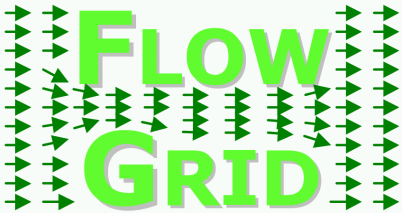
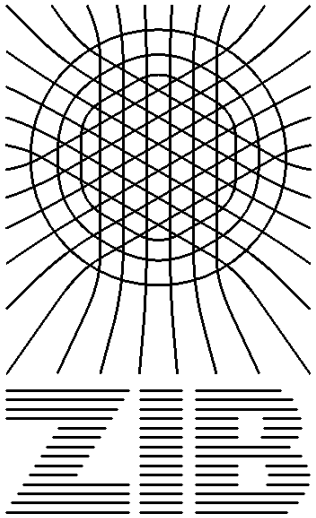


Computational Fluid Dynamics in the Grid using FlowGrid

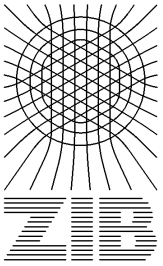


Jan Wendler (wendler@zib.de)

ZIB (Zuse Institute Berlin)

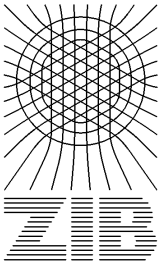
Cracow Grid Workshop'03, 27th October 2003

Outline



- Motivation: Goal, CFD-applications, properties of CFD simulations
- FlowGrid Architecture
- Protocol between FlowServe (Grid middleware) and GenIUS (user client)
- Features of FlowGrid
- Outlook

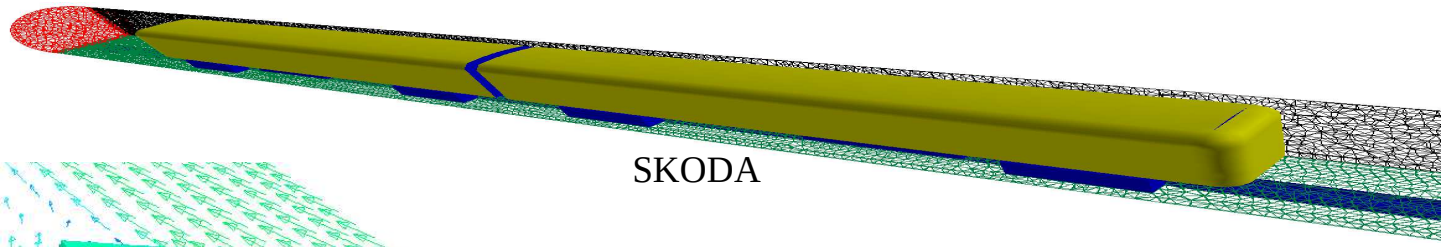
Revolutionize the way CFD simulations are set up, executed and monitored



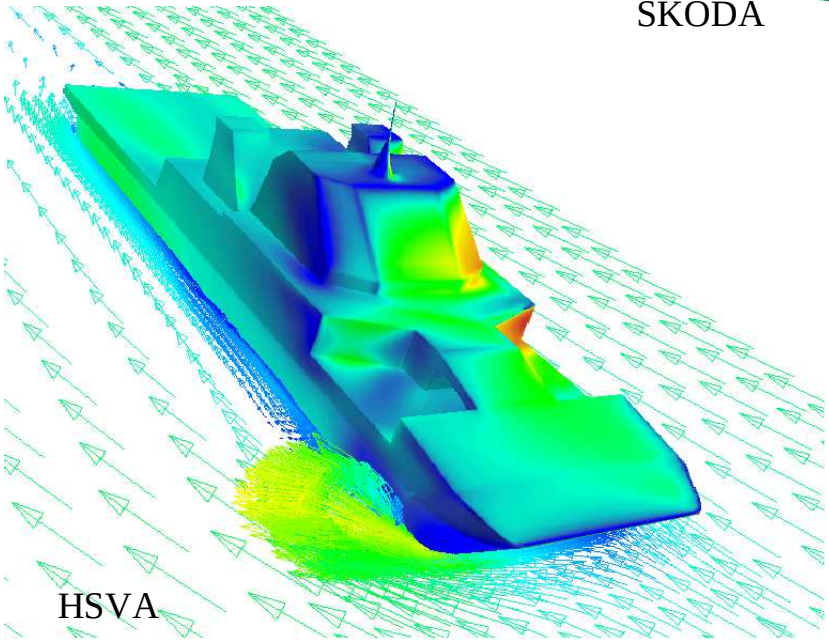
- Run CFD (Computational Fluid Dynamics) simulations on-demand
  - Company: resources insufficient or often stay idle
- Virtual organisation
  - GRID environment (uniform interfaces) for the solution of CFD problems
  - Share software and computing resources
  - Computing resources distributed
    - geographically, organisationally
  - User-friendly and efficient access

Real Challenging CFD-Applications (Partner)^2

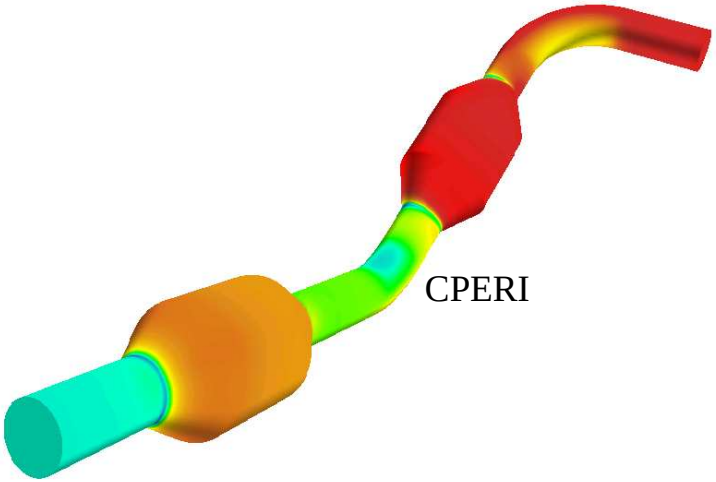
- Shipbuilding, exhaust gases, combustion of fossil fuels, trains travelling through tunnels, …



SKODA

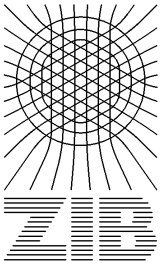


HSVA



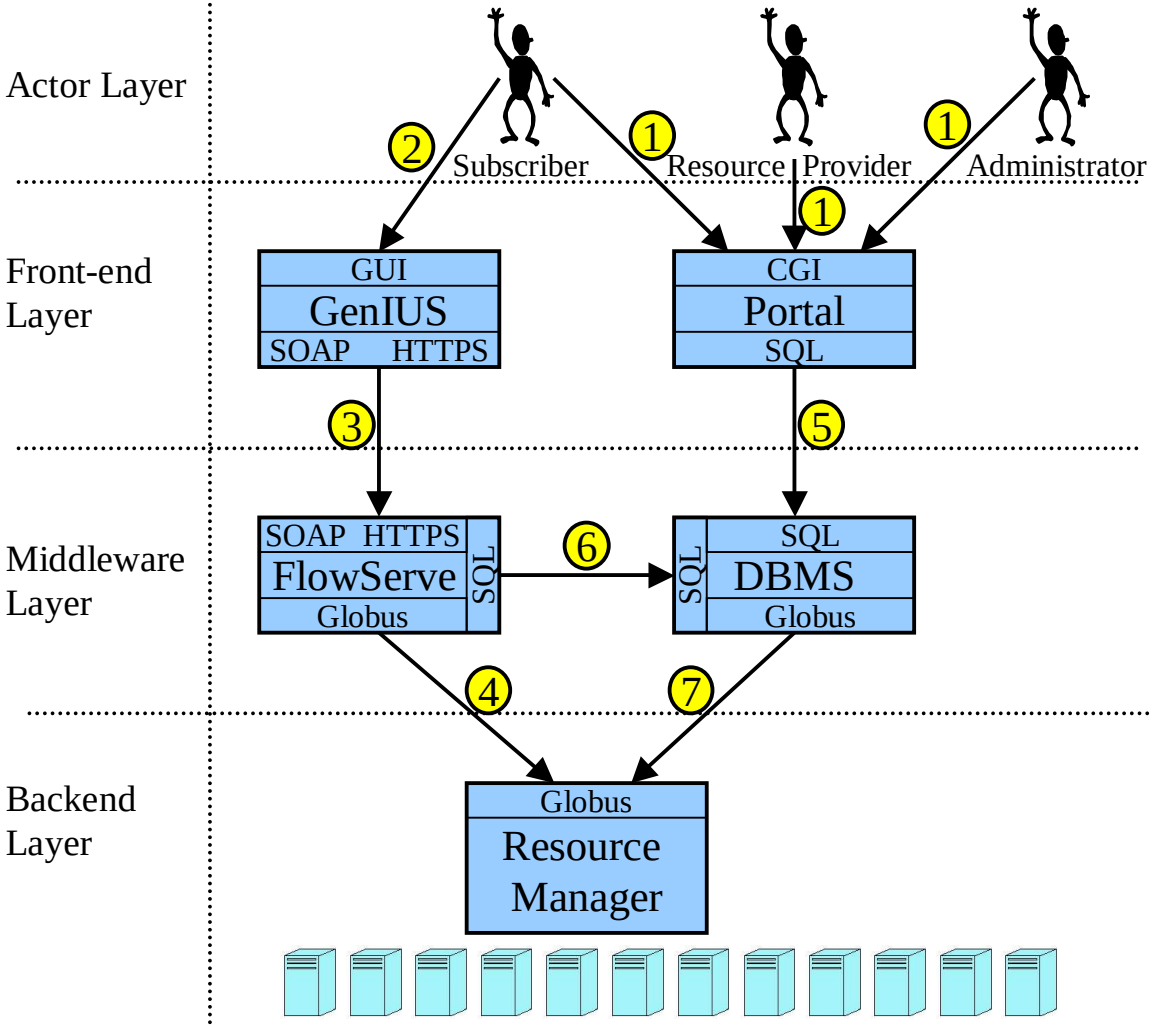
CPERI

Properties of CFD simulations



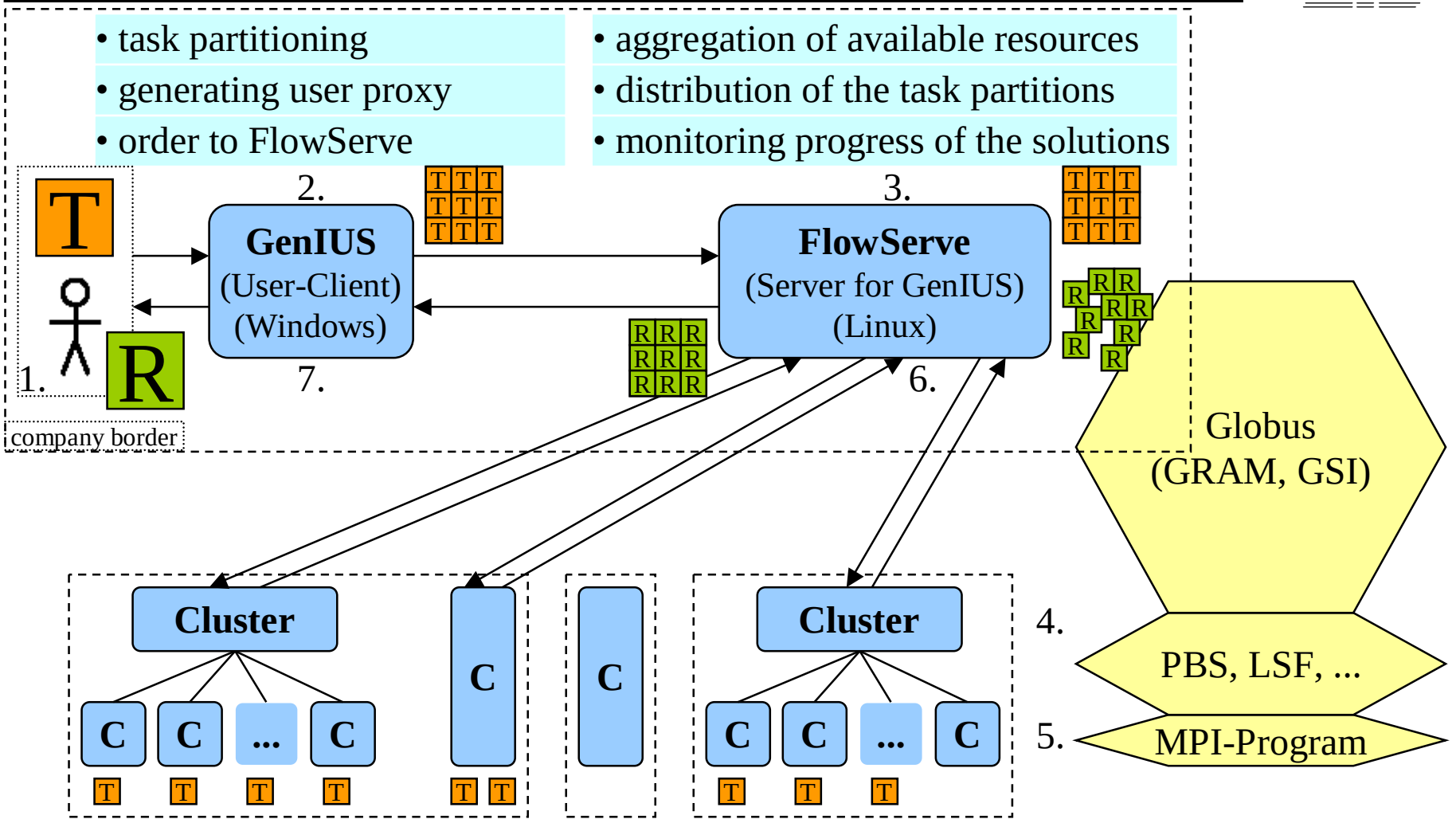
- Current CFD applications
  - Run on homogenous clusters
  - Allow monitoring and adaptations during runtime
  - Examples: FLUENT, STARCD, CFX, …
- Special property of CFD simulations
  - Synchronous communications between subjobs

FlowGrid Architecture

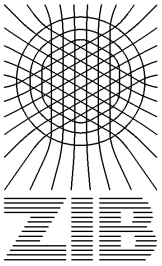


1. Getting/Setting data, downloading GenIUS
2. User-Interaction via a GUI
3. Communication via SOAP, file transfer via HTTPS
4. Running and monitoring a job on the Grid using GT2
5. Updating DBMS and retrieving information
6. Update of resource information (availability …)
7. Retrieving accounting information from the RMs

Typical Run of a Job

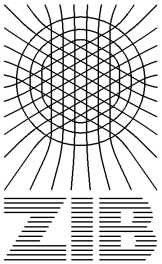


FlowServe - GenIUS Protocol: General Info



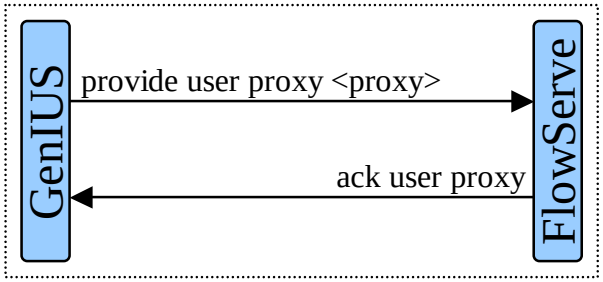
- FlowServe runs jobs using the user's proxy certificate
- All interactions are initiated by the user client
- Protocol as general as possible
  - Control communication by SOAP messages
    - Preferred GSI-enabled SOAP
  - Data transfer by https protocol
- FlowServe offers a Web-service / Grid-service to CFD applications
  - several functions (user-triggered, time-triggered)

Time-triggered Functions

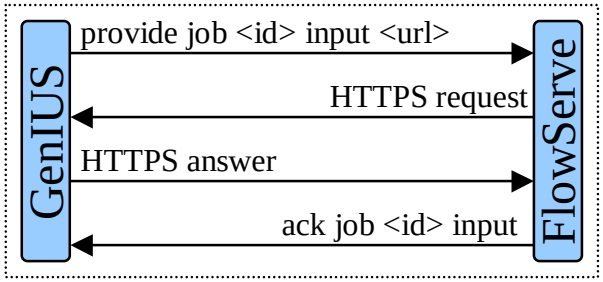


- Goal: limit the response time of FlowServe by requesting information and generating reports in advance, can also be executed on requests
  - Generate resource information data
    - [FlowServe -> DBMS + Information Provider]
  - Generate job's general information (accounting info etc.)
    - [FlowServe -> JobManager -> Controljob]
  - Download job's progress information
    - [FlowServe -> JobManager -> Controljob]
  - Download job's output data
    - [FlowServe -> JobManager -> Controljob]

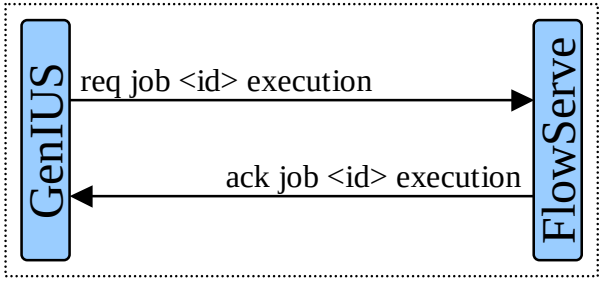
FlowServe Protocol: Standard Interactions 1



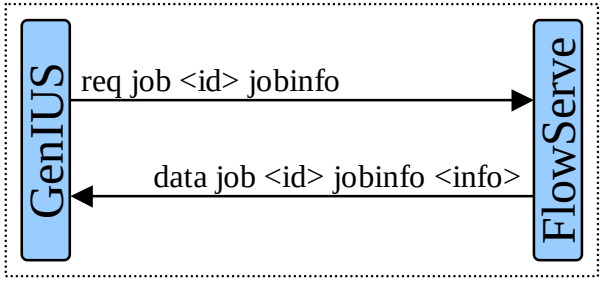
(a) update proxy certificate



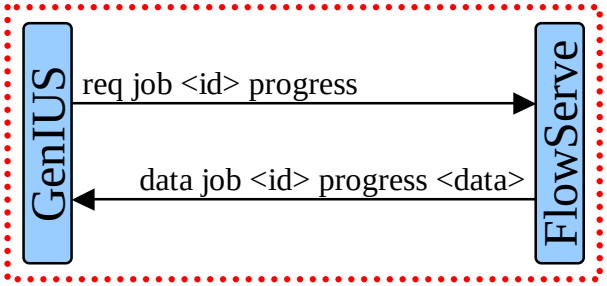
(b) provide input data



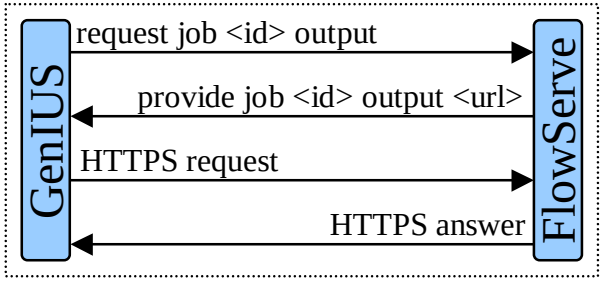
(c) job execution



(d) job status information

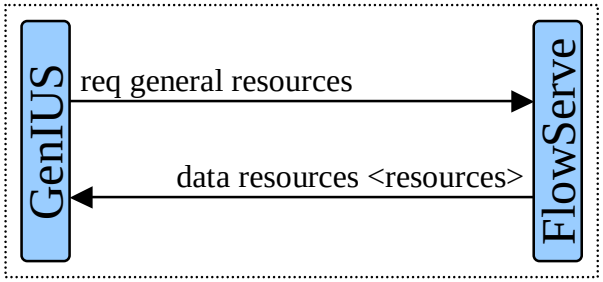


(e) job progress information

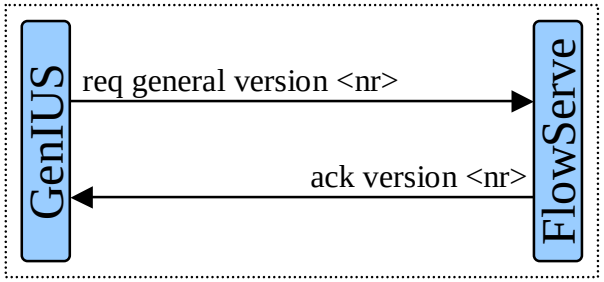


(f) download output data

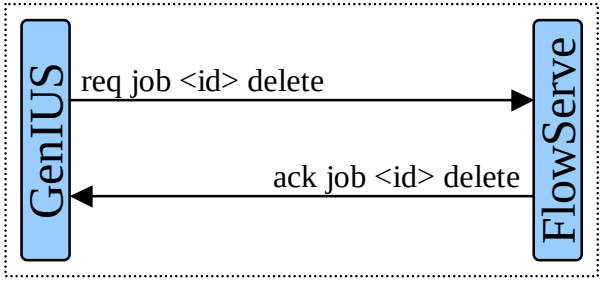
FlowServe Protocol: Standard Interactions 2



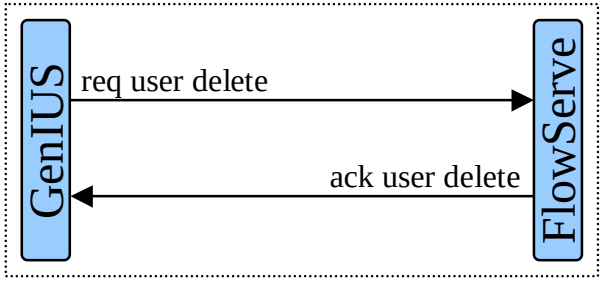
(g) resource information



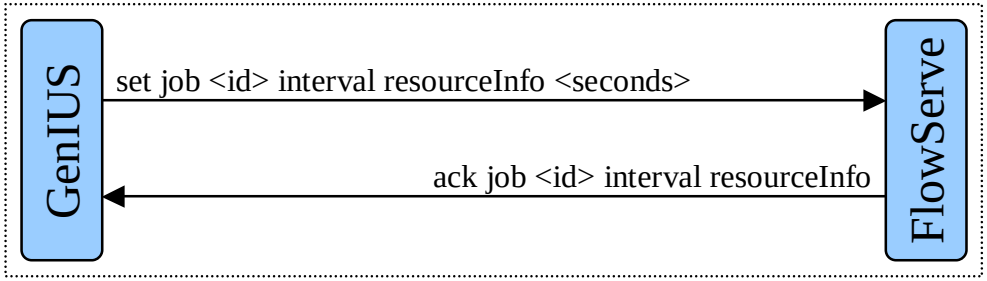
(h) protocol version



(i) job delete



(j) delete all

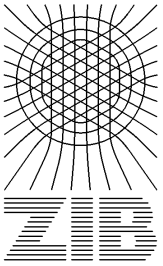


(k) set update intervals

Similar interactions for:

- **jobinfo**
- **progress**

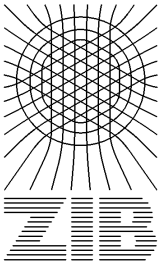
FlowServe Protocol: Generalization



- Goal
  - Make FlowServe as general as possible so it can be universally used by other (CFD-)applications (FLUENT, StarCD, CFX)

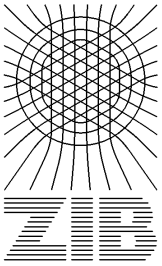
- New functions of FlowServe
  - Allow specification of progress and output files
  - Allow retrieval of any job specific file
  - Provide input data during simulation
    - Changing variables

Features of FlowGrid and FlowServe



- Combination of Microsoft Windows and Linux within a single Grid environment
- Preliminary results are provided to the user during runtime
  - Possibility to discover problems already during runtime
- FlowServe does support adaptations to simulation parameters during runtime
- General FlowServe protocol, allowing other client-solver-pairs to make use of FlowServe

Outlook



- Heterogeneous computing resources
  - Partitions with different sizes
  - Automatic identification of best partition sizes
- Dynamic job partitioning
  - Re-partitioning during runtime
- Reservation of resources among different clusters
  - Unique start time of all subjobs (synchronous comm.)
  - Minimizing latency

ZIB (Zuse Institute Berlin), Computer Science Division,
Computer Science Research Department

E-Mail: wendler@zib.de
Web : www.zib.de