

# Case Study: FLOWer

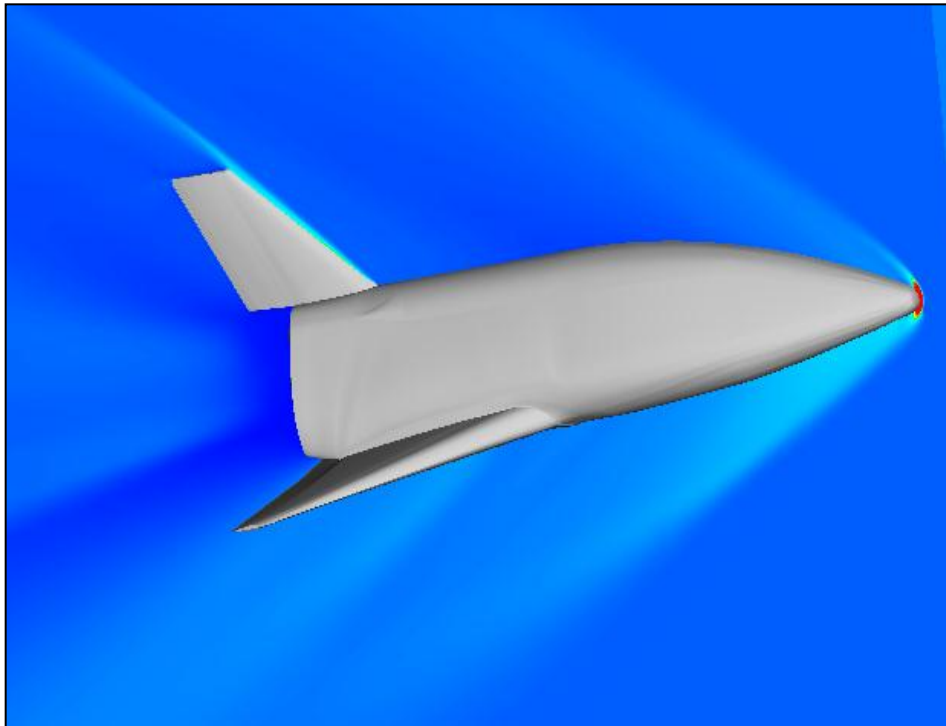
## – Combining MPI and Autoparallelization

*Dieter an Mey*

*Center for Computing and Communication  
Aachen University*

*anmey@rz.rwth-aachen.de*

## Case Study: FLOWer



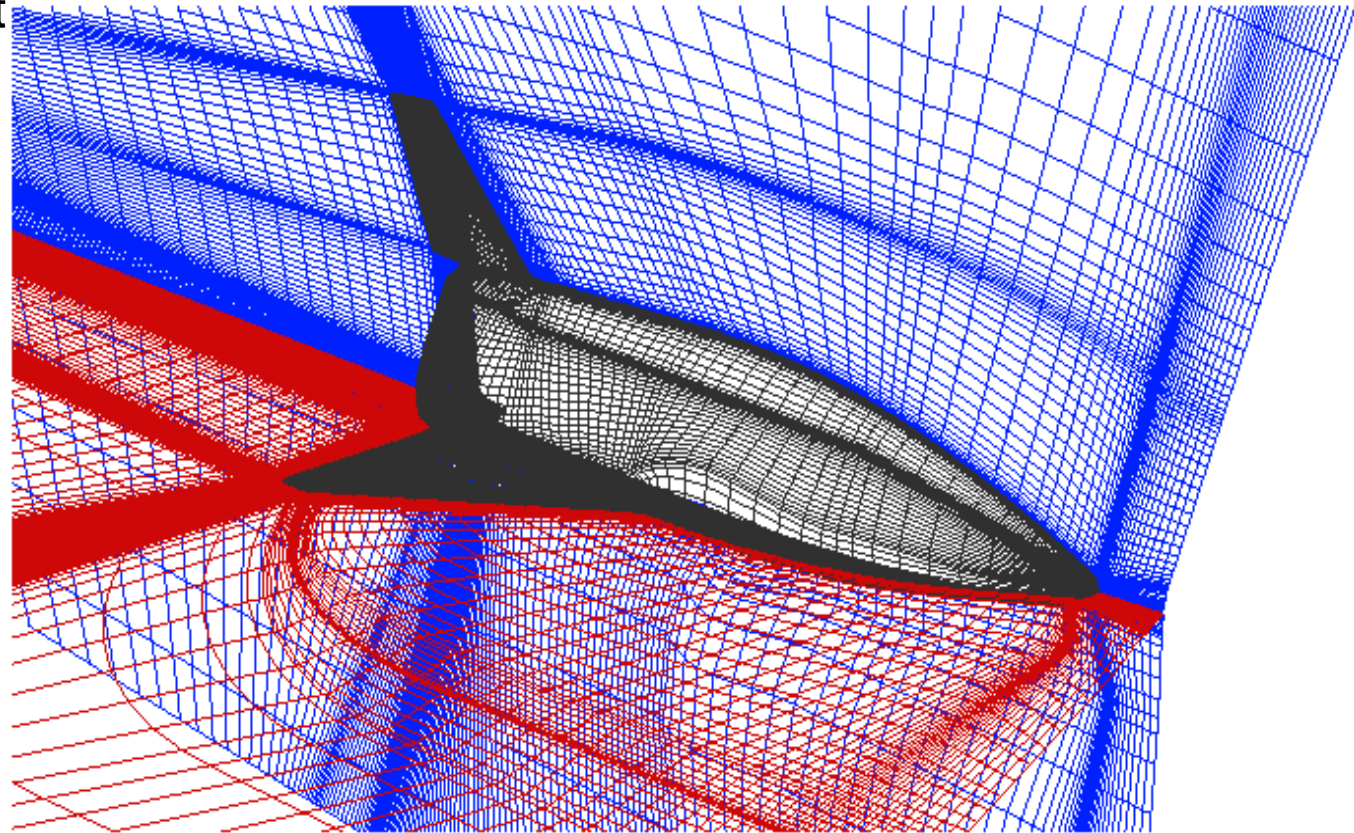
# Case Study: FLOWer

---

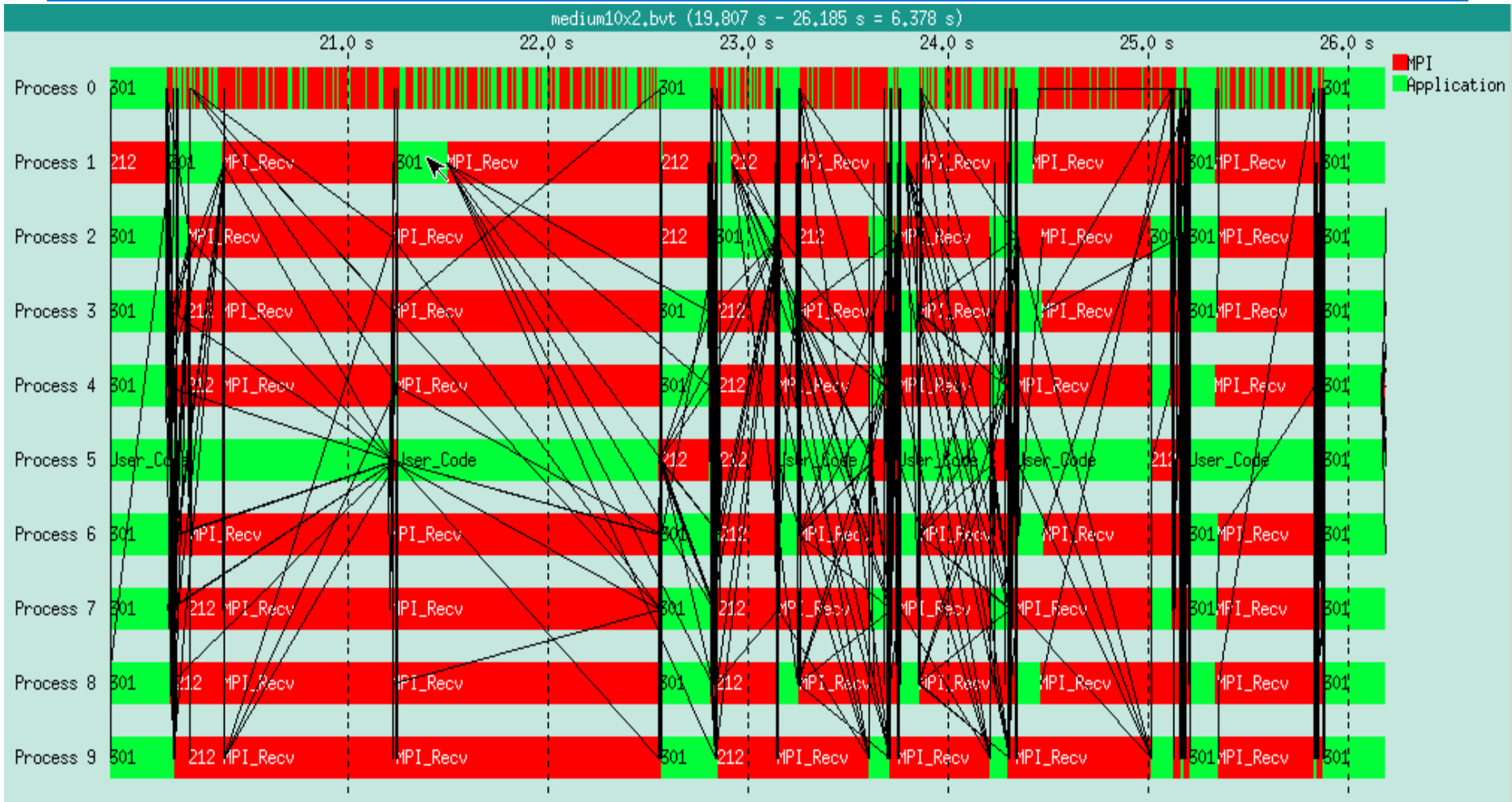
- Sponsored by the German Research Council (DFG), Birgit Reinartz and Michael Hesse in the Lehr- und Forschungsgebiet für Mechanik, RWTH Aachen University simulated PHOENIX, a small scale prototype of the **Space Hopper**, of a **space launch vehicle** designed to take off horizontally and glide back to earth after placing its cargo in orbit.
- Solution of the Navier-Stokes-Equations with the **FLOWer** flow solver, developed at the German Aerospace Center (DLR)
- Parallelized with the **CLIC-3D** communication library which handles all **MPI** communication.
- Block-oriented information exchange => the number of blocks is limits the number of MPI tasks.
- Underneath to the coarse-grained parallelization with CLIC-3D/MPI, on a lower level, most of the compute intense loops can be **parallelized automatically** by the Fortran compiler => **Hybrid Parallelization**

# Small Testcase

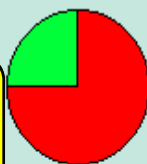
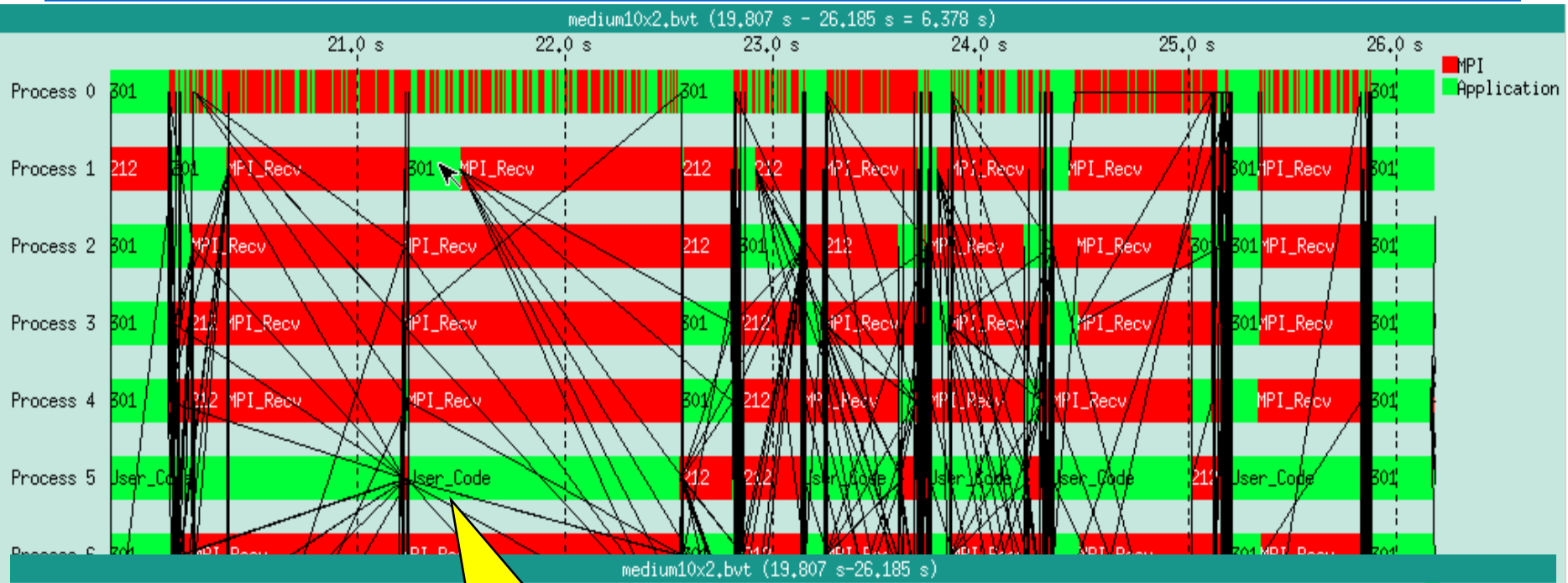
- Euler - Solver
- 1 Mio grid point
- inviscid flow
- 20 blocks



# First Performance Experiments: 10x2 Processors



# First Performance Experiments: 10x2 Processors



Application  
MPI

# Hybrid Parallelization on a Shared Memory System

run10x1+5.ksh:

```
#!/bin/ksh
export OMP_NUM_THREADS=1;
if [[ ${MP_RANK} = 0 ]]
then
    export OMP_NUM_THREADS=
fi;
if [[ ${MP_RANK} = 5 ]]
then
    export OMP_NUM_THREADS=5;
fi;
flower.exe
```

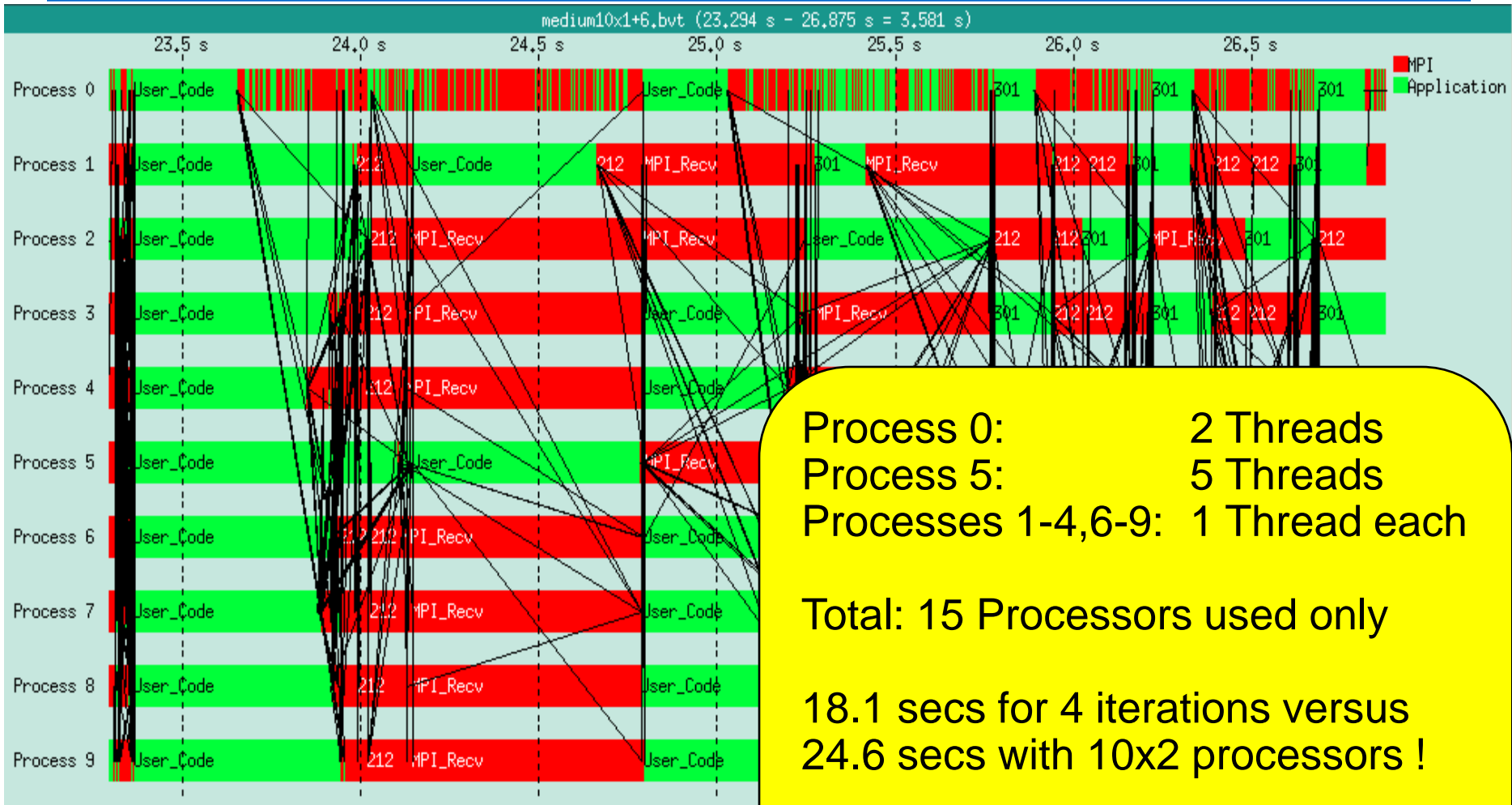
Process 0: 2 Threads  
 Process 5: 5 Threads  
 Processes 1-4,6-9: 1 Thread each

Total: 15 Processors used only

Depends on MPI implementation

mprun -np 10 run10x1+5.ksh

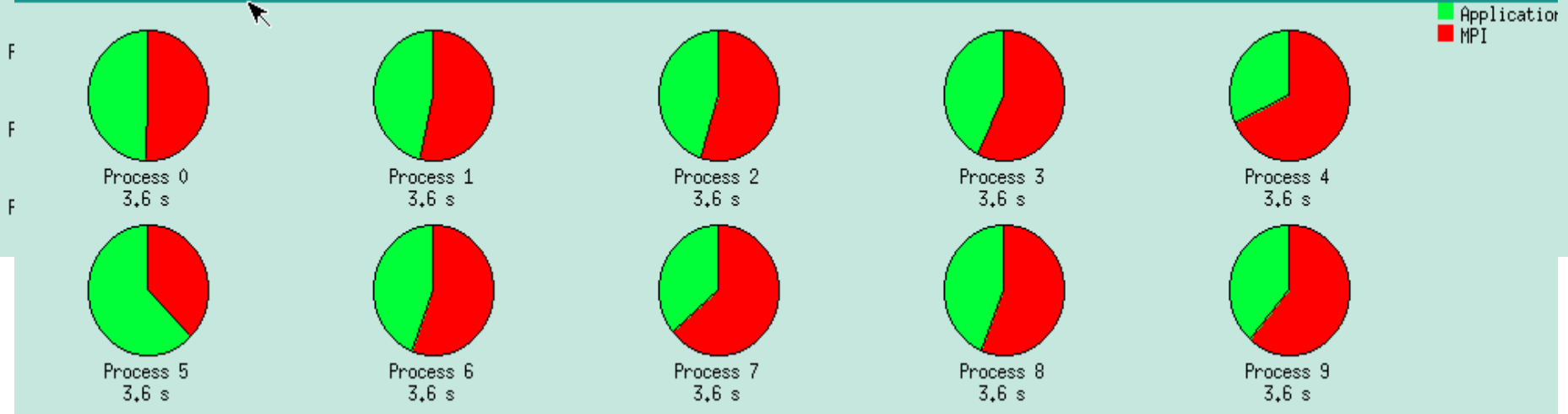
# First Performance Experiments: 10x1+5 Processors



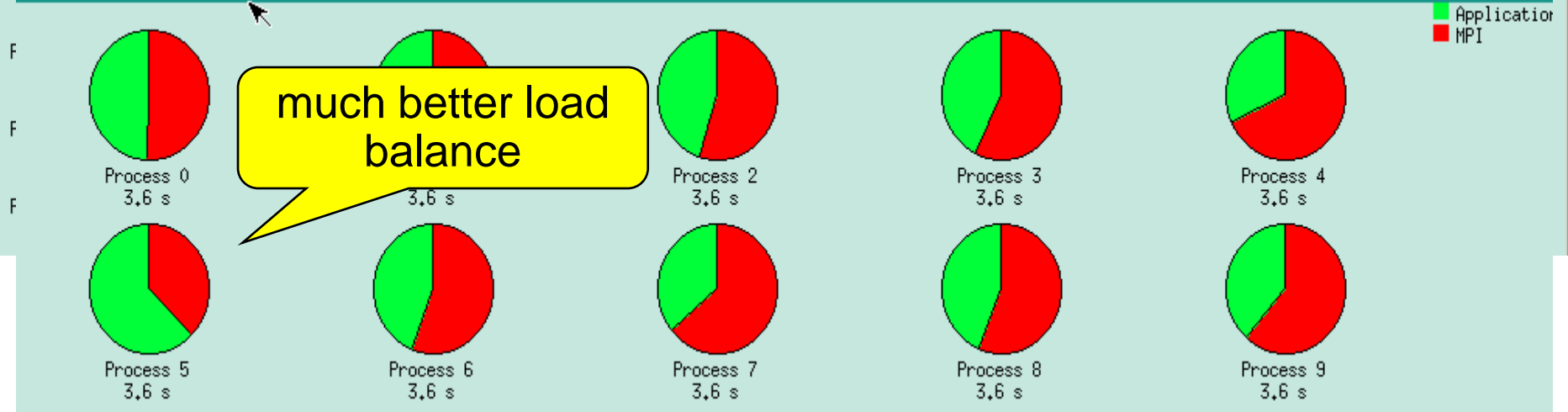
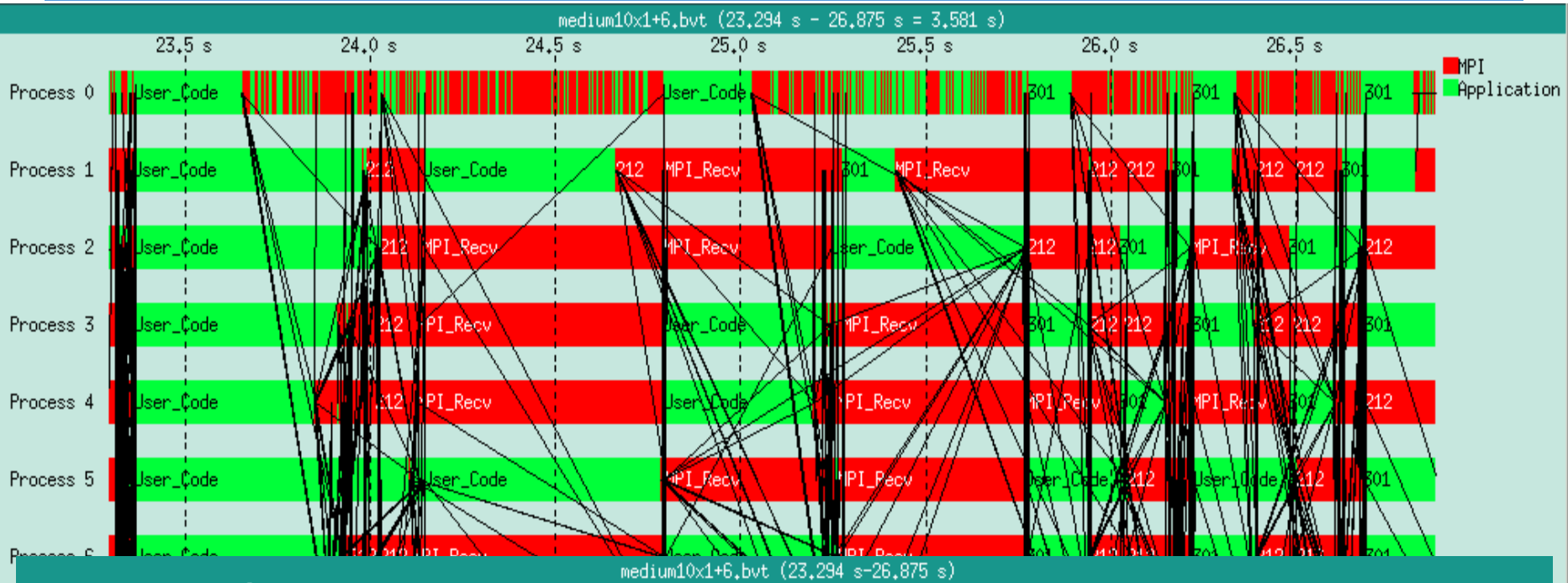




# First Performance Experiments: 10x1+5 Processors

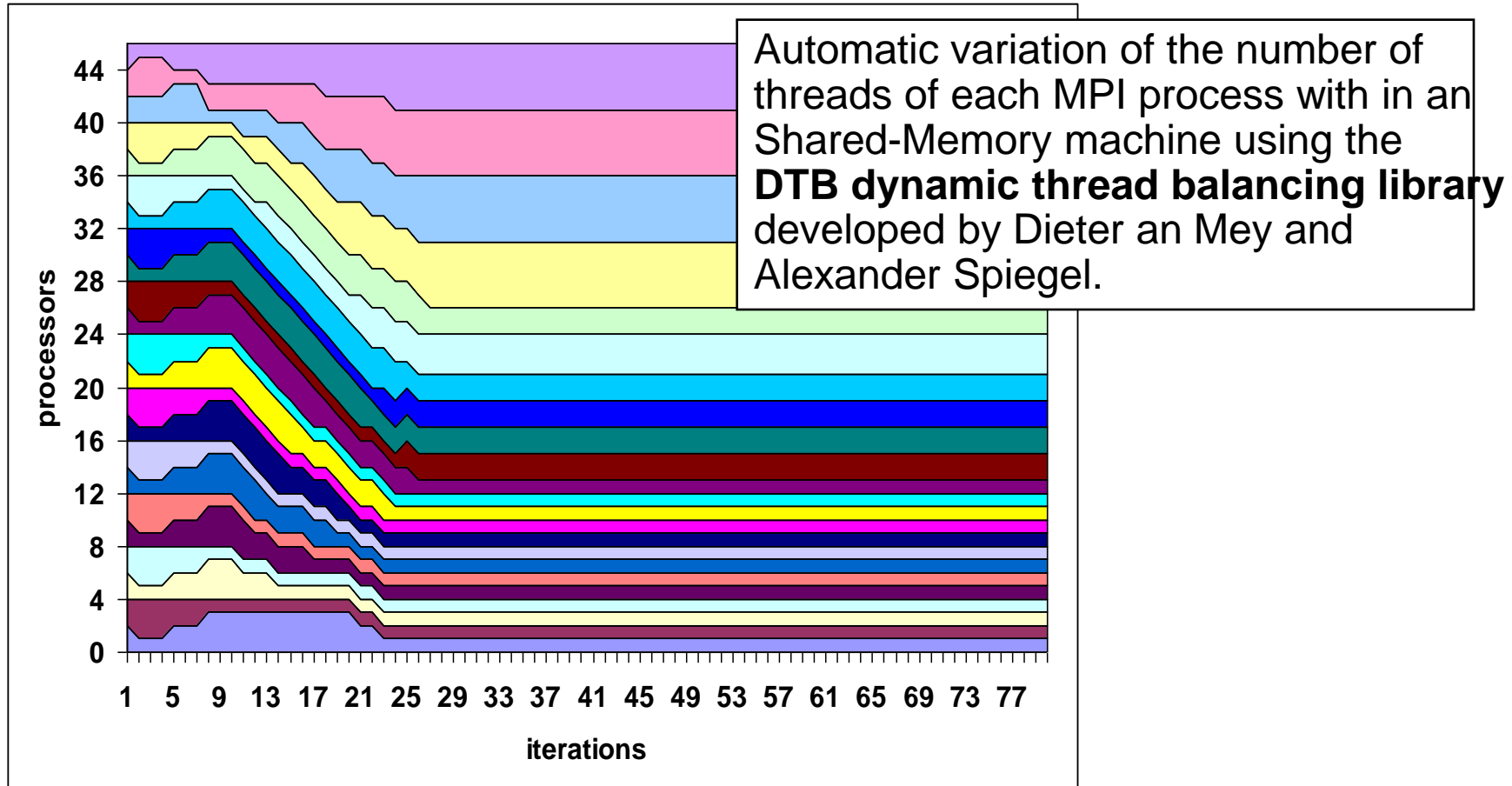


# First Performance Experiments: 10x1+5 Processors



# FLOWer on a Sun Fire 15K

## 23 MPI-Tasks x 2 OpenMP Threads



- **Tracing libraries** typically use the **MPI profiling interface**.
- It offers an easy-to-use collection of runtime information for MPI application after **re-linking**.
- The amount of **trace information** can be quite **large**,  
Tools for efficiently **controlling** and **filtering** trace data are essential.
- Trace data can be with the comfortable GUI s
- The performance of the **Flower CFD package** on a large **shared memory system** could be easily improved without any code changes:  
By **adjusting the number of threads per MPI task** manually, a much better **load balance** could be achieved.
- The **DTB** dynamic thread balancing library was developed to do this automatically