Deutscher Wetterdienst

# Preparing NWP-Models for Tera-Computing

Ulrich Schättler, Elisabeth Krenzien,

Michael Czajkowski

Deutscher Wetterdienst

# Contents

- T3E $\rightarrow$ IBM: Late Lament
- Upgrade of NWP-System at DWD
- Optimizations for Boundary Exchange and I/O
- Environment for the NWP-System: Now / Future
- LM_RAPS_3.0
- Conclusions

# T3E → IBM: Late Lament

- If profiling various applications on a big IBM-System, which MPI-Routine do you expect to take the most time?
- It is: MPI_BARRIER!
- Because most codes have been developed on a T3E:
  - Barriers did take almost no time
  - MPI_SEND, MPI_RECV was not really blocking and a barrier could be helpful to ensure correct program execution!
- Is IBM-communication really so bad?

# Timings for LM on T3E / IBM

| Timings in Seconds | **T3E** | **IBM (pwr3)** |
|---|---|---|
| Dyn. Computations | 1146.58 | 1105.09 |
| Communications | 324.90 | 568.19 |
| Barrier waiting | 187.09 | 373.63 |
| Phys. Computations | 708.80 | 712.22 |
| Communications | 30.84 | 60.05 |
| Barrier waiting | 200.41 | 241.18 |
| I/O | 576.05 | 358.05 |
| # Processors | 484 | 160 |

# Timings for LM on T3E / IBM

- Today´s operational domain size: $325 \times 325 \times 35$
- 48 hour forecast (should be finished within 1 hour)
- Timings: „Not so good, but acceptable“
- But what happens, if O(1000) processors are used?

# Upgrade of NWP-System

- New model components
  - Cloud ice scheme (GME / LM)
  - Multi-layer soil model (GME / LM in QI 2005)
  - Sea-ice model (GME)
  - Prognostic Precipitation (LM)
  - 2 time level Runge-Kutta numerical core
    - 3rd order in time; 5th order (horizontal) in space
    - at the moment tested for very high resolution runs

# Upgrade of NWP-System

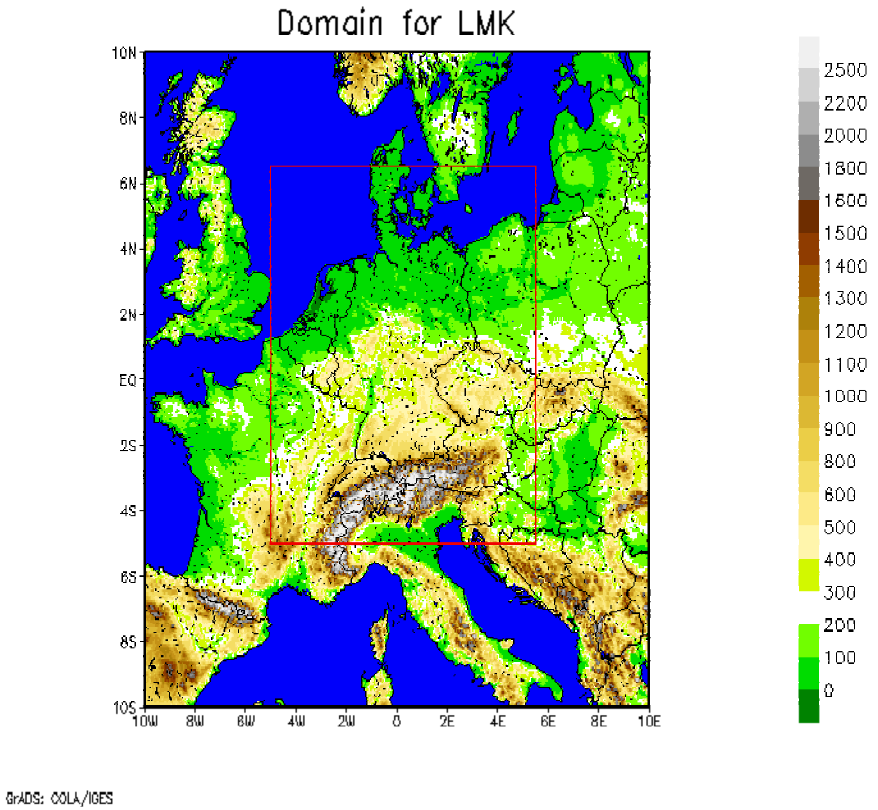
- In Preparation
  - 3D Var Physical Space Assimilation System (GME)
  - Assimilation of radar data (latent heat nudging;  LM)
  - 3D Turbulence scheme (LM)
  - Graupel scheme (LM)
  - Parameterization of shallow convection (LM)
  - Lake-Model (LM)

# Upgrade of NWP-System

- Local Model (LM):
  - The LM is used and further developed within the Consortium for Small Scale Modeling (COSMO)
  - Aim is to run LM with a very high resolution (≤ 3 km): LMK ( LM Kürzestfrist)
  - But coming up next: Running the LM with 7 km resolution over the whole of Europe (LME)
- Global Model (GME):
  - GME is now run with a resolution of about 40 km and 40 vertical layers

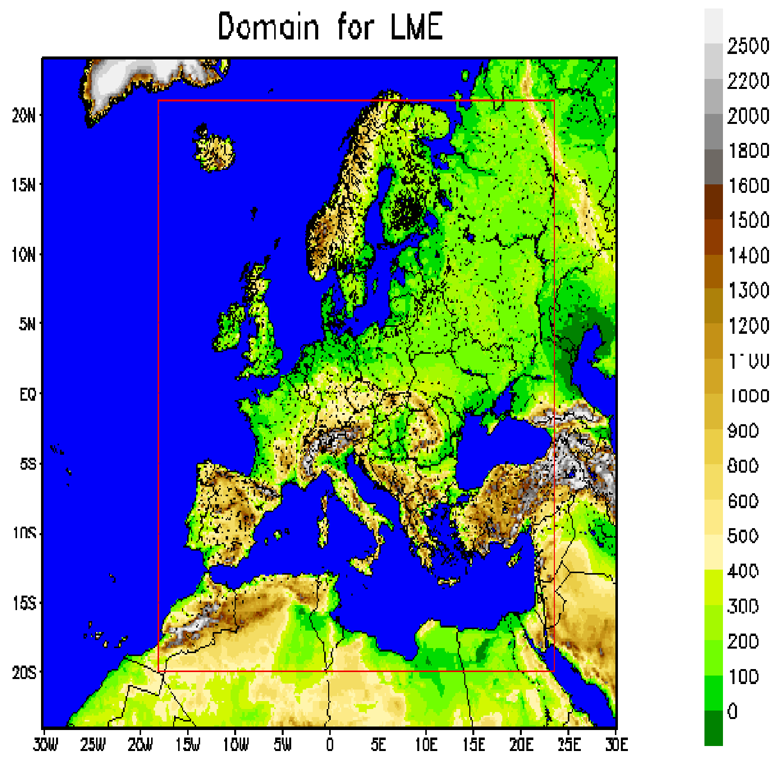
# LMK

- 2.8 km grid spacing
- 421×461 grid points
- 50 vertical layers
- 30 s time step
- 2 time level Runge-Kutta
- continuous upgrade by new components
- operational in 2006



# LME

- Still 7 km grid spacing
- 665×657 grid points
- 40 vertical layers
- 40 s time step
- 3 time level Leapfrog
- (or Runge-Kutta: 72 s)
- operational in 2005 (QII)




GrADS: COLA/IGES

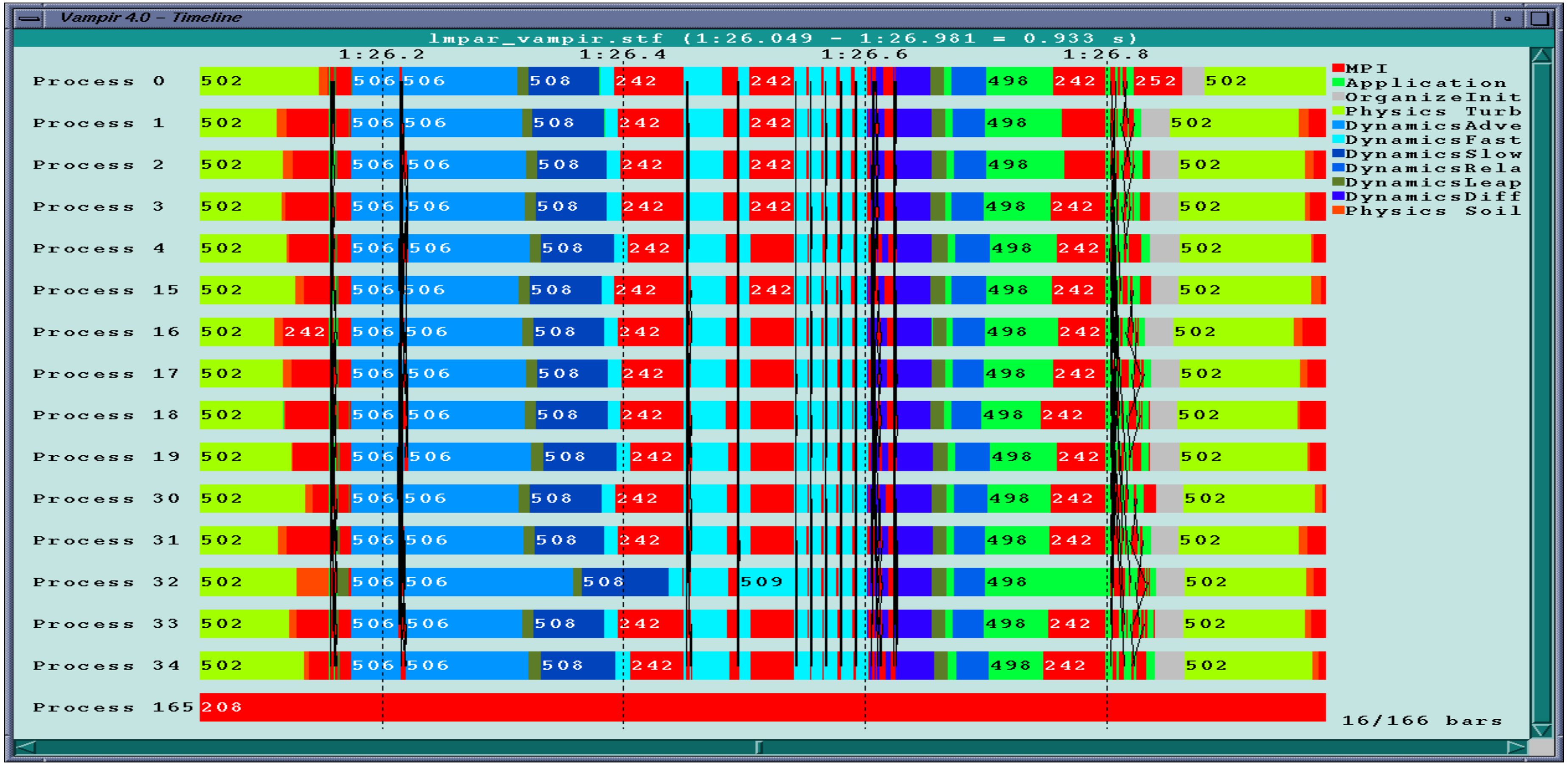
# LME: First Timings

| | |
|---|---|
| Dyn. Computations | 367.51 |
| Communications | 85.30 |
| Barrier waiting | 130.37 |
| Phys. Computations | 188.16 |
| Communications | 7.31 |
| Barrier waiting | 31.76 |
| Input | 126.62 |
| Output | 258.70 |
| Total Time for the Job | 1285.46 |

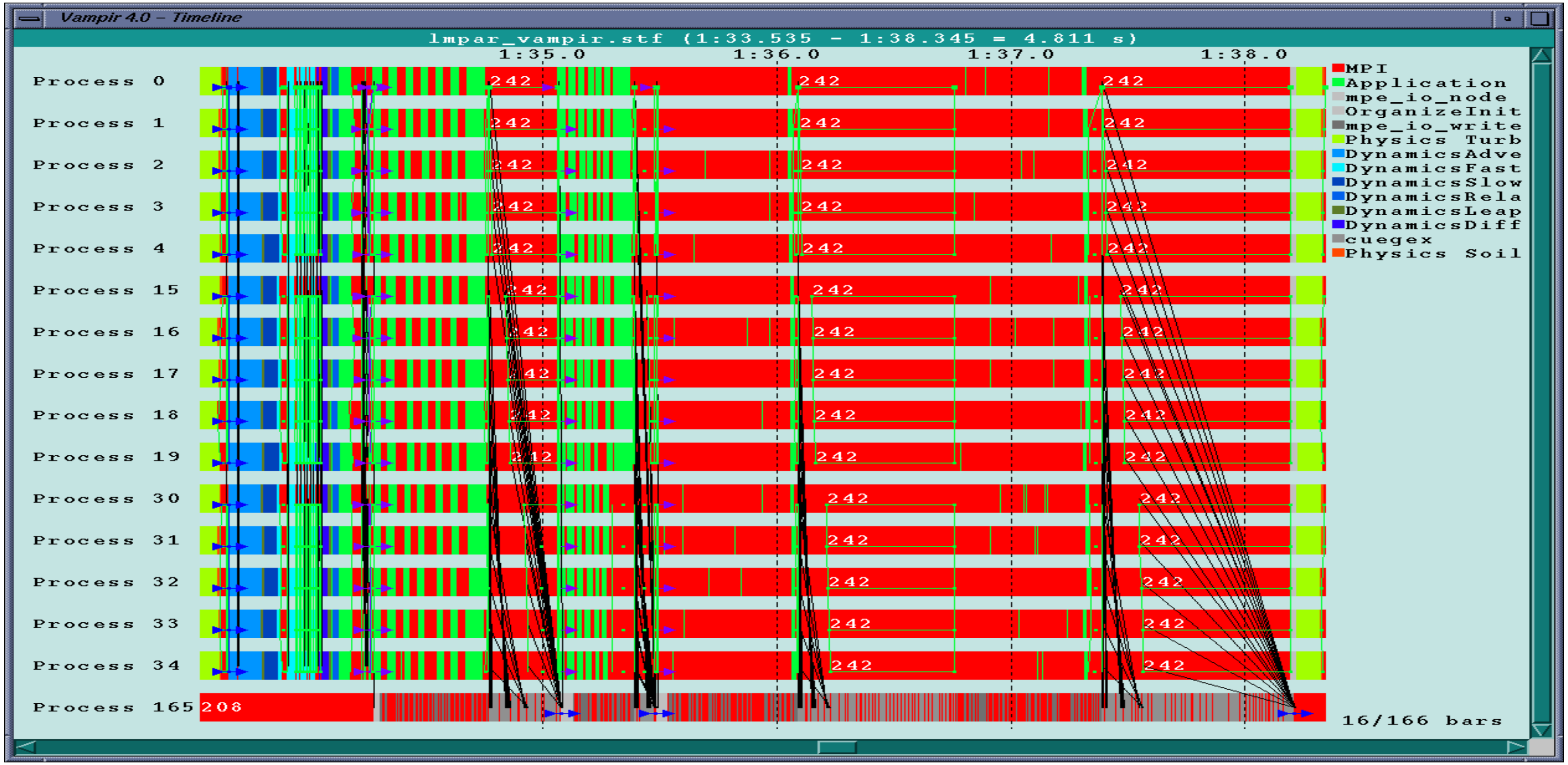
# LME: First Timings

- 12 hour forecast on 900+1 procs (should be $\leq$ 900s)
- Boundary Exchange with MPI_ISEND, MPI_WAIT on the sender and MPI_RECV on receiver side
- Explicit buffering of data
- Extra processor for I/O (asynchronous IO); but realization with blocking communication
- Still using all that barriers!

# LM Time Step with Trace Analyzer

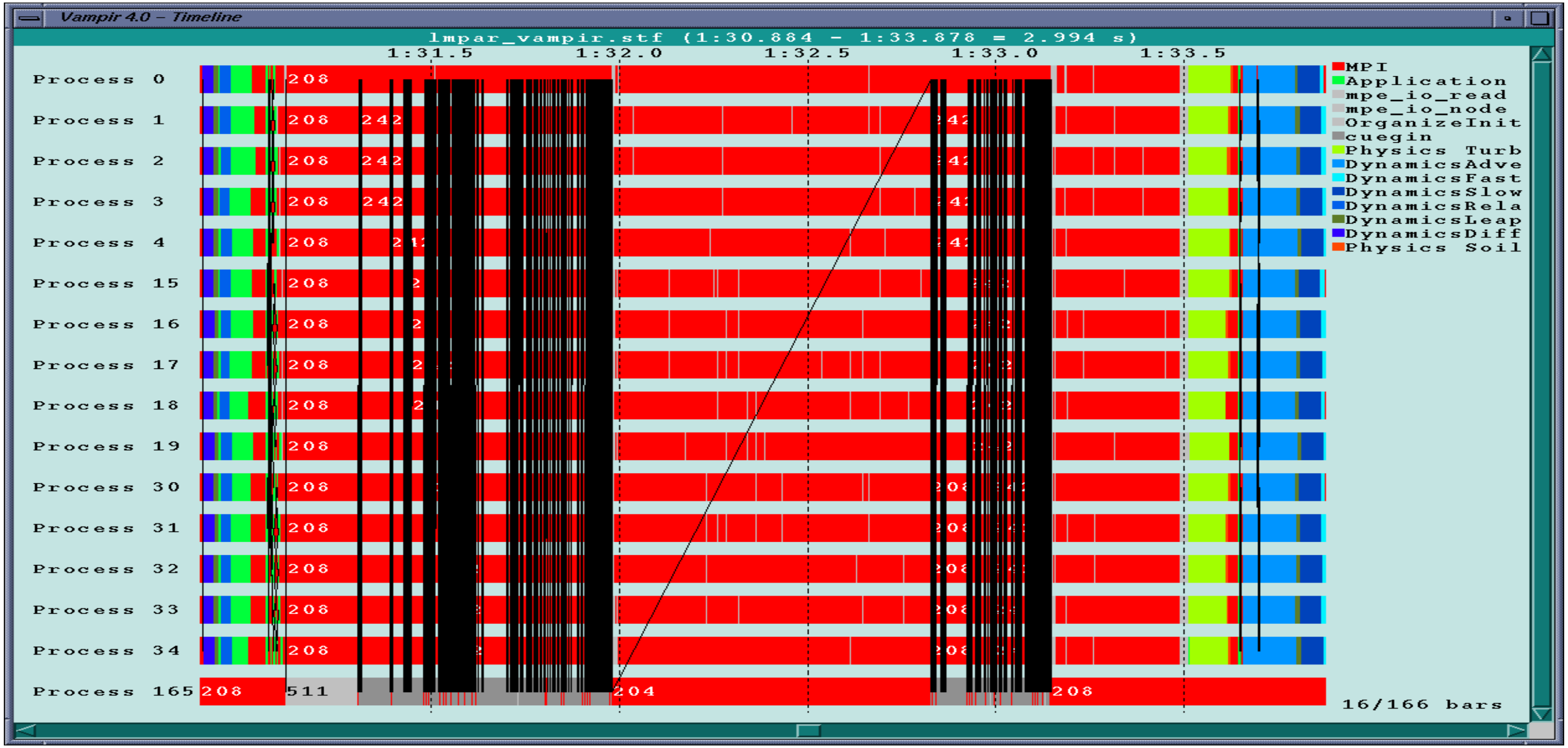


# Output Step


Vampir 4.0 – Timeline
lmpar_vampir.stf (1:33.535 - 1:38.345 = 4.811 s)
1:35.0
1:36.0
1:37.0
1:38.0
Process 0
Process 1
Process 2
Process 3
Process 4
Process 15
Process 16
Process 17
Process 18
Process 19
Process 30
Process 31
Process 32
Process 33
Process 34
Process 165
208
242
MPI
Application
mpe_io_node
OrganizeInit
mpe_io_write
Physics Turb
DynamicsAdve
DynamicsFast
DynamicsSlow
DynamicsRela
DynamicsLeap
DynamicsDiff
cuegex
Physics Soil
16/166 bars

Deutscher Wetterdienst

# Input Step



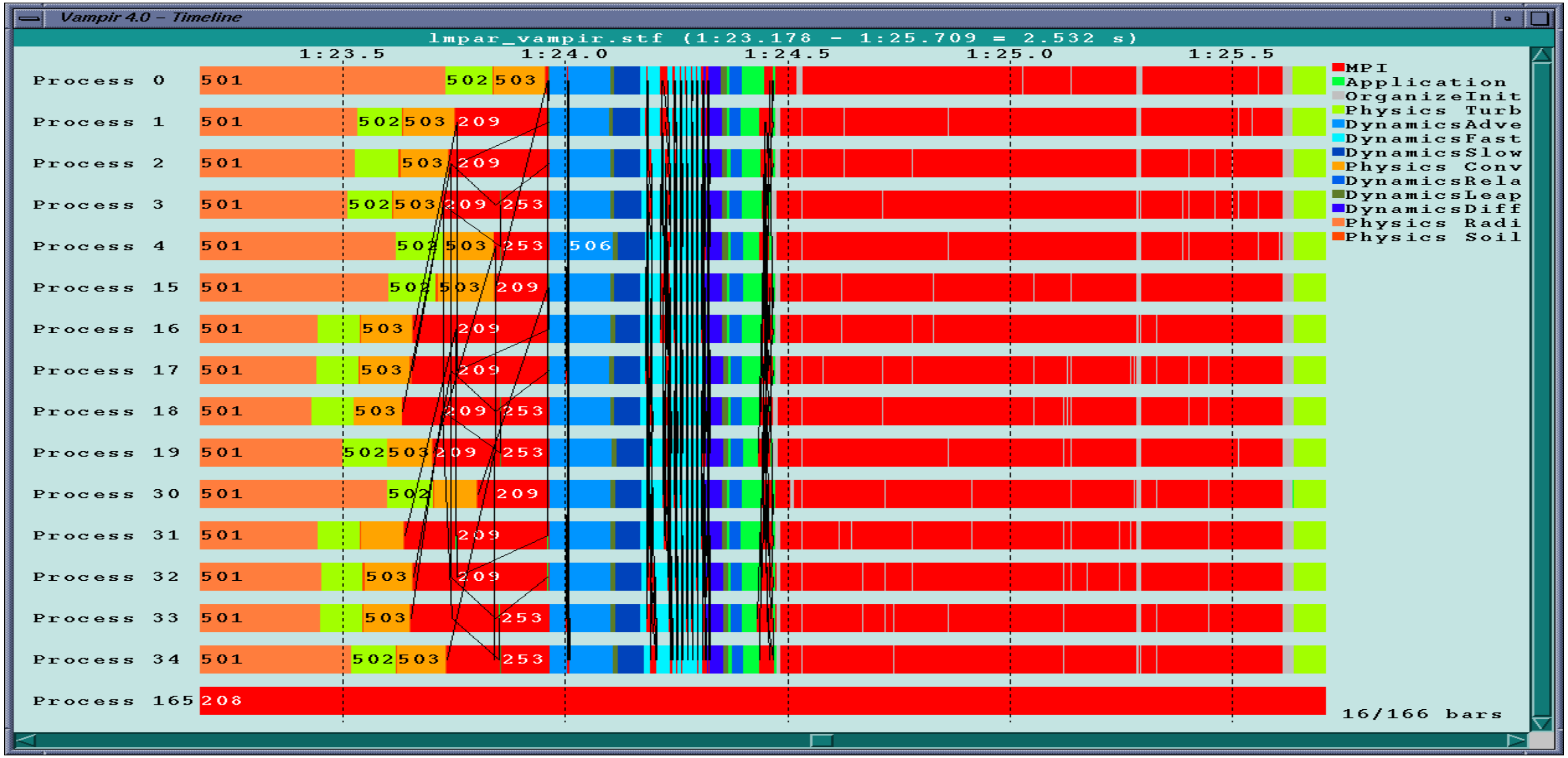
# Optimizations

- Boundary Exchange with
  - MPI_ISEND, MPI_WAIT and MPI_RECV
  - MPI_IRECV, MPI_WAIT and MPI_SEND
  - MPI_SENDRECV
- Implicit buffering of data by using MPI_DATATYPES
- Non-blocking communication for extra I/O processor
- Try to do a look-ahead reading
- There is really no need for barriers!

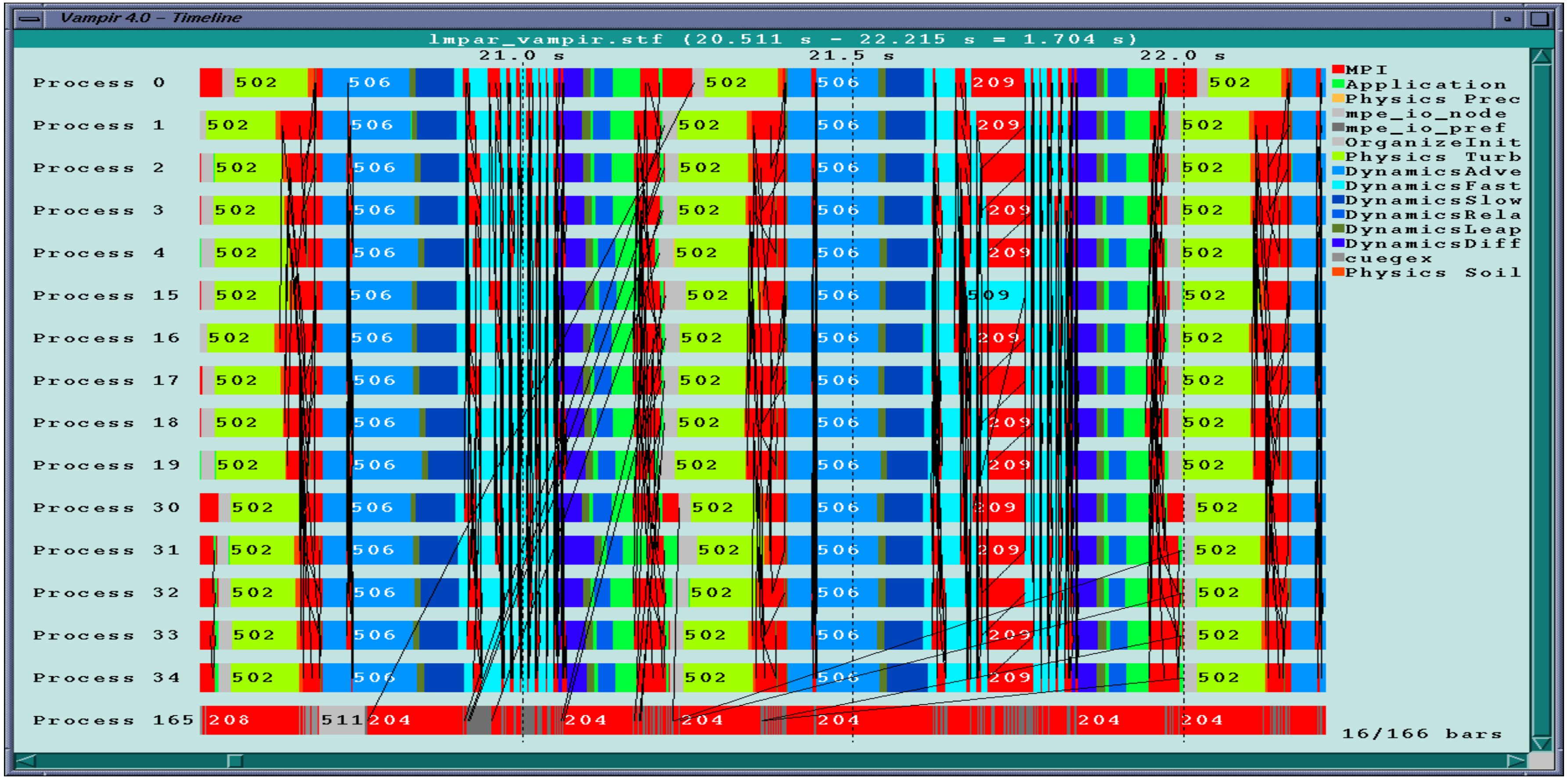
# Optimized LM Time Step

# Optimized Output Step

# Optimized Input Step

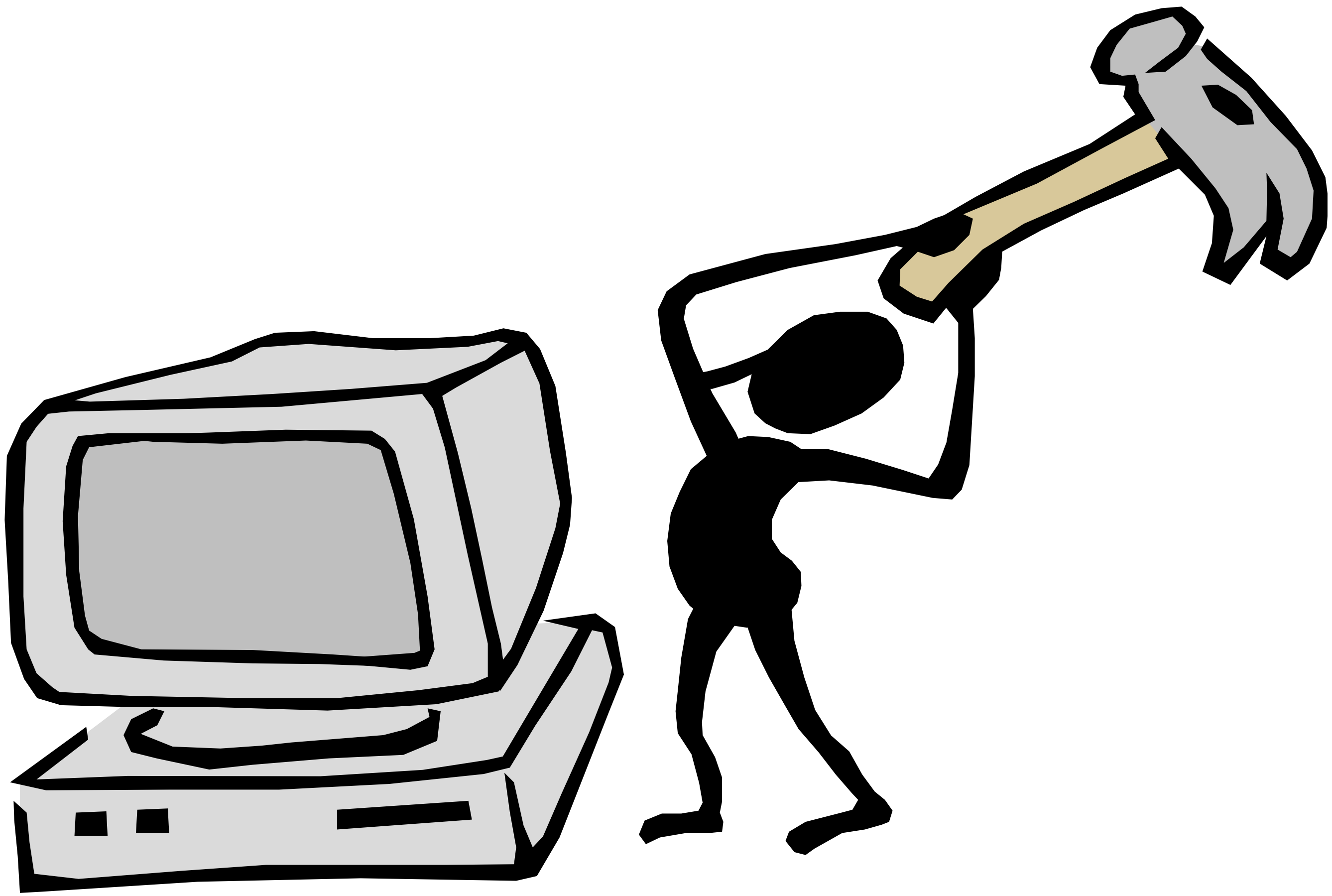


# Optimized Input Step - 2



# LME: Old vs. Optimized Timings

| | Old | Comm+ Output | + Input |
|---|---|---|---|
| Dyn. Computations | 367.51 | 349.29 | 349.65 |
| Communications | 85.30 | 138.96 | 221.58 |
| Barrier waiting | 130.37 | | |
| Phys. Computations | 188.16 | 187.36 | 187.50 |
| Communications | 7.31 | 25.40 | 26.09 |
| Barrier waiting | 31.76 | | |
| Input | 126.62 | 135.29 | 80.70 |
| Output | 258.70 | 142.21 | 142.19 |
| Total Time for the Job | 1285.46 | 1066.35 | 1095.01 |



# Profiling with MPI_Trace (IBM)

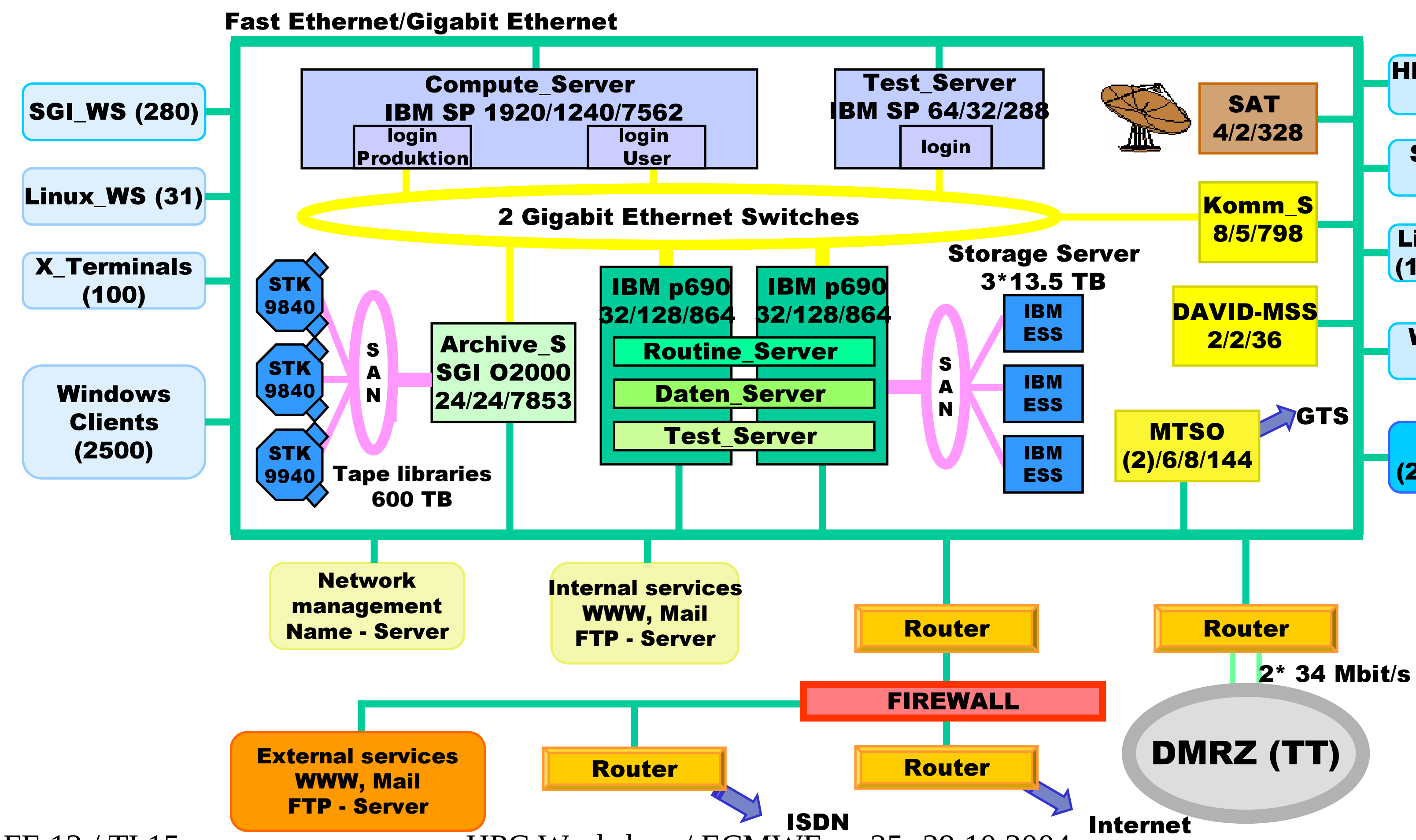
$325 \times 325 \times 35$

| | Old Version | Optimized |
|---|---|---|
| MPI_ALLREDUCE | 0.24 | 3.71 |
| MPI_BARRIER | 30.08 | 0.01 |
| MPI_ISEND | 0.40 | 0.00 |
| MPI_RECV | 8.07 | 3.18 |
| MPI_SEND | 2.46 | 0.00 |
| MPI_SENDRECV | - | 21.49 |
| MPI_WAIT | 0.77 | 0.15 |

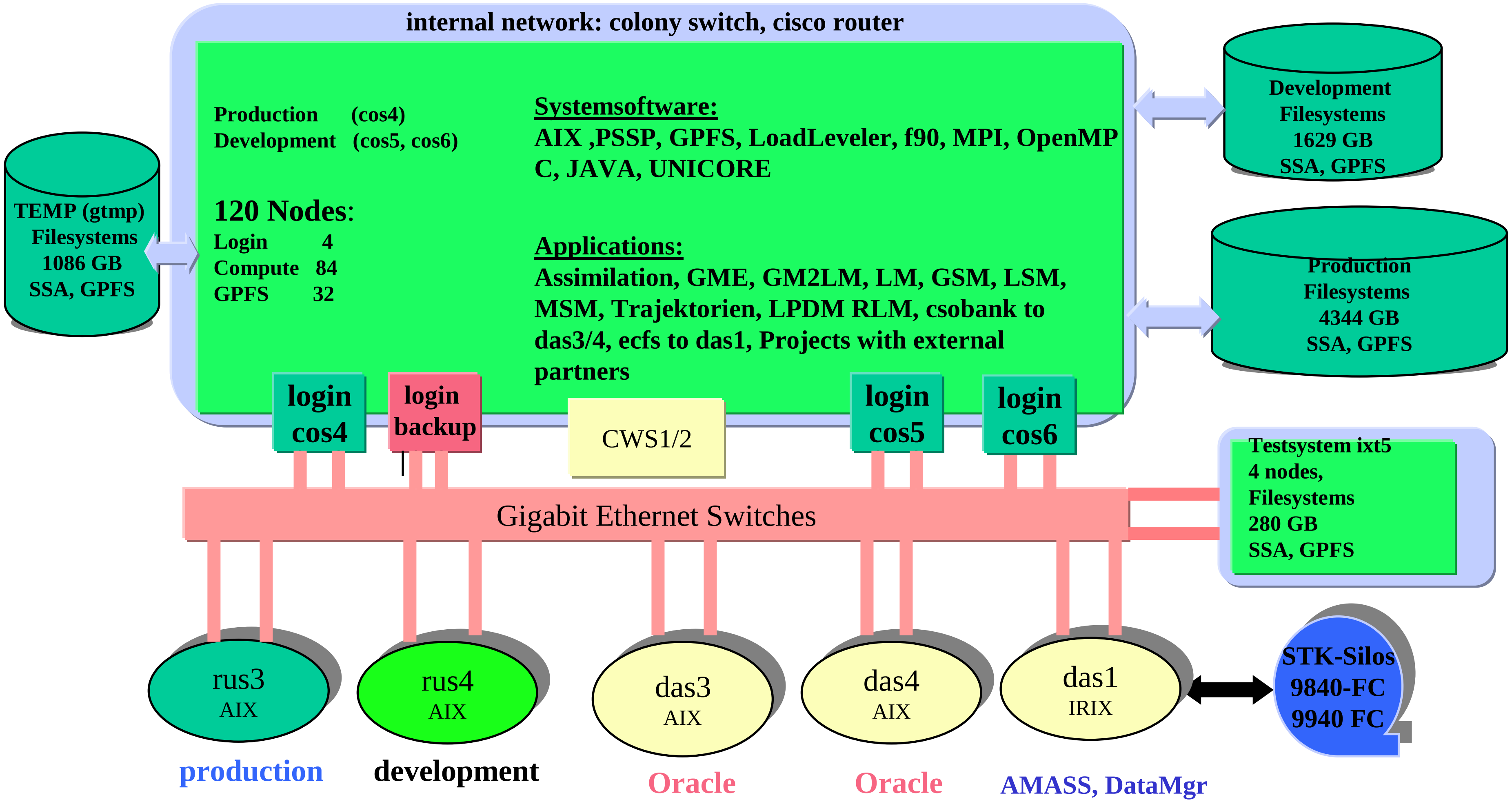
# Tools for Performance Analysis

- Trace Analyzer (VAMPIR)
  - good to detect problems in communication
  - helps to understand your code
- MPI_Trace
  - to detect hot-spots in the communication
  - helps to understand MPI Performance
- HPMCOUNT (Hardware Performance Monitor)
  - produces a lot of data (at least on IBM)
  - results are not easy to understand

# NWP Environment: IT Structure 2004



# IBM SP 9076 550 (NH II, 375 MHZ)



# Future Plans for DMRZ

2005: Application for funds

2006: Invitation to tender

2008: Start of operation in a new building

How does „forecast-process“ look like in 2008?

Performance enhancement by 8-10
(ca. 30 TeraFlop/s peak performance)

Investigate possible use of Linux-Clusters

# LM_RAPS_3.0

- To reflect the recent model changes, a new RAPS-Benchmark has been released
  - Changes in communication for boundary exchange are included
  - Changes for asynchronous IO are not included
- Benchmark is available for vendors (and interested people), if an „Agreement" is signed ($\Rightarrow$ change in the RAPS structure)

# Conclusions

- A programmer´s work is never done
- Getting good communication performance on the IBM is not impossible, but takes time
- Optimizations may be machine dependent (offer choice of selection)
- Next: How to optimize the computations (Flop/s)?

# Thank You