

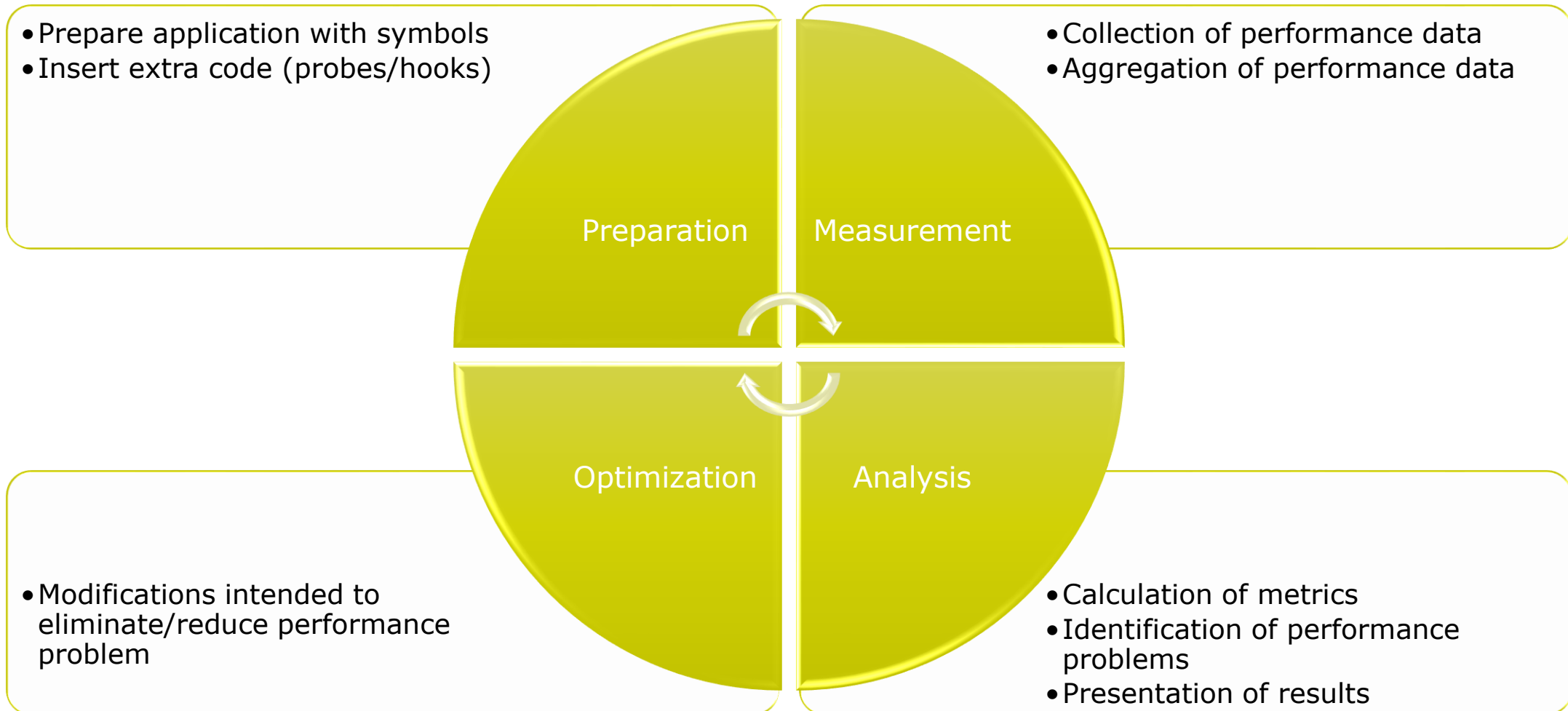
# Score-P – A Joint Performance Measurement Run-Time Infrastructure for Scalasca, TAU, and Vampir

---

VI-HPS Team



# Performance engineering workflow





## Score-P

---

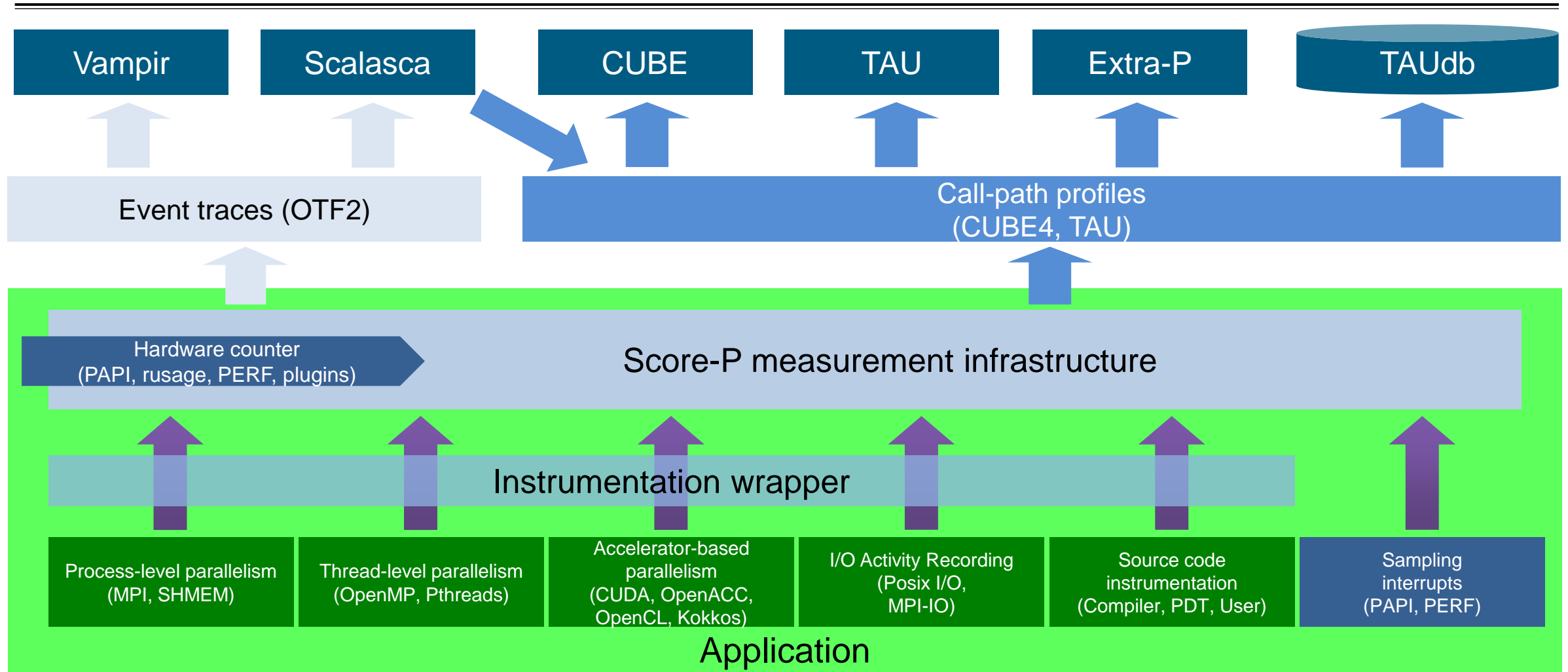
- Infrastructure for instrumentation and performance measurements
- Instrumented application can be used to produce several results:
  - Call-path profiling: CUBE4 data format used for data exchange
  - Event-based tracing: OTF2 data format used for data exchange
- Supported parallel paradigms:
  - Multi-process: MPI, SHMEM
  - Thread-parallel: OpenMP, Pthreads
  - Accelerator-based: CUDA, OpenCL, OpenACC, Kokkos
- Open Source; portable and scalable to all major HPC systems
- Initial project funded by BMBF
- Further developed in multiple 3<sup>rd</sup>-party funded projects

GEFÖRDERT VOM



Bundesministerium  
für Bildung  
und Forschung

## Score-P overview



# Partners

---

- Forschungszentrum Jülich, Germany
- Gesellschaft für numerische Simulation mbH Braunschweig, Germany
- RWTH Aachen, Germany
- Technische Universität Darmstadt, Germany
- Technische Universität Dresden, Germany
- Technische Universität München, Germany
- University of Oregon, Eugene, USA



## Hands-on: NPB-MZ-MPI / BT

---



# Performance analysis steps

---

- 0.0 Reference preparation for validation
  
- 1.0 Program instrumentation
  - 1.1 Summary measurement collection
  - 1.2 Summary analysis report examination
  
- 2.0 Summary experiment scoring
  - 2.1 Summary measurement collection with filtering
  - 2.2 Filtered summary analysis report examination
  
- 3.0 Event trace collection
  - 3.1 Event trace examination & analysis

## Local installation (IvyMUC)

---

- Latest/recent versions/combinations of VI-HPS tools not yet installed system-wide
  - Source module use `/lrz/sys/courses/vihps/modulefiles` to prepare the environment
  - Required for each shell session
  - Score-P installation is toolchain specific

```
% module load intel/19.0 mpi.intel/2019
% module use /lrz/sys/courses/vihps/modulefiles
% module load cubegui/4.6 scorep/7.0-intel
```

- Check `module avail scorep` for alternate Score-P modules available
- Copy tutorial sources to your `$HOME` directory (should be done already)

```
% cd $HOME
% tar zxvf /lrz/sys/courses/vihps/material/NPB3.3-MZ-MPI.tar.gz
% cd NPB3.3-MZ-MPI
```



## NPB-MZ-MPI / BT instrumentation

```
#-----  
# The Fortran compiler used for MPI programs  
#-----  
#MPIF77 = mpif77  
  
# Alternative variants to perform instrumentation  
...  
MPIF77 = scorep --user mpif77  
  
# This links MPI Fortran programs; usually the same as ${MPIF77}  
FLINK    = $(MPIF77)  
...
```

- Edit config/make.def to adjust build configuration
  - Modify specification of compiler/linker: MPIF77

Uncomment the Score-P  
compiler wrapper  
specification

## NPB-MZ-MPI / BT instrumented build

```
% make clean

% make bt-mz CLASS=B NPROCS=28
cd BT-MZ; make CLASS=B NPROCS=28 VERSION=
make: Entering directory 'BT-MZ'
cd ../sys; cc -o setparams setparams.c -lm
../sys/setparams bt-mz 8 B
scorep --user mpif77 -g -c -O3 -qopenmp bt.f
[...]
cd ../common; scorep --user mpif77 -g -c -O3 -qopenmp timers.f
[...]
scorep --user mpif77 -g -O3 -qopenmp -o ../bin.scorep/bt-mz_B.28 \
bt.o initialize.o exact_solution.o exact_rhs.o set_constants.o \
adi.o rhs.o zone_setup.o x_solve.o y_solve.o exch_qbc.o \
solve_subs.o z_solve.o add.o error.o verify.o mpi_setup.o \
../common/print_results.o ../common/timers.o
Built executable ../bin.scorep/bt-mz_B.28
make: Leaving directory 'BT-MZ'
```

- Return to root directory and clean-up
- Re-build executable using Score-P compiler wrapper

# Measurement configuration: scorep-info

---

```
% scorep-info config-vars --full
SCOREP_ENABLE_PROFILING
  Description: Enable profiling
  [...]
SCOREP_ENABLE_TRACING
  Description: Enable tracing
  [...]
SCOREP_TOTAL_MEMORY
  Description: Total memory in bytes for the measurement system
  [...]
SCOREP_EXPERIMENT_DIRECTORY
  Description: Name of the experiment directory
  [...]
SCOREP_FILTERING_FILE
  Description: A file name which contain the filter rules
  [...]
SCOREP_METRIC_PAPI
  Description: PAPI metric names to measure
  [...]
SCOREP_METRIC_RUSAGE
  Description: Resource usage metric names to measure
  [...] More configuration variables ...
```

- Score-P measurements are configured via environmental variables

## Summary measurement collection

```
% cd bin.scorep
% cp ../jobscript/ivymuc/scorep.sbatch .
% cat scorep.sbatch
...
# Score-P measurement configuration
export SCOREP_EXPERIMENT_DIRECTORY=scorep_bt-mz_sum
#export SCOREP_FILTERING_FILE=../config/scorep.filt
#export SCOREP_METRIC_PAPI=PAPI_TOT_INS,PAPI_TOT_CYC,...
#export SCOREP_METRIC_PAPI_PER_PROCESS=PAPI_L2_TCM
#export SCOREP_METRIC_RUSAGE=ru_stime
#export SCOREP_METRIC_RUSAGE_PER_PROCESS=ru_maxrss
#export SCOREP_TIMER=gettimeofday

# Run the application
mpiexec -n $SLURM_NTASKS ./bt-mz_${CLASS}.${PROCS}

% sbatch --reservation=hhps1s21_workshop scorep.sbatch
```

- Change to the directory containing the new executable before running it with the desired configuration
- Check settings

Leave these lines commented out for the moment

- Submit job

# Summary measurement collection

---

```
% less npb_btmz.o<job_id>

NAS Parallel Benchmarks (NPB3.3-MZ-MPI) - BT-MZ MPI+OpenMP \
>Benchmark

Number of zones:  8 x  8
Iterations: 200    dt:  0.000300
Number of active processes:      28
Use the default load factors with threads
Total number of threads:    112  (  4.0 threads/process)

Calculated speedup = 71.69

Time step    1

[... More application output ...]
```

- Check the output of the application run

# BT-MZ summary analysis report examination

```
% ls
bt-mz_B.28 bt-mz.<job_id>.out scorep_bt-mz_sum/
% ls scorep_bt-mz_sum
MANIFEST.md profile.cubex scorep.cfg
```

```
% cube scorep_bt-mz_sum/profile.cubex
```

```
[CUBE GUI showing summary analysis report]
```

- Creates experiment directory including
  - A brief content overview (MANIFEST.md)
  - A record of the measurement configuration (scorep.cfg)
  - The analysis report that was collated after measurement (profile.cubex)
- Interactive exploration with Cube

## Hint:

Copy 'profile.cubex' to local system (laptop) using 'scp' to improve responsiveness of GUI

## Reference results available:

[/lrz/sys/courses/vihps/public/reference results](/lrz/sys/courses/vihps/public/reference/results)

## Further information

---

- Community instrumentation & measurement infrastructure
  - Instrumentation (various methods)
  - Basic and advanced profile generation
  - Event trace recording
- Available under 3-clause BSD open-source license
- Documentation & Sources:
  - <http://www.score-p.org>
- User guide also part of installation:
  - `<prefix>/share/doc/scorep/{pdf,html}/`
- Support and feedback: [support@score-p.org](mailto:support@score-p.org)
- Subscribe to [news@score-p.org](mailto:news@score-p.org), to be up to date