

Geant 4

*IEEE Nuclear Science Symposium and Medical Imaging Conference
Short Course*

Simulation Techniques Using Geant4

Maria Grazia Pia (*INFN Genova, Italy*)
MariaGrazia.Pia@ge.infn.it

Dresden, 18 October 2008

<http://www.ge.infn.it/geant4/events/nss2008/geant4course.html>

This course exploits training material developed by several Geant4
Collaboration members: thanks to all of them!

Execution in a distributed computing environment

Parallelisation

Speed of Monte Carlo simulation

Speed of execution is often a concern in Monte Carlo simulation
Often a trade-off between **precision** of the simulation and **speed** of execution

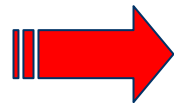
Semi-interactive response

- Detector design/optimisation
- Oncological radiotherapy

Very long execution time

- High statistics simulation
- High precision simulation

Methods for faster simulation response

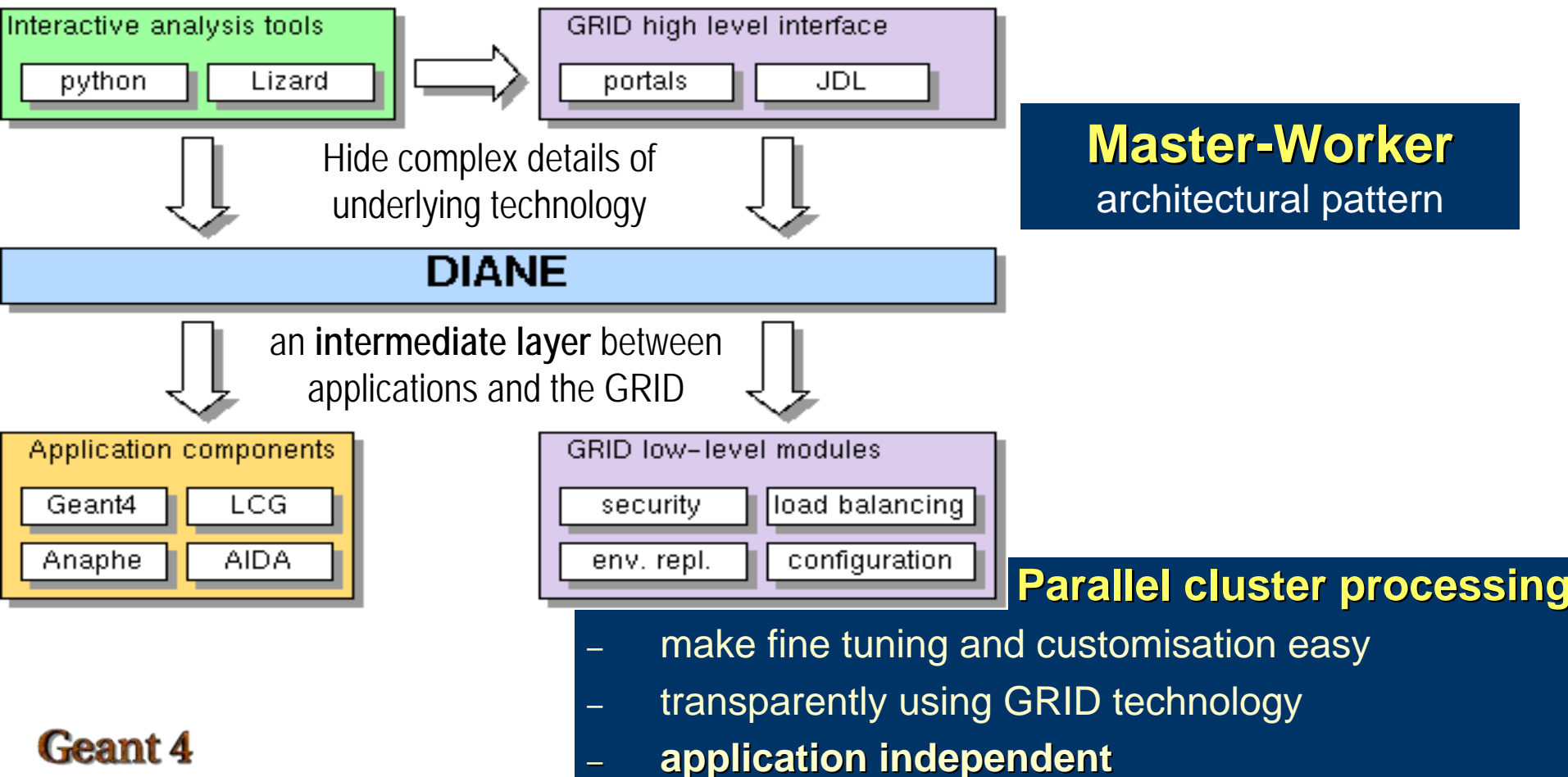


Fast simulation
Variance reduction techniques
(event biasing)
Inverse Monte Carlo methods
Parallelisation

Parallel/distributed execution of Geant4 applications

- Monte Carlo simulation for radiation transport is relatively easy to execute in parallel mode
 - “natural” parallelism at event level
- Various options available to use with Geant4
 - It is not a specific Geant4 issue
- Execution in a distributed computing environment
 - In a local farm
 - In a geographically distributed computing grid
- Ideally transparent to the user
 - Same **sequential/parallel** code
 - Same user code in **local farm/grid**

Intermediate layer to shield application developers/users from the complexity of the underlying computing environment



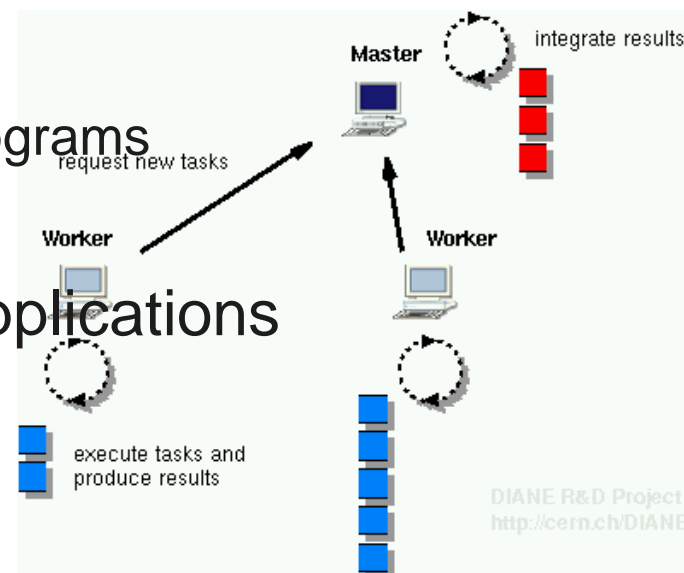
Practical example: Geant4 simulation with analysis

• Master-worker model

- client starts a job
- workers perform tasks and produce histograms
- master integrates the results

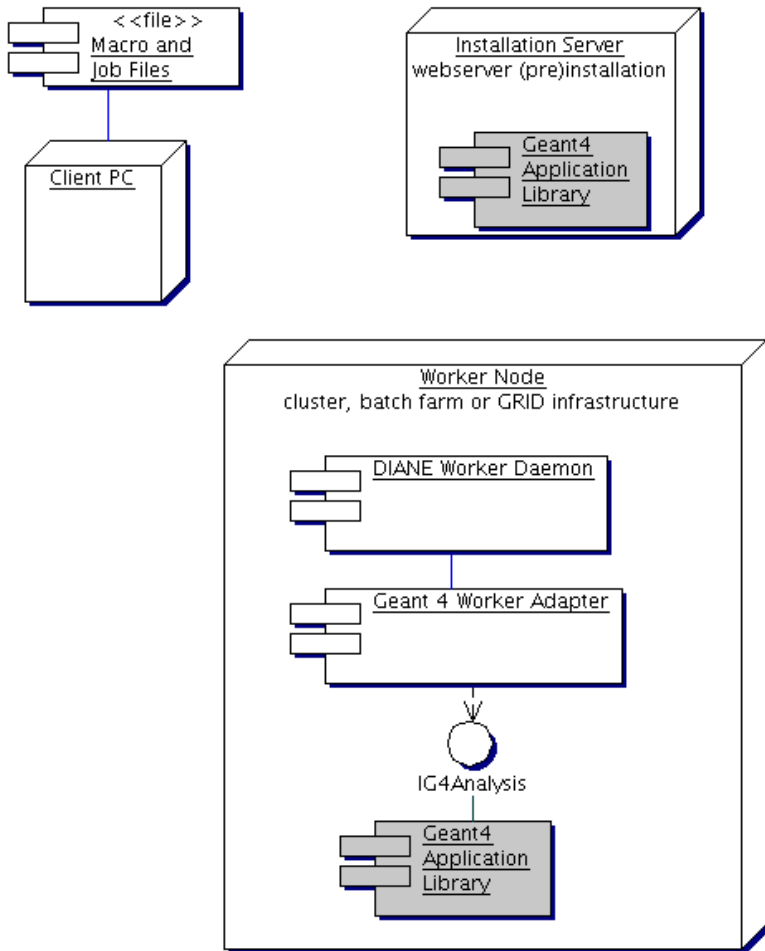
• Distributed Processing for Geant4 Applications

- task = N events
- job = M tasks
- tasks may be executed in parallel
- tasks produce histograms/ntuples
- task output is automatically combined (add histograms, append ntuples)

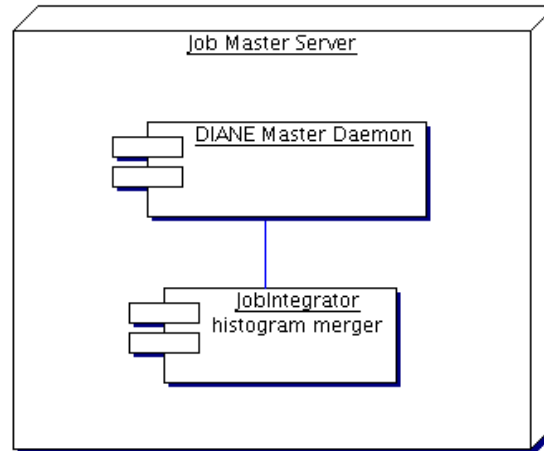


• Master-Worker Model

- **Master** steers the execution of job, automatically splits the job and merges the results
- **Worker** initializes the Geant4 application and executes macros
- **Client** gets the results



UML Deployment Diagram for Geant4 applications



Geant 4 simulation with DIANE

- Completely transparent to the user: same Geant4 application code
- G4Simulation class is responsible of managing the simulation
 - manage random number seeds
 - Geant4 initialisation
 - macros to be executed in batch mode