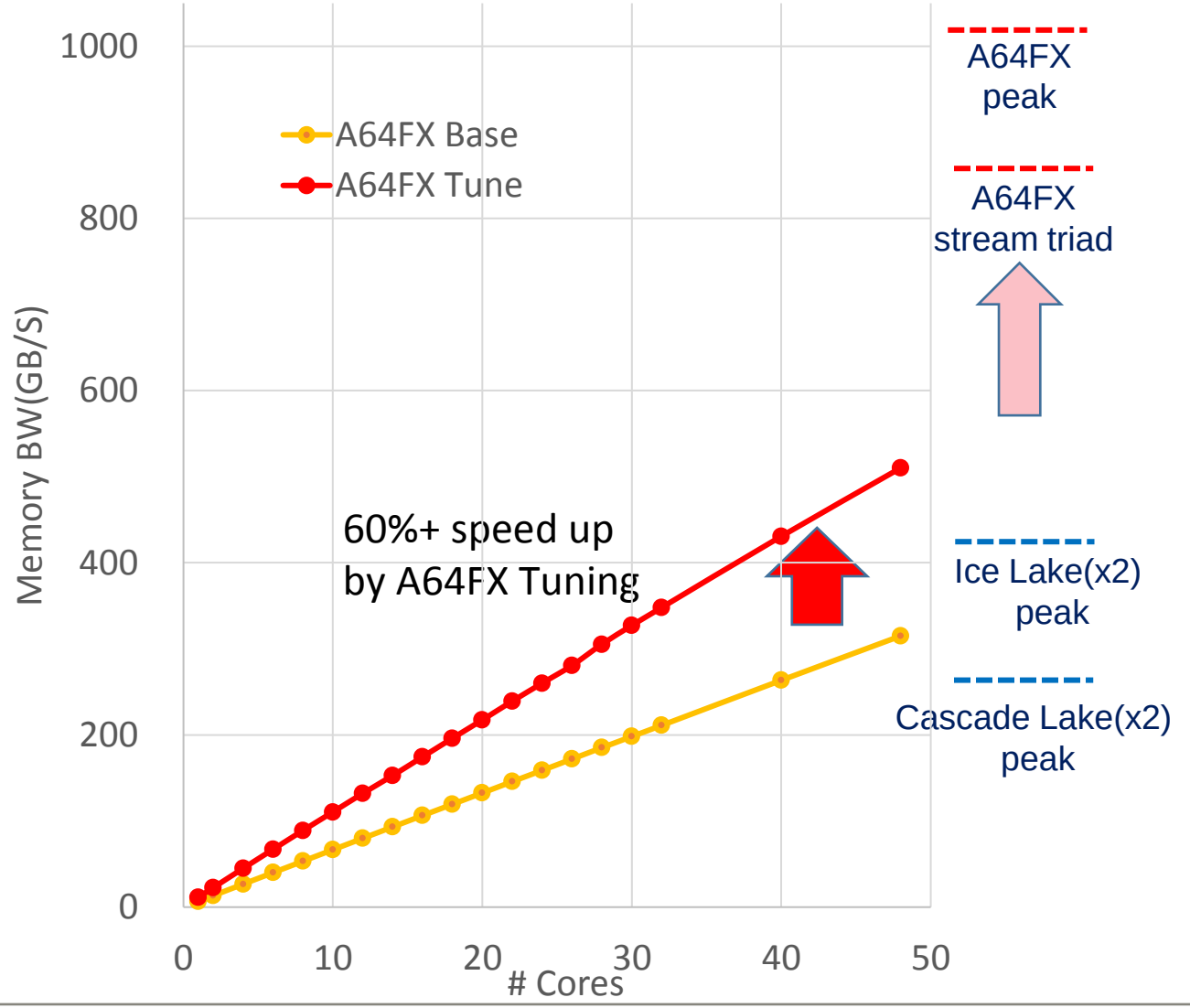


Kernel loop performance (brief analysis)

FUJITSU

Memory Bandwidth

- A64FX performance scales up to 48 cores.
- A64FX base performance was 30% of peak performance(1024GB/s). Memory latency looks bottle neck for these lower efficiency
- By the tuning, we could derive more than 50% of peak performance.
- We might have room to improve this performance in the viewpoint of memory bandwidth.



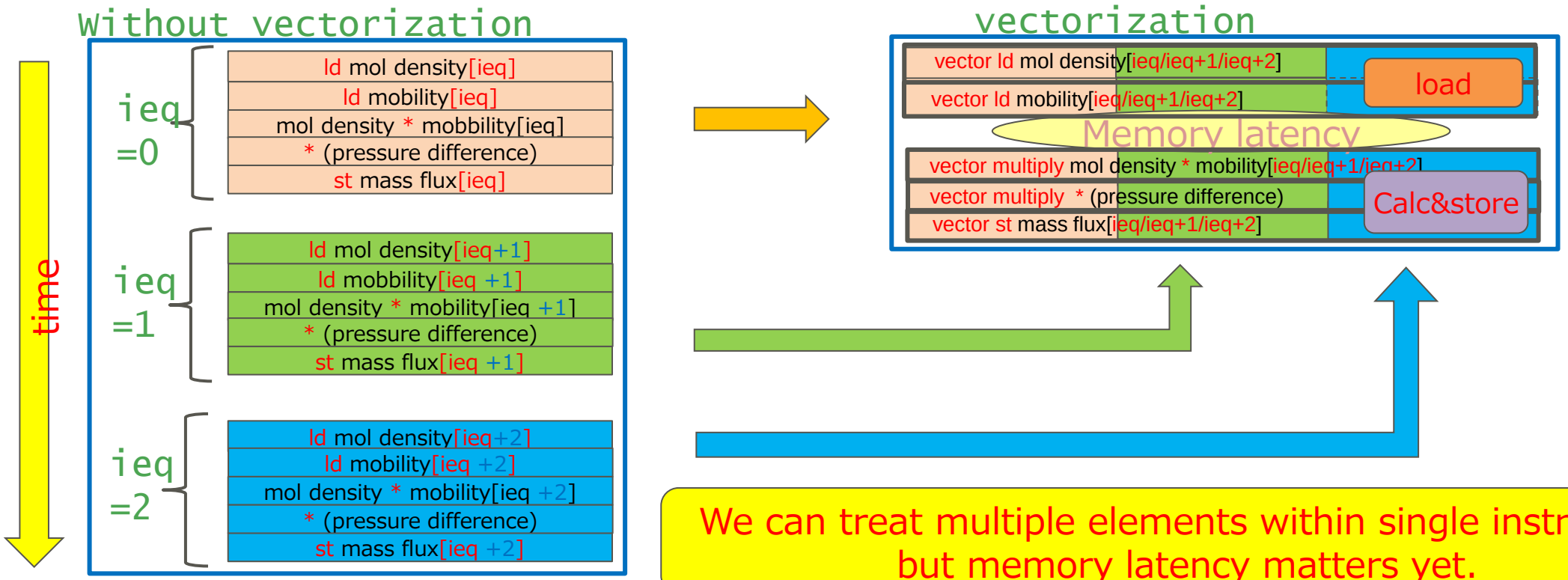
Example of kernel loop vectorization

FUJITSU

Vectorize inner loop

Predicate (mask) register in SVE enables vectorization
even 3 iterations within 512bit vector register

```
for each equation[ieq] (3 in this slide)
  mass flux[ieq] = mol density[ieq] * mobility[ieq]) * (pressure difference)
  (mass update follows)
```



We can treat multiple elements within single instruction
but memory latency matters yet.

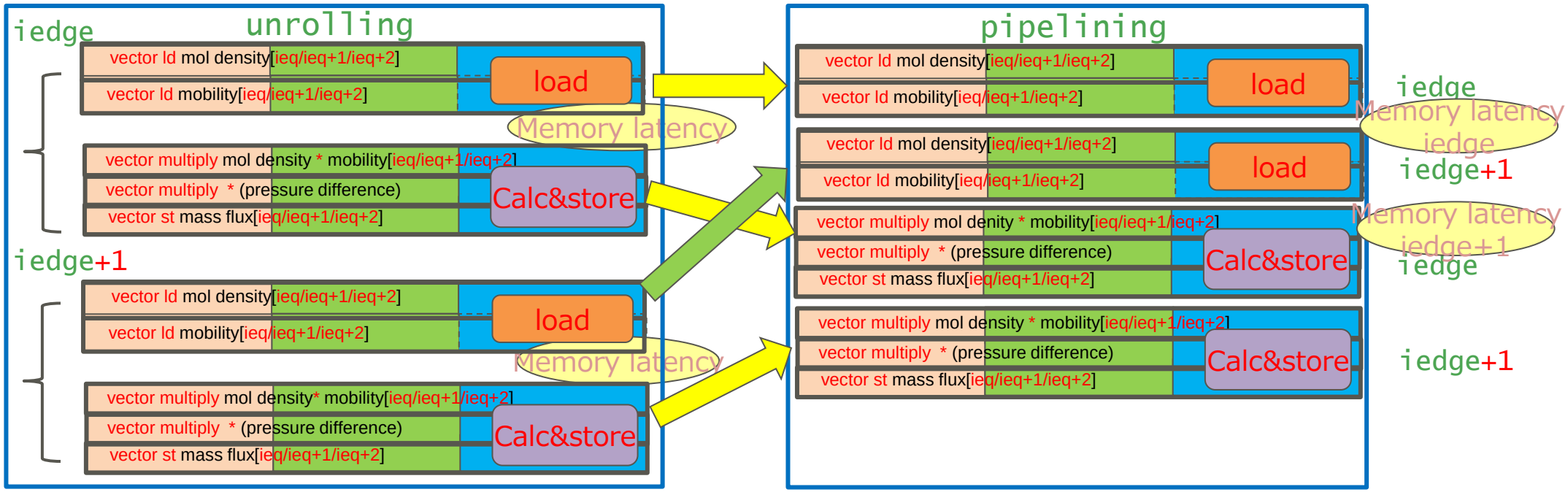
Unrolling outer loop to hide memory latency

FUJITSU

Loop unrolling is effective to hide memory latency in this kernel loop

```
for each edge [iedge] // unroll outer loop
  for each equation [ieq] // 3 iteration in this model // inner loop (vectrized)
     mass flux[ieq] = mol density[ieq] * mobility[ieq]) * (pressure difference)
  (mass update follows)
```

Memory latency could be hidden by moving load for iedge + 1



Take away/Future work

FUJITSU

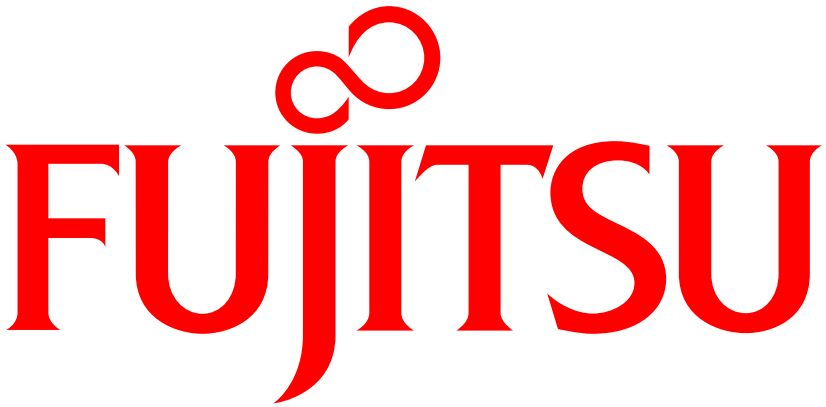
- A64FX shows good performance with SVE(Scalable Vector Extension) and HBM(High Bandwidth Memory)
  on a reservoir kernel code.
- We needed hand assembly tuning to derive A64FX hardware potential as of now, because inner loop has small iteration.

- We'll proceed more practical application evaluation (hopefully without hand assembly tuning)

Acknowledgement

The authors would like to acknowledge the assistance of this evaluation by;

- Schlumberger
  - Eamon Dodds
- Arm
  - Daniele Piccarozzi
- Fujitsu
  - Yoshio Nakao
  - Pierre Lagier



shaping tomorrow with you