allinea

High performance tools to debug, profile, and analyze your applications

Weather and Climate Models: Preparing Development Workflows for Exascale

Florent Lebeau
flebeau@allinea.com

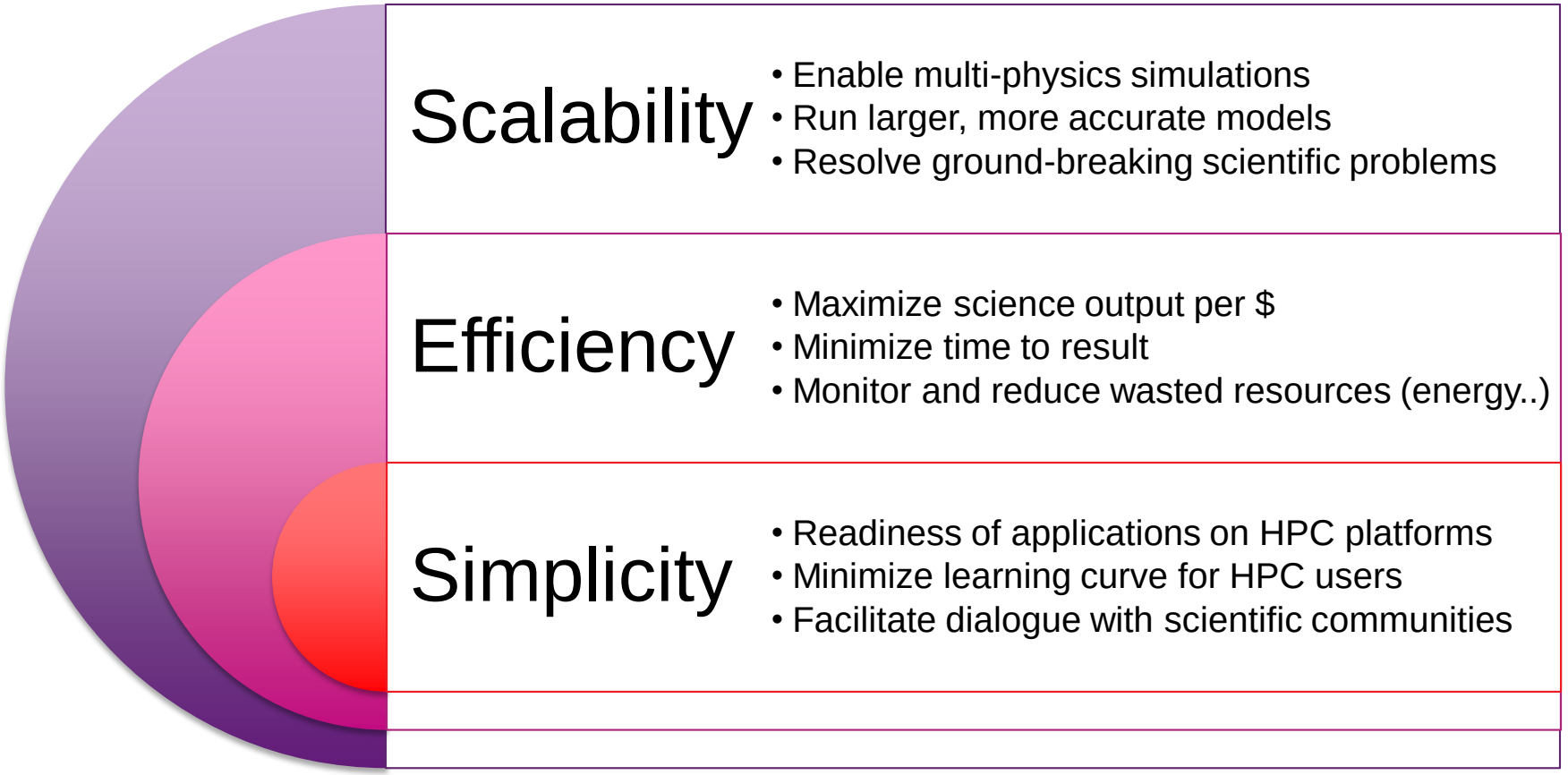
Outline

- How to Handle Increasingly Complex Models?
- Allinea's Tool Solution
- Automate Fault Detection of Weather and Climate Models
- What is coming next

Weather and Forecasting models

allinea

As the complexity increases, the demands are evolving



allinea

Allinea's vision

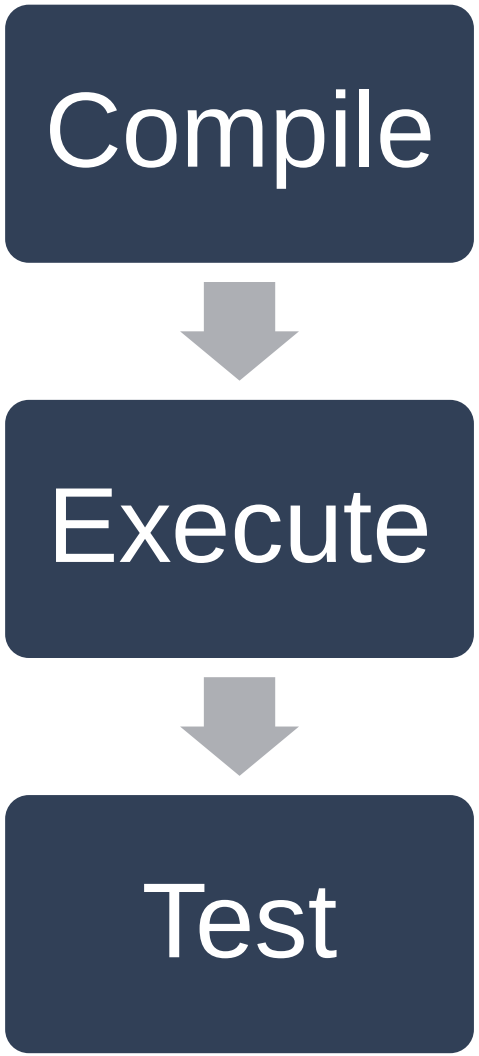
- **Helping maximize HPC production**
  - Reduce HPC systems operating costs
  - Resolve cutting-edge challenges
  - Promote Efficiency (as opposed to Utilization)
  - Transfer knowledge to HPC communities

- **Helping the HPC community design the best applications**
  - Reach highest levels of performance and scalability
  - Improve scientific code quality and accuracy

allinea

Automation script example

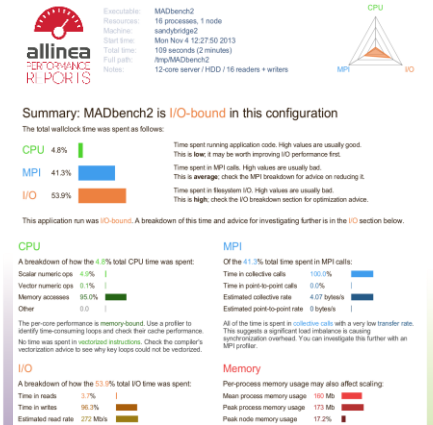
```
#!/bin/bash –l
# Job submission file name
jobfile=test_jacobi_mpi_omp_gnu.sub
# Load environment
module load compiler/gnu mpi/openmpi_gnu
module load allinea/perf-report
# Compile
make clean && make

# Job submission file configuration
cat << EOF > $jobfile
#!/bin/bash –l
#SBATCH --job-name=‘test_jacobi_mpi_omp_gnu‘
#SBATCH --time=00:05:00
#SBATCH --ntasks=128
#SBATCH –ntasks-per-node=2
export OMP_NUM_THREADS=16
srun ./jacobi_omp_mpi_gnu.exe
EOF

# Submit
sbatch $jobfile

# Check results
[…]
```



Performance monitoring

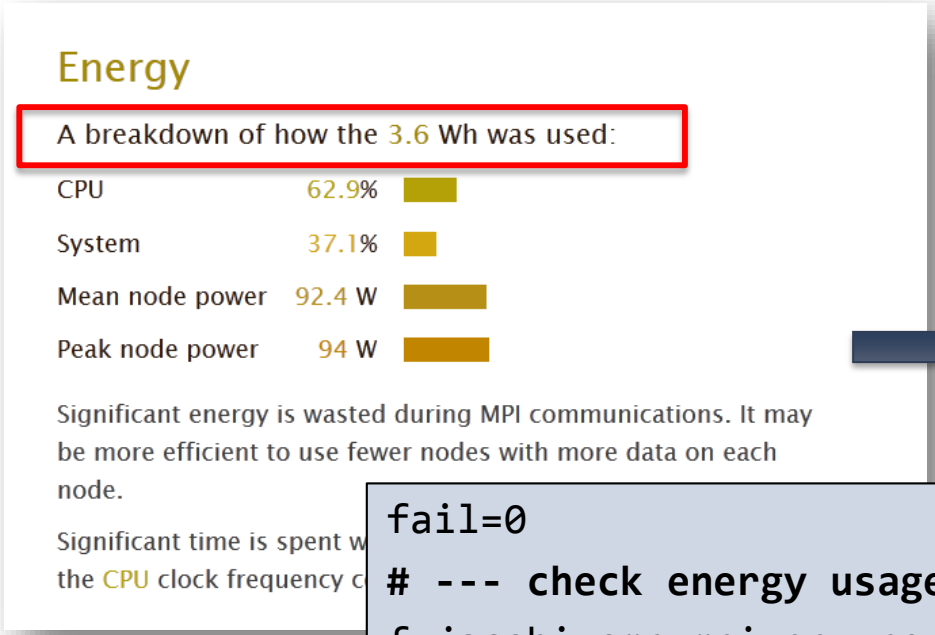
```
perf-report -o jacobi_omp_mpi_gnu_perf.csv \
    srun ./jacobi_omp_mpi_gnu.exe
```

- -o specifies the name and format of the output
  - Html
  - Txt
  - CSV



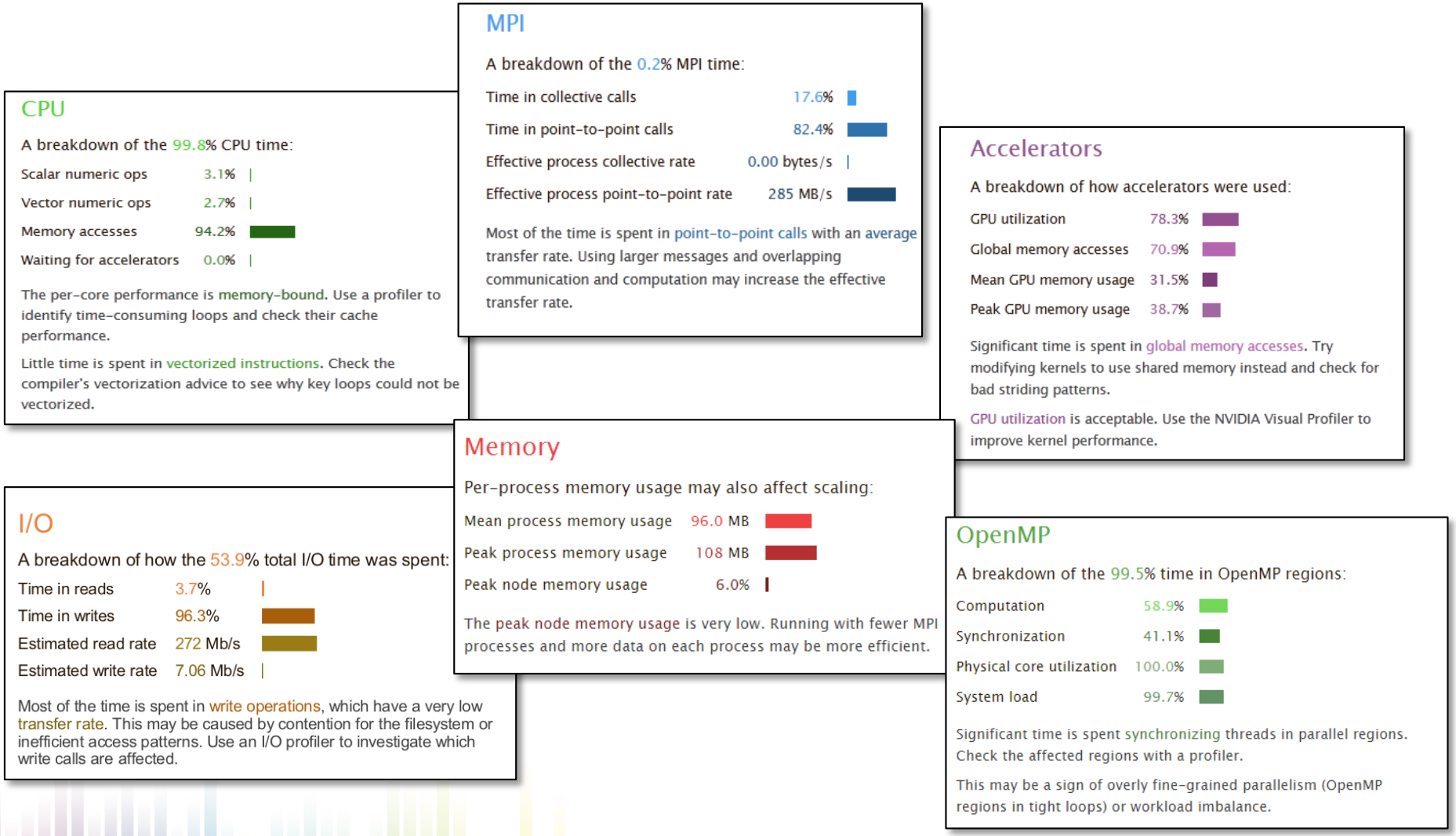
Name	Value
Executable	jacobi_omp_mpi_gnu.exe
Command	srun ./jacobi_omp_mpi_gnu.exe
Processes	120
Nodes	64
Physical cores per node	16
Logical cores per node	32
Memory per node (GiB)	32
Machine	mars
Started on	Wed Sep 28 17:04:42 2016
Total time (s)	1534

Energy efficiency monitoring




```
fail=0
# --- check energy usage
f=jacobi_omp_mpi_gnu_perf.csv
tot_energy=`grep “Total energy”|awk -F, '{print $2}'`
if [ "$t" > “3000" ] ; then
         ((fail++))
        echo "Test has failed: Total energy =$tot_energy“
else
    echo “Test has succeeded”
fi
```

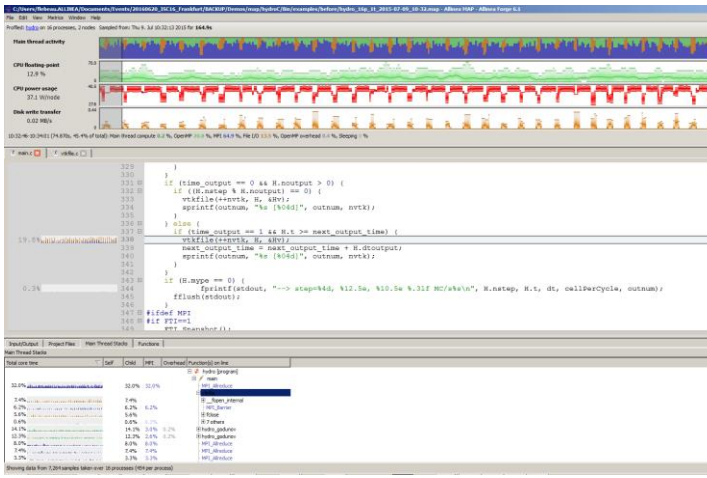
Efficiency monitoring



Automate profiling

```
map --profile -o jacobi_omp_mpi_gnu_perf.map \
    srun ./jacobi_omp_mpi_gnu.exe
perf-report -o jacobi_omp_mpi_gnu_perf.csv \
    jacobi_omp_mpi_gnu_perf.map
```

- -o specifies the name of the output
- The output can be turned into a report with Allinea Performance Reports for pre-processing
- The output can be open for afterwards for further investigation:
  - On the login node using X forwarding with Allinea MAP
  - Or copied locally and using the remote client
    - Linux, Windows and MacOS X builds
    - http://www.allinea.com/products/forge/download



Automate debugging

```
ddt --offline --output=jacobi_omp_mpi_gnu_debug.txt \
    --trace-at _jacobi.F90:83,residual \
    srun ./jacobi_omp_mpi_gnu.exe
```

- --offline enable non-interactive debugging
- --output specifies the name and output of the non-interactive debugging session
  - Html
  - Txt

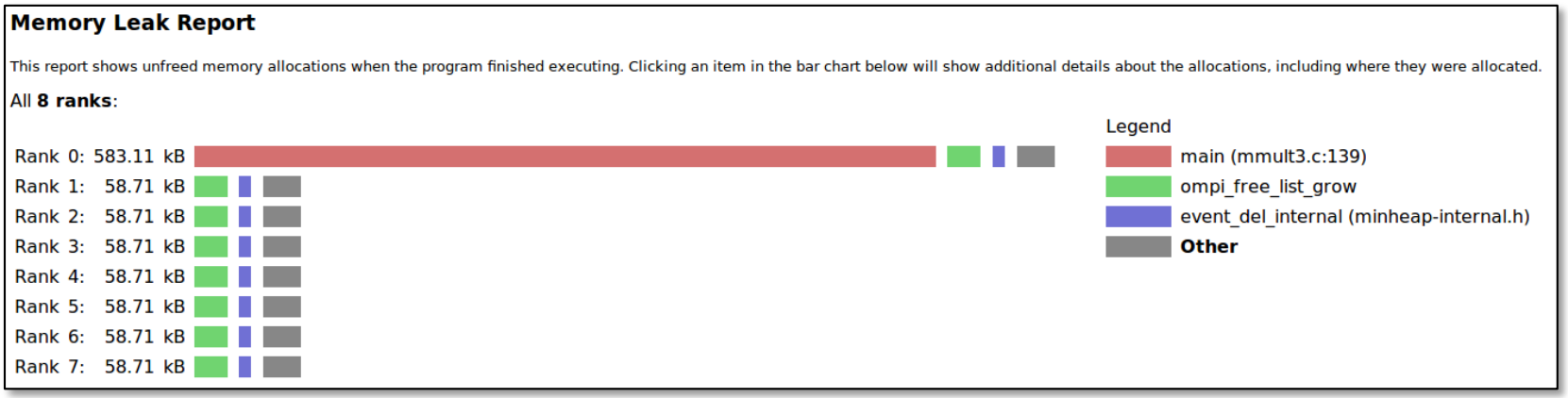
Automate debugging

#	Time	Tracepoint	Processes	Values
1	21:18.172	jacobi_mpi_omp_gnu.exe (_jacobi.f90:83)	0-127	residual: 2.57

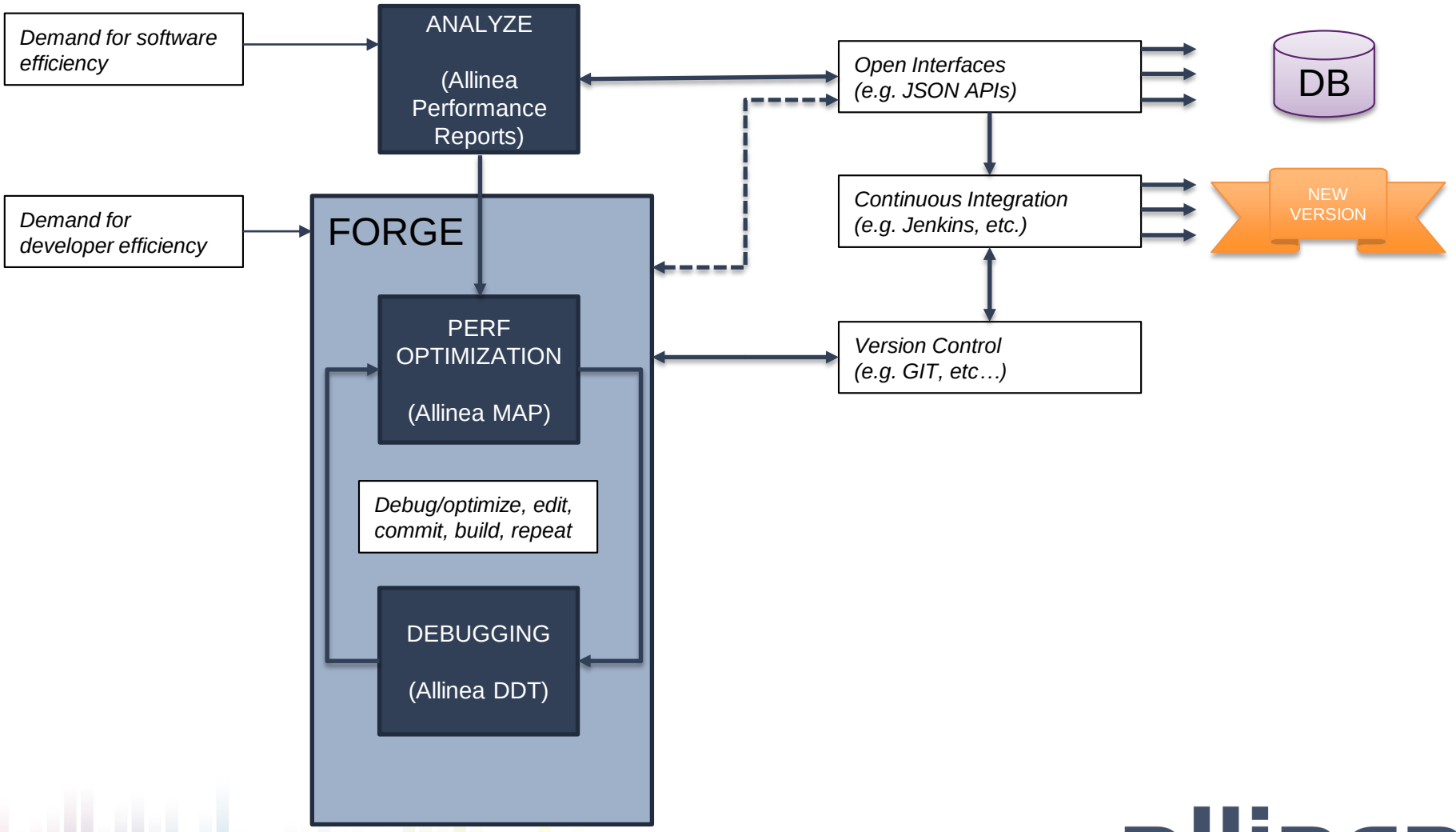
```
fail=0
# --- check DDT tracepoint (residual)
f=jacobi_omp_mpi_gnu_debug.txt
resid=`grep ^tracepoint $f |awk -Fresidual:  '{print $2}' |tail -1 |cut -c2-5`
if [ "$resid" != "2.57" ] ; then
         ((fail++))
        echo "Test has failed resid=$resid“
else
     echo “Test has succeeded”
```

Automate debugging

- Other available options:
  - --trace-changes: set a tracepoint on the variable introduced by the latest commit (git, svn, mercurial)
  - --break-at: set a breakpoint
  - --mem-debug: check for memory defects and leaks
  - --check-bounds: check for out of bounds array accesses

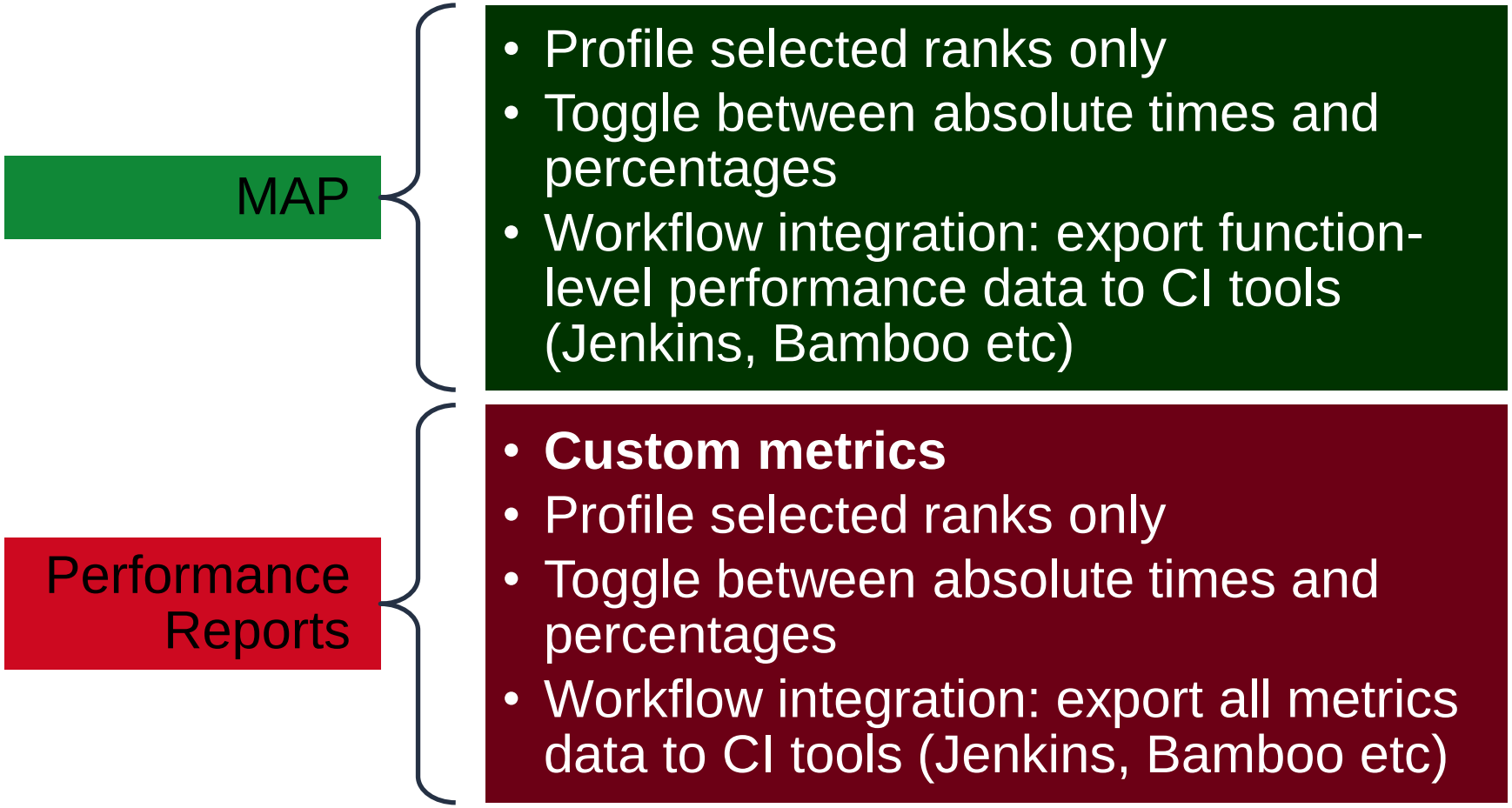


Development process workflow



allinea

What is coming next?



allinea

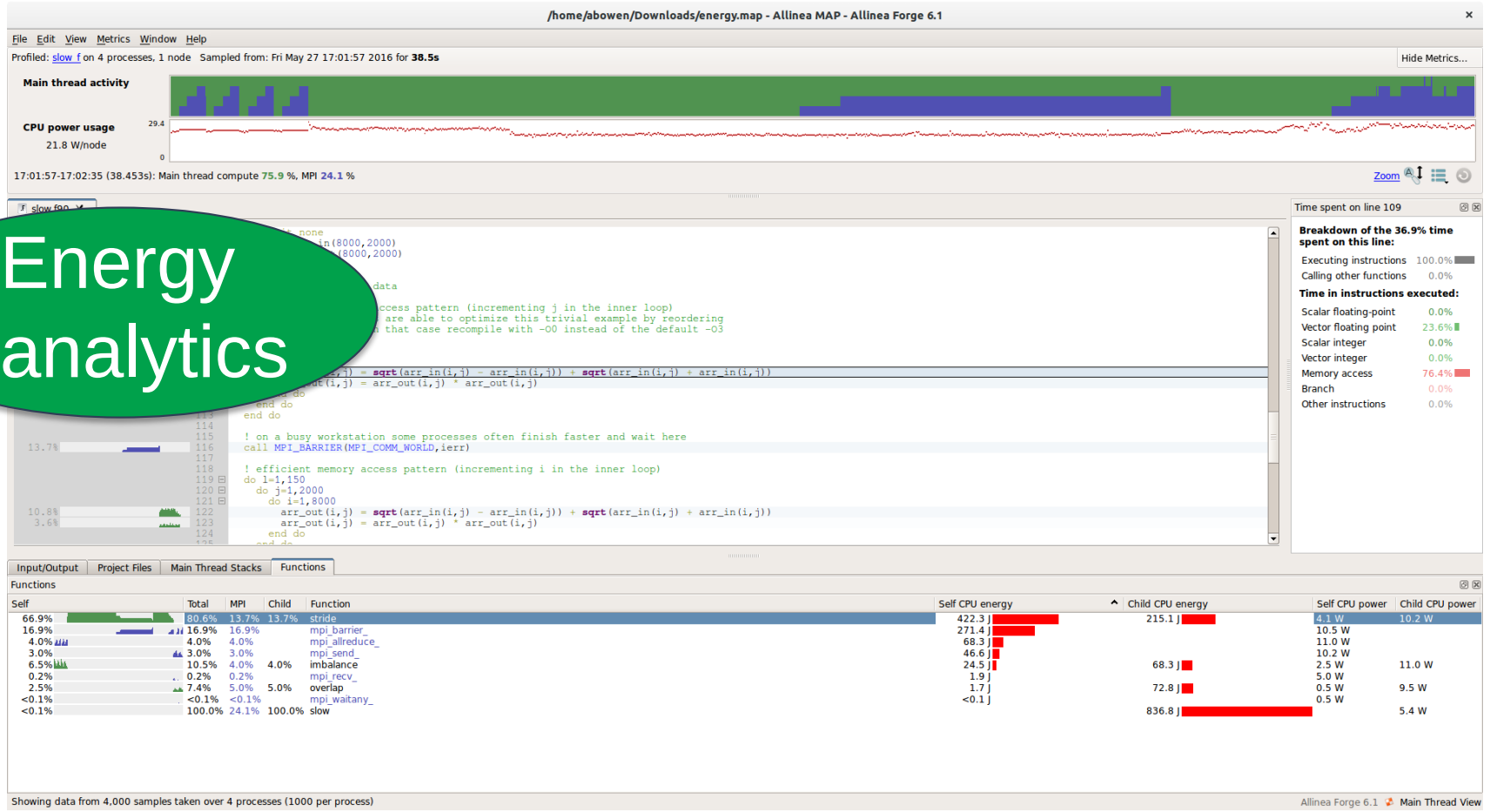
allinea

High performance tools to debug, profile, and analyze your applications

Thank you !

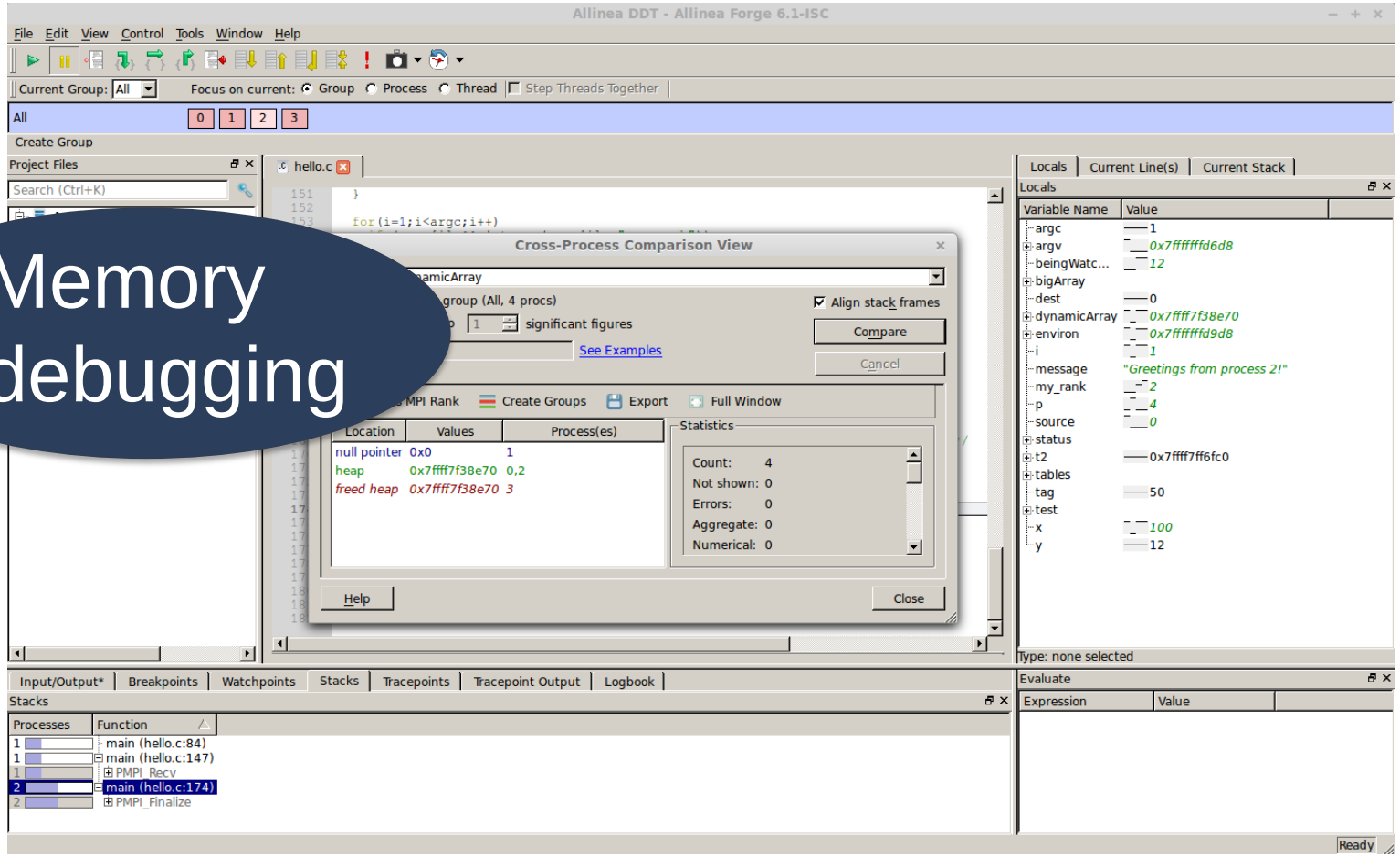
Technical Support team : support@allinea.com
Sales team : sales@allinea.com

Energy analytics



allinea

Memory debugging



allinea

6		0:05.230	n/a	Allinea Forge - Allinea DDT Debugging : mpirun -n 4 ./examples/wave_c 10 MPI implementation : Auto-Detect (Open MPI) * number of processes : 4 * number of nodes : 1 Memory debugging enabled : Yes * setting : Balanced * check bounds : 1 after
7		0:05.230	0-3	Play
8		0:05.295	0-3	Process stopped at breakpoint in do_math (wave.c:145).
				dditional Information

Offline debugging

nds	Function	Source
	main (wave.c:334)	▸ iterations = update(left, right);
0-3 4	update (wave.c:211)	▾ do_math(j); 208. if ((first + j - 1 == 1) || (first + j - 1 == tpoints)) 209. newval[j] = 0.0; 210. else 211. do_math(j); 212. } 213. for (j = 1; j <= npoints; j++) 214. {
0-3 4	do_math (wave.c:145)	▸ newval[i] = (2.0 * values[i]) - oldval[i]
0-3 4	orte_progress_thread_engine	
0-3 4	opal_libevent2021_event_base_loop	
0-3 4	poll_dispatch	
0-3 4	poll (syscall-template.S:81)	

▶ Current Stack

Variables

▾ Rank 0, thread 1

Name	Value
allt	<optimized out>
argc	2
argv	0x7fffffffb858
calculation_rate	<optimized out>
iterations	<optimized out>
left	*-2 (from -2 to 2)*
right	*1 (from -2 to 3)*

▾ Rank 0, thread 1

Name	Value
iterations	<optimized out>
j	*102 (from 101 to 102)*
left	*-2 (from -2 to 2)*
now	<aggregate value>
right	*1 (from -2 to 3)*
stop	0

▾ Rank 0, thread 1

Name	Value
i	*102 (from 101 to 102)*

10		0:07.795	0-3	Play
11		0:07.815	0-3	Process stopped at breakpoint in do_math (wave.c:145).

allinea