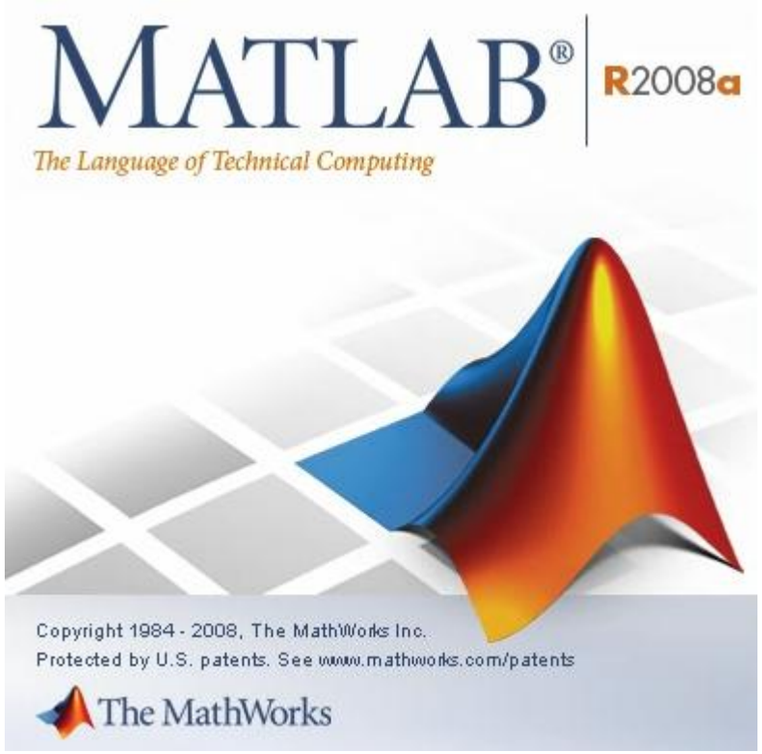
GridMate
The Grid Matlab Extension

Forschungszentrum Karlsruhe,
Institute for Data Processing and Electronics

T. Jejkal, R. Stotzka, M. Sutter, H. Gemmeke

What is the Motivation?

- Graphical development environment
- High-level interpreter language
  - Rapid algorithm development
- Increasing complexity of algorithms
  - Local computation not realizable in acceptable time frame



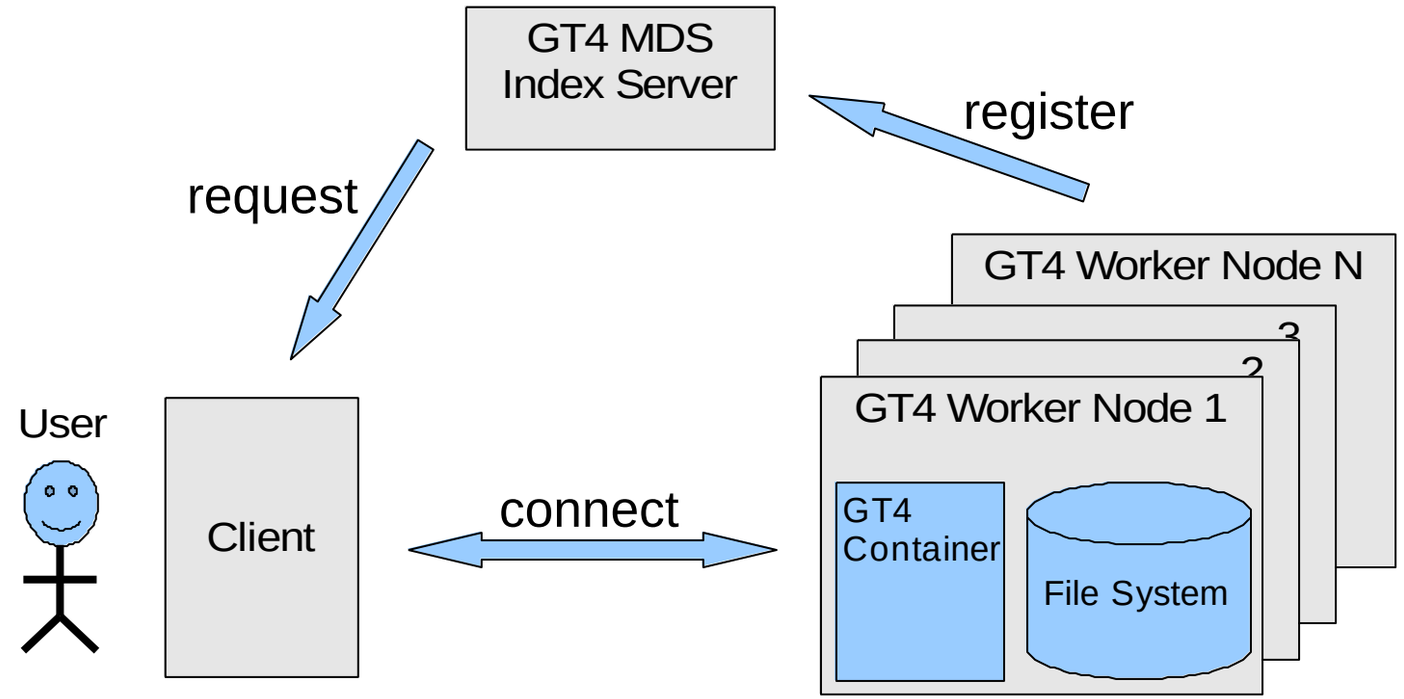
But for the Matlab user the traditional Grid is:

- Complex
- Hard to use

What are the goals?

- Composition of Matlab and Grid
  - For the user:
    - Seamless integration into the users working environment
    - Intuitive and easy parallelization of arbitrary algorithms
      - Least intrusive to existing code
  - For the underlying Grid architecture:
    - Dynamical extensibility
    - Return immediate response
    - Support interactivity

The Grid architecture



- Service Oriented Grid Architecture (SOGA)
- Standard Globus WS-Core extended by some services
- Functionalities implemented as WSRF-compliant Web services
- Platform independent on client and server side

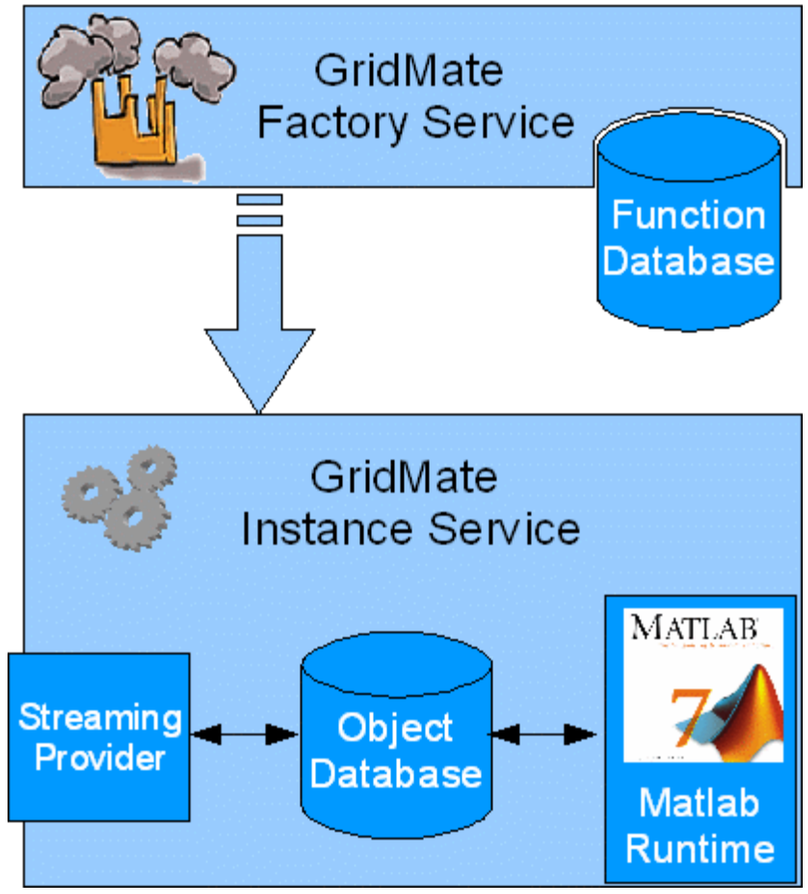
WSRF: Web Services Resource Framework

Design of GridMate - Client

- Integrated into Matlab as toolbox
  - Functionality implemented in Java
  - Can be used stand-alone
- Broker between Matlab user and SOGA
  - No direct contact between user and Grid

Design of GridMate - Server

- WSRF-compliant Web services
- Factory Service for management tasks
  - Function database to hold functions
- Instance Service for computation
  - Streaming Provider for data transfer
  - Object Database to hold arguments
  - MRE to execute Matlab code



MRE: Matlab Runtime Environment

Application 1 – Characteristics

Application Type:

- 3D volume reconstruction

Input:

- 2D signal data
  - 20 datasets, approx. 23 MB each

Special Feature:

- Highly optimized Matlab code
  - Partly implemented in C and assembly code

Application 1 – Introducing GridMate

```
initReconstruction();
for 1 : signals
   loadSignal();
   addSignalToVolume(<Arguments>);
end;
```

```
connect();
initFunction('addSignalToVolume', nodes);
initReconstruction();
for 1 : signals
   loadSignal();
   runFunction(<Arguments>);
end;

for 1 : nodes
   downloadResult();
end;
```

Original / Unmodified code

Slightly modified code

New code

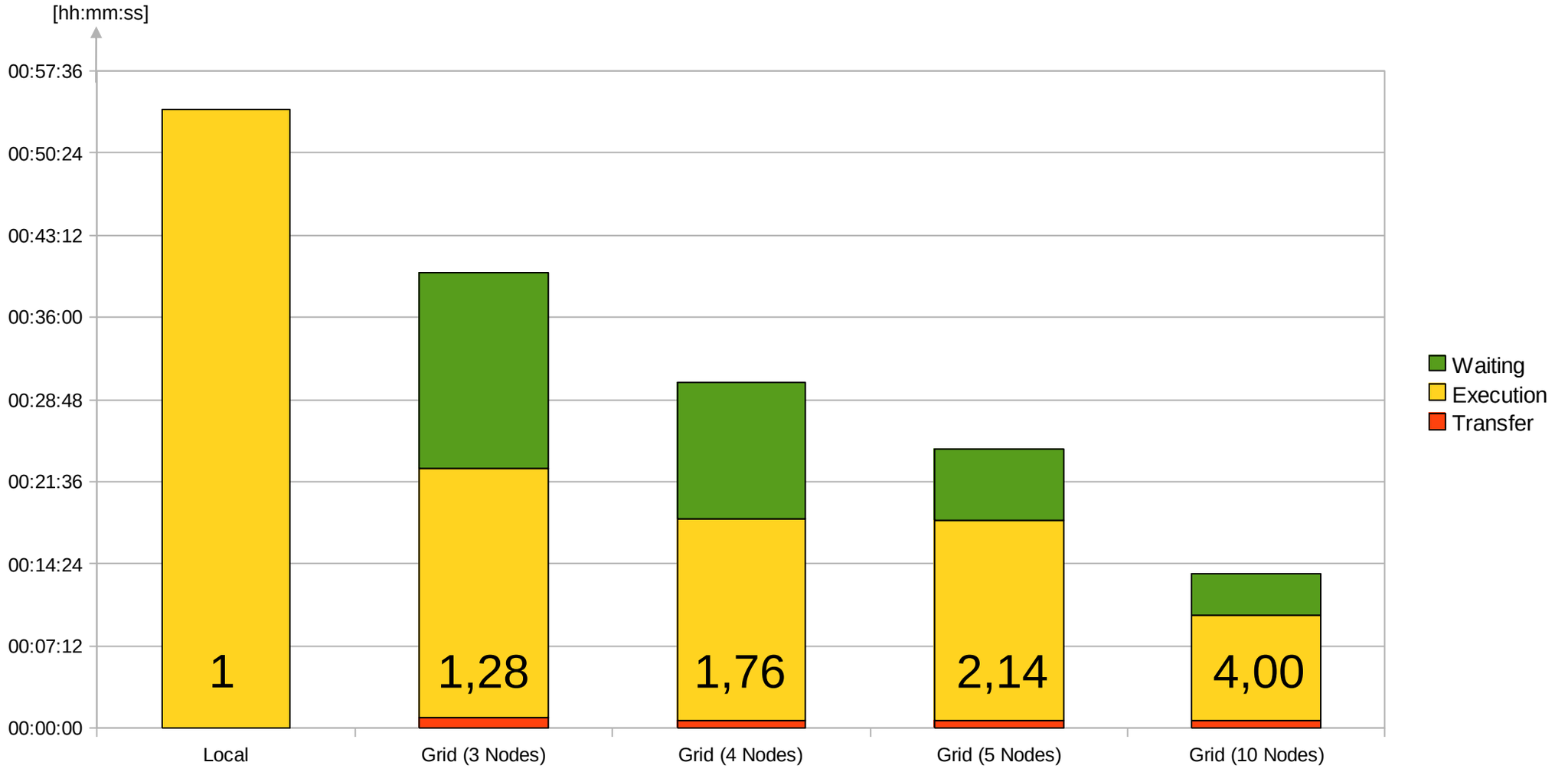
Original code

Using GridMate

Application 1 – Performance Measurement

Local: Intel P4 3.2 GHz, 2 GB Ram, 100 Mbit Network, Windows XP

Grid: VM@Intel Xeon, 2 GB Ram, 100 Mbit Network, OpenSUSE 10.3



Application 2 – Characteristics

Karlsruhe Institute of Technology

Application Type:

- Digital image correlation

Input:

- 2D images
  - 17 images, approx. 20 MB each

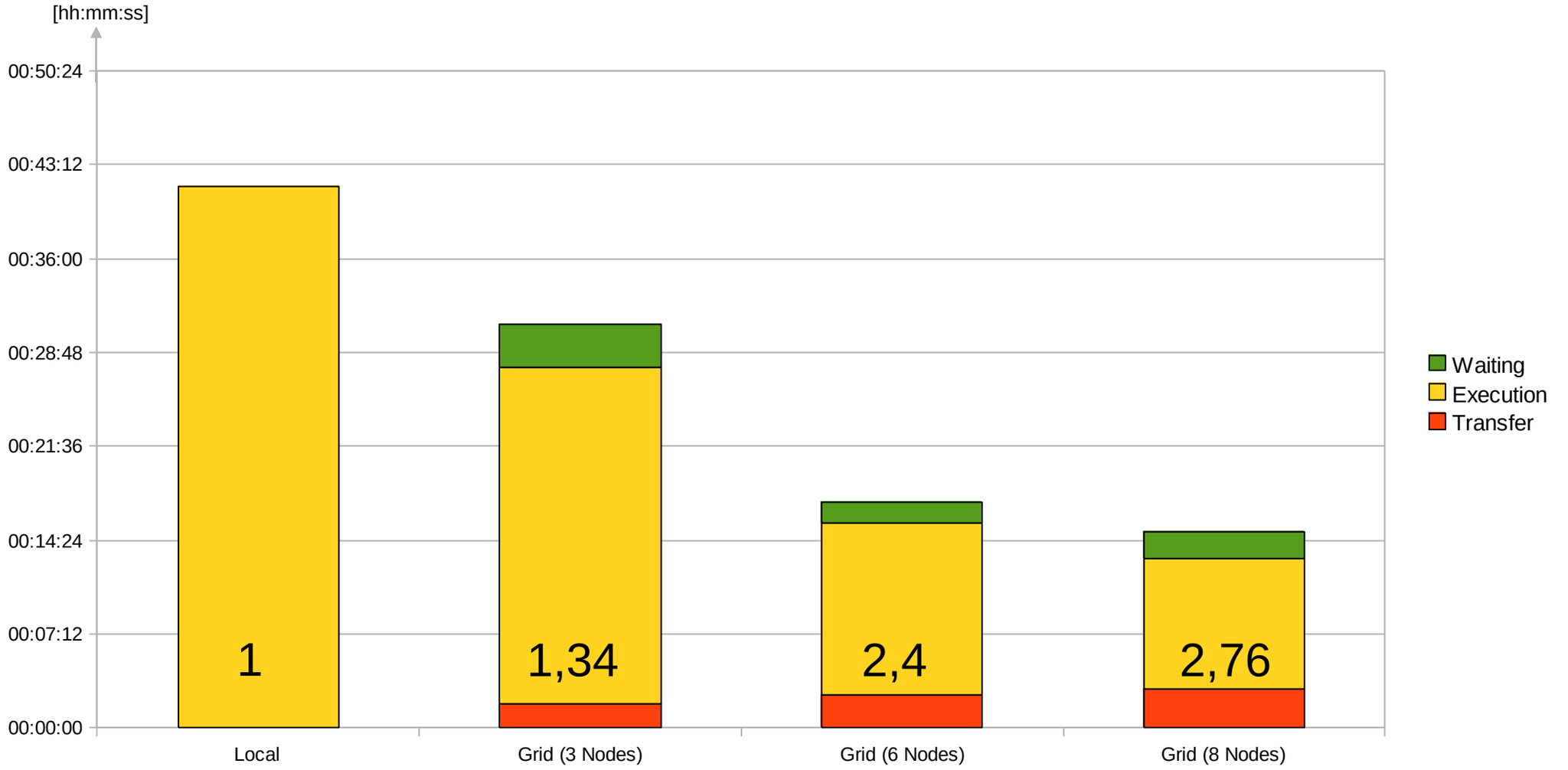
Special Feature:

- Images must be processed synchronously
  - Each node needs each image (5.4 GB for 16 Nodes)

Application 2 – Performance Measurement (1)

Local: Intel Core2Quad 6600, 8 GB Ram, 100 Mbit Network, OpenSUSE 10.3

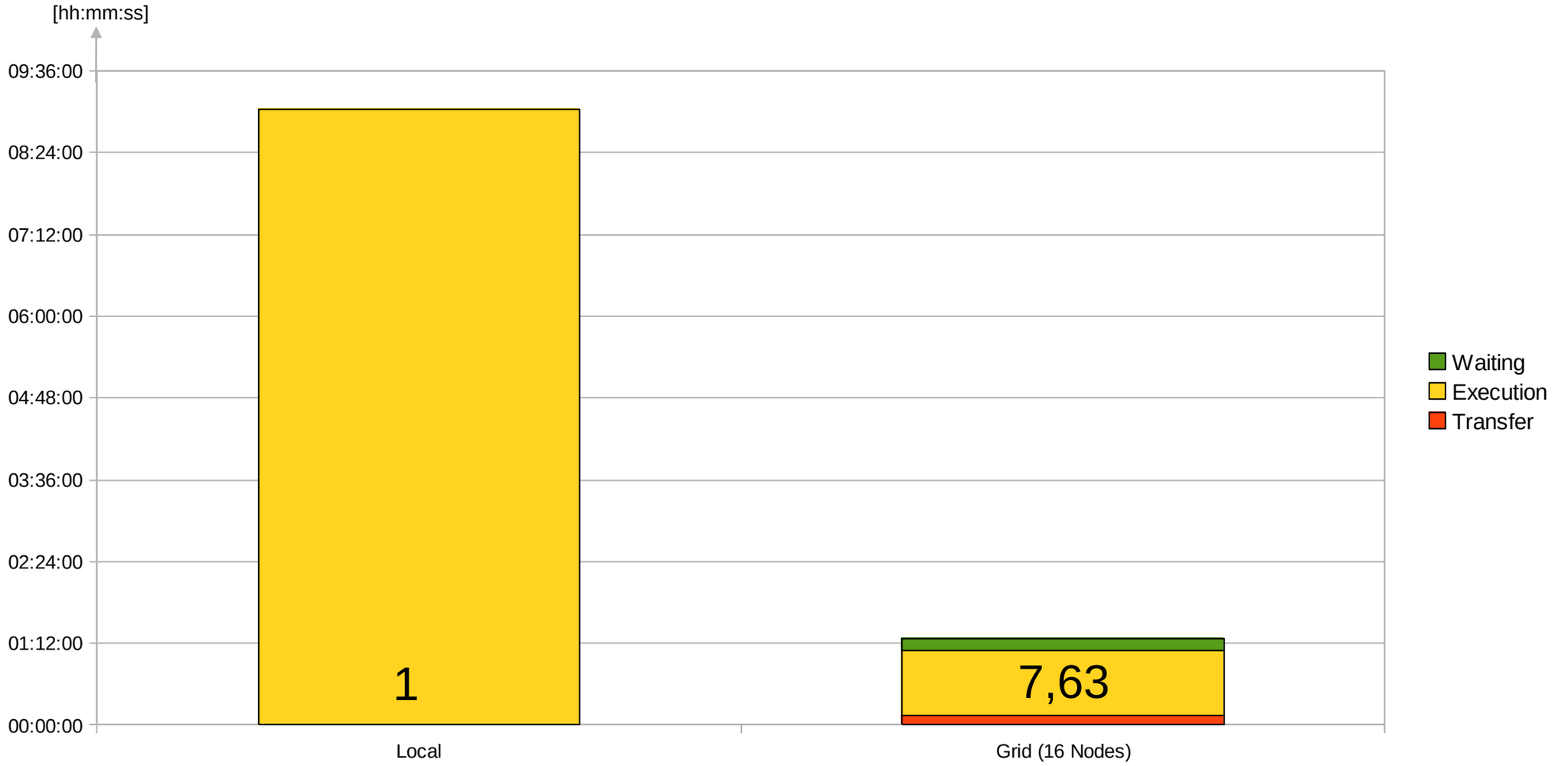
Grid: VM@Intel Xeon, 2 GB Ram, 100 Mbit Network, OpenSUSE 10.3



Application 2 – Performance Measurement (2)

Local: Intel Core2Quad 6600, 8 GB Ram, 100 Mbit Network, OpenSUSE 10.3

Grid: VM@Intel Xeon, 2 GB Ram, 100 Mbit Network, OpenSUSE 10.3



Conclusions

Realized goals:

- Seamless integration into the users working environment
- Intuitive and easy parallelization of arbitrary algorithms
- Dynamically extensible
- Immediate response and interactively accessible

Future work:

- Integrate access to Grid storage solutions
- Realize checkpoint system to increase fault tolerance