



## UvA-DARE (Digital Academic Repository)

### Understanding and mastering dynamics in computing grids: processing moldable tasks with user-level overlay

Mościcki, J.T.

**Publication date**  
2011

[Link to publication](#)

#### **Citation for published version (APA):**

Mościcki, J. T. (2011). *Understanding and mastering dynamics in computing grids: processing moldable tasks with user-level overlay*.

#### **General rights**

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

#### **Disclaimer/Complaints regulations**

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

- agent factory, [64](#), [81](#), [87](#), [126](#), [130](#), [137](#),  
[138](#), [140](#)
  - heuristic, [82](#), [84](#)
  - hybrid, [85](#), [87](#)
  - simple, [81](#), [82](#)
- Amdhal's Law, [7](#)
- ARC , [64](#)
- Athena analysis framework, [102](#)
- ATLAS, [27](#), [102](#), [103](#), [146](#)
  - VO, [44](#), [46](#), [48](#)
  
- BLAST (Basic Local Alignment Tool), [10](#),  
[15](#), [106](#)
  
- Capability Computing, *see* HPC
- Capacity Computing, *see* HTC
- Central Limit Theorem, [50](#)
- CERN, [4](#), [8](#), [20](#), [23](#), [46](#), [55](#), [102](#), [114](#), [122](#)
- CMS
  - VO, [46](#)
- Condor, [6](#), [9–11](#), [34](#), [64](#)
  - M/W, [10](#), [11](#)
- connection management, [71](#)
- CORBA, [69](#), [80](#)
- CREAM, [8](#), [34](#)
  
- degree of parallelism, [3](#)
- DEISA, [9](#)
- DIANE framework, [48](#), [55](#), [59](#), [63–68](#), [71–](#)  
[73](#), [76](#), [77](#), [80](#), [82](#), [84](#), [93–95](#), [97](#),  
[99–106](#), [113](#), [115](#), [121](#), [125](#), [126](#),  
[128](#), [129](#), [140](#), [145](#), [146](#), [150](#), [157](#),  
[158](#)
- DIRAC workload management system, [45](#),  
[102](#)
- divisible load, [42](#), [50](#), [86](#)
  
- early binding, [10](#), [11](#), [16](#), [28](#), [43](#), [56](#), [92](#), [93](#),  
[101](#), [104](#)
- EGEE Grid, [80](#)
- evolving jobs, [2](#)
  
- FORTRAN, [7](#), [130](#)
  
- Ganga interface, [59](#), [63–66](#), [73–82](#), [84](#), [93](#),  
[95](#), [99–107](#), [113](#), [115](#), [121](#), [126](#),  
[140](#), [145](#), [146](#), [150](#), [157](#)
- Gaudi analysis framework, [102](#)
- Geant4, [91](#), [97](#)
  - VO, [31](#), [44](#), [46](#), [48](#), [93](#)
- glxexec, [35](#)
- gLite workload management system, [67](#), [76](#)
- Globus, [64](#)
- Google
  - Summer of code, [106](#)
- GSI, [72](#), [76](#)
- Gustafson's Law, [7](#)
  
- High Performance Computing, [5](#), [9](#), [10](#), [125](#),  
[141](#)
- High Throughput Computing, [5](#), [8](#), [43](#)
- HPC, *see* High Performance Computing

- HTC, *see* High Throughput Computing
- in silico experiments, 104
- IPython interface, 73
- job queuing time, 37, 43–46, 48, 53, 144  
dispersion coefficient, 52, 54
- job submission, 33, 41, 56, 64, 76, 83, 97,  
102, 104, 117
- Kerberos, 76, 80
- late binding, 10, 11, 16, 41, 43, 48, 53, 57,  
61, 62, 89, 101, 105, 121, 144
- layering, 67
- LHC project, 102
- LHCb, 102, 103, 146  
VO, 44, 46, 48
- LSF, 9, 64, 81
- makespan, 14, 43, 50, 54, 98, 102, 116, 144  
bound of, 49  
distribution, 42, 48
- malleable jobs, 2
- Many Task Computing (MTC), 5
- Massively Parallel Processors (MPP), 9
- moldable jobs, 2
- Moldable Task Applications, 2, 7
- moldable workload, 42
- Monte Carlo simulation, 3, 4, 6, 14, 42, 55,  
91, 95, 121, 123, 125, 127, 132
- monte Carlo simulation, 48
- MPI, 7, 10, 13, 15, 66, 76, 92, 139, 140, 146
- MTA  
see Moldable Task Applications, 2
- MTC (Many Task Computing), 5
- MyProxy, 35
- non-deterministic selection, 82
- NWS, 11
- omniORB, 66, 71
- OpenMP, 7, 66, 139, 140, 146
- PANDA workload management system, 11,  
27, 102, 103
- PBS, 9, 64  
pilot jobs, 10  
placeholder scheduling, 10  
prioritization, 60, 130  
Python programming language, 61, 62, 66–  
69, 73, 75, 83, 96, 107, 125
- Quality of Service, 9, 12, 13, 16, 41, 50, 53,  
57, 62, 144  
metrics, 48, 53
- resource discovery, 83
- resource heterogeneity metric, 89
- resource selection, 62–64, 80, 82, 85
- resubmission, 6, 14, 37  
deep, 28  
shallow, 28
- rigid jobs, 2
- SAGA, 12, 64, 140
- scheduling, 10, 14, 53, 60, 68, 72, 89, 93,  
117, 130
- security, 5, 11, 12, 72, 76
- SGE, 64
- software bus, 67
- speedup, 7, 53, 57, 81, 85, 88, 89, 97, 104,  
110, 140
- SSL, 72
- Symmetric Multi-Processors (SMP), 9
- task paging, 54
- taxonomy of parallel jobs, 2
- Tera Grid, 9, 139, 140
- VOMS, 35
- WISDOM, 104, 105
- WLCG, 8, 82
- WMS, *see* workload management system
- workload balancing, 85–89, 117, 120, 144  
effects, 54
- workload management system, 12, 21, 27,  
35, 44, 81–83