

Performance study of ARC middleware components



J. Cernak, M. Kocan, and E. Cernakova

P. J. Safarik University in Kosice

Kosice

Slovak Republic

EMI is partially funded by the European Commission under Grant Agreement RI-261611

- Introduction (motivation)
- New ARC middleware features implemented during EMI project (2010-13)
- Test descriptions and the results
- Conclusions

- The goal of performance studies of ARC middleware is to evaluate quality of delivered software in the long-term period.
- Systematic studies started in KnowARC project [1,2]. After KnowARC project a WIKI page was created to share experience between testers and developers[3]. The valuation of key performance parameter continued in the EMI project [4].
- New requirements on ARC Middleware progressively raise (users, security, standards,...). Implementation of a new features is connected with changes of the software. Obviously, each change can influence performance.
- Our motivation was to monitor and analyse an influence of main software changes on selected performance parameters.

Simplifyfication of the problem



- Topology of grid infrastructure is complex:
 - Network of servers and clients
 - Existence of hierarchy
- We assume that single connection “server-client” provides important information about performance parameter
- We focus to study properties of the single connection i.e. topological network factors are not subject of our study.
- To study the performance parameters of networks a different approach and tools are needed.

New features implemented during EMI project (2010-13) [4]



- ARC computing element
 - Support for EMI ES interface version 1.16 is implemented in A-REX service
 - Communications with ARGUS PEP and PDP are implemented and integrated into A-REX
 - CAR 1.2 accounting messages can be sent to an APEL destination
 - nordugridmap is updated to version 2.0
 - DTR is the default mechanism for A-REX data staging
- ARC client
 - EMI ES interface is implemented in ARC client tools
 - xRSL parser is extended to parse "countpermode" and "exclusiveexecution" attributes used for expressing parallel setup requirements
 - New arcrename command as part of data clients
 - arcproxy supports the detection of NSS DB from default location of Firefox, Seamonkey and Thunderbird.
 - A new package nordugrid-arc-arcproxyalt has been released, containing a version of the arcproxy command line tool which depends on CAnL++
 - Some test and misc tools are moved to a new sub package nordugrid-arc-misc-utils: arcwsrf, arcperftest (renamed from perftest), saml_assertion_init
 - A new test utility arcemiestest is part of nordugrid-arc-misc-utils
- ARC Information system
 - The information system start-up script grid-infosys has been split into separate scripts. The scripts to run depend on whether you are running a local info system (ARIS) or information index (EGIIS).
 - GLUE2 rendering is finalised, also in the EMI-ES context
 - Introduced a relay backend in LDAP configuration for interoperability with GLUE2 Top-BDII, and changed the top element of Glue1 schema to match the needs of Glue1 Top-BDII. Now an ARC CE can directly be aggregated by a Top-BDII if ARIS LDAP
 - URL is configured correctly in GOCDB.
 - Improved automatic slapd/BDII backend configuration. In particular, BDII paths are automatically generated and can be removed from arc.conf. Removing them from arc.conf during update is recommended, unless one knows exactly how BDII configuration works.
- ARC gridftp

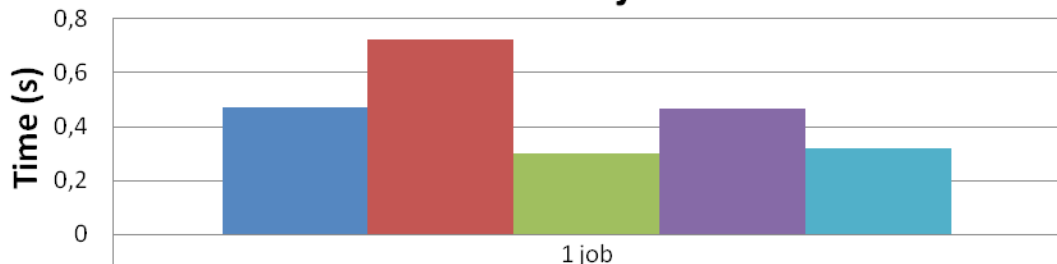
The hardware is the same for whole period of the performance study 2007-2013.

Processor	AMD Athlon 64 processor 3500+
Memory	1024 MB
Ethernet card	NVIDIA nForce 430 Gigabit ethernet
Operating system	Scientific Linux 5.3 64bit
Kernel version	2.6.18-128.4.1.el5
Page size	4096 bytes

- OS Scientific Linux 5.3 64bit
- Middleware: EMI 2.0.1, EMI 3.0.0
- Service configurations are public available [3].

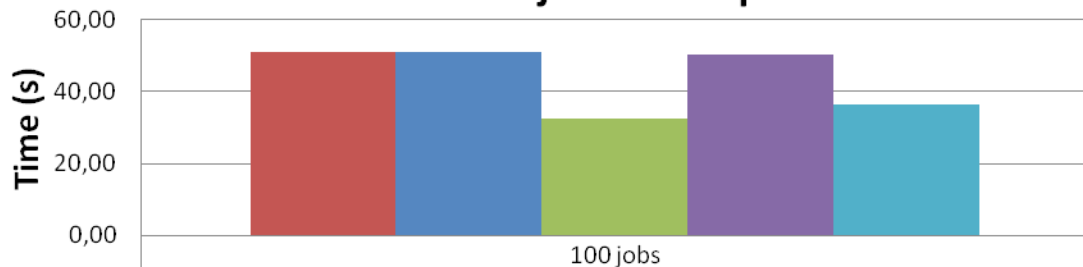
- There is no brokering involved since the target is requested directly with the `-c` command line option. The time measured corresponds to the time elapsed between executing the first `arcsub` command and receiving the last grid job ID indicating successful job submissions.
- Each scenario is repeated multiple times and resulting time is average from the times obtained.

Job submission - 1 job



■ WS-ARC preEMI	0,470666667
■ WS-ARC EMI2	0,725
■ non-WS ARC EMI2	0,302
■ WS-ARC EMI3	0,46625
■ non-WS ARC EMI3	0,3176

Job submission - 100 jobs in sequence



■ WS-ARC preEMI

100 jobs
50,99366667

■ WS-ARC EMI2

51,0495

■ non-WS ARC EMI2

32,5248

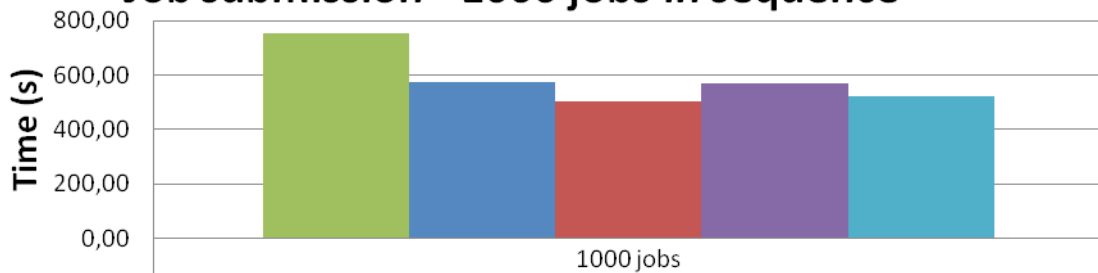
■ WS-ARC EMI3

50,47175

■ non-WS ARC EMI3

36,511

Job submission - 1000 jobs in sequence



■ WS-ARC preEMI

■ WS-ARC EMI2

■ non-WS ARC EMI2

■ WS-ARC EMI3

■ non-WS ARC EMI3

1000 jobs

753,7

574,9574

503,2788

571,7095

522,2004

- There is no brokering involved since the target is requested directly with the `-c` command line option. The time measured corresponds to the time elapsed between executing the first `arcsub` command and receiving the last grid job ID indicating successful job submissions.
- We measure the time it takes to submit (sequentially) 1, 100 and 1000 simple echo jobs directly to an A-REX service by using one `arcsub` command to submit all jobs.
- Each scenario is repeated multiple times and resulting time is average from the times obtained.

Job submission - 1 job



■ WS-ARC preEMI

0,470666667

■ WS-ARC EMI2

0,725

■ non-WS ARC EMI2

0,302

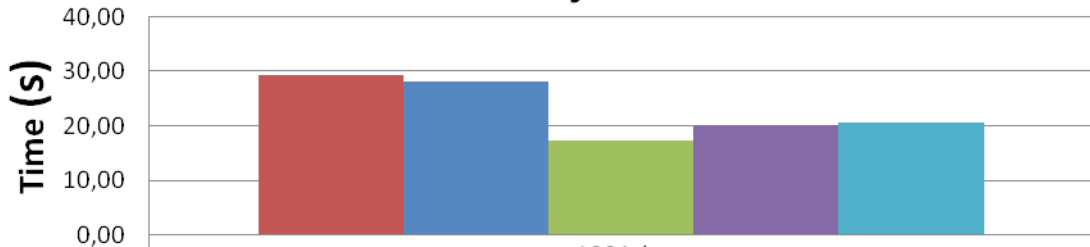
■ WS-ARC EMI3

0,46625

■ non-WS ARC EMI3

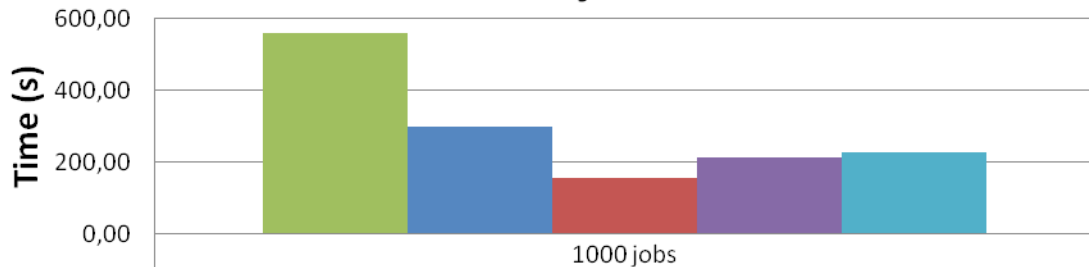
0,3176

Job submission - 100 jobs in bunch



	100 jobs
■ WS-ARC preEMI	29,31333333
■ WS-ARC EMI2	28,213
■ non-WS ARC EMI2	17,36625
■ WS-ARC EMI3	20,103
■ non-WS ARC EMI3	20,523

Job submission - 100 jobs in bunch



	1000 jobs
■ WS-ARC preEMI	557,33
■ WS-ARC EMI2	299,7
■ non-WS ARC EMI2	157,11275
■ WS-ARC EMI3	213,9562
■ non-WS ARC EMI3	227,4426667

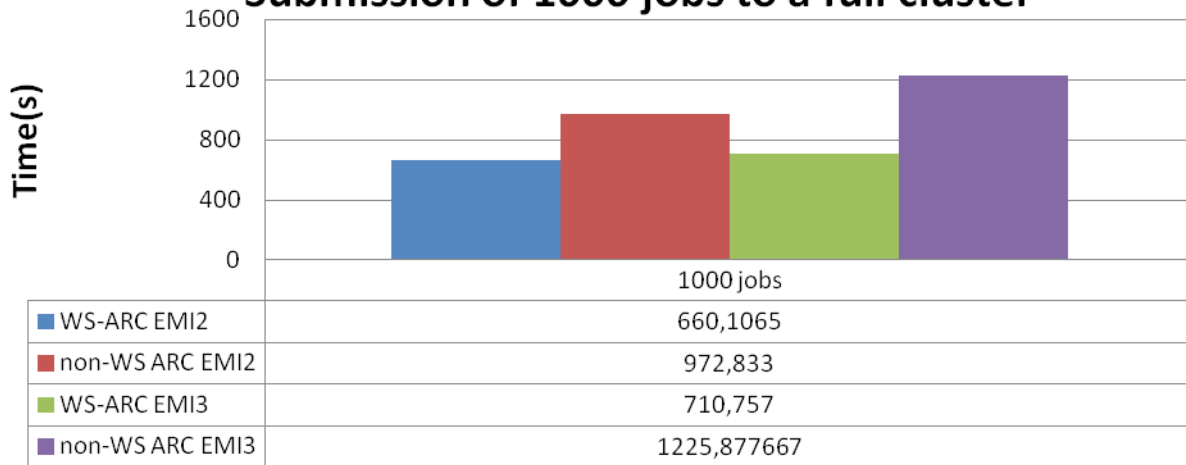
- We measure the time it takes to submit (sequentially) 1000 simple echo jobs directly to an A-REX service that is filled with 1000 finished jobs.
- There is no brokering involved since the target is requested directly with the `-c` command line option
- The time measured corresponds to the time elapsed between executing the first `arcsub` command and receiving the last grid job ID indicating successful job submissions.
- Each scenario is repeated multiple times and resulting time is average from the times obtained.

Submission of 1000 jobs to cluster with 1000 finished jobs

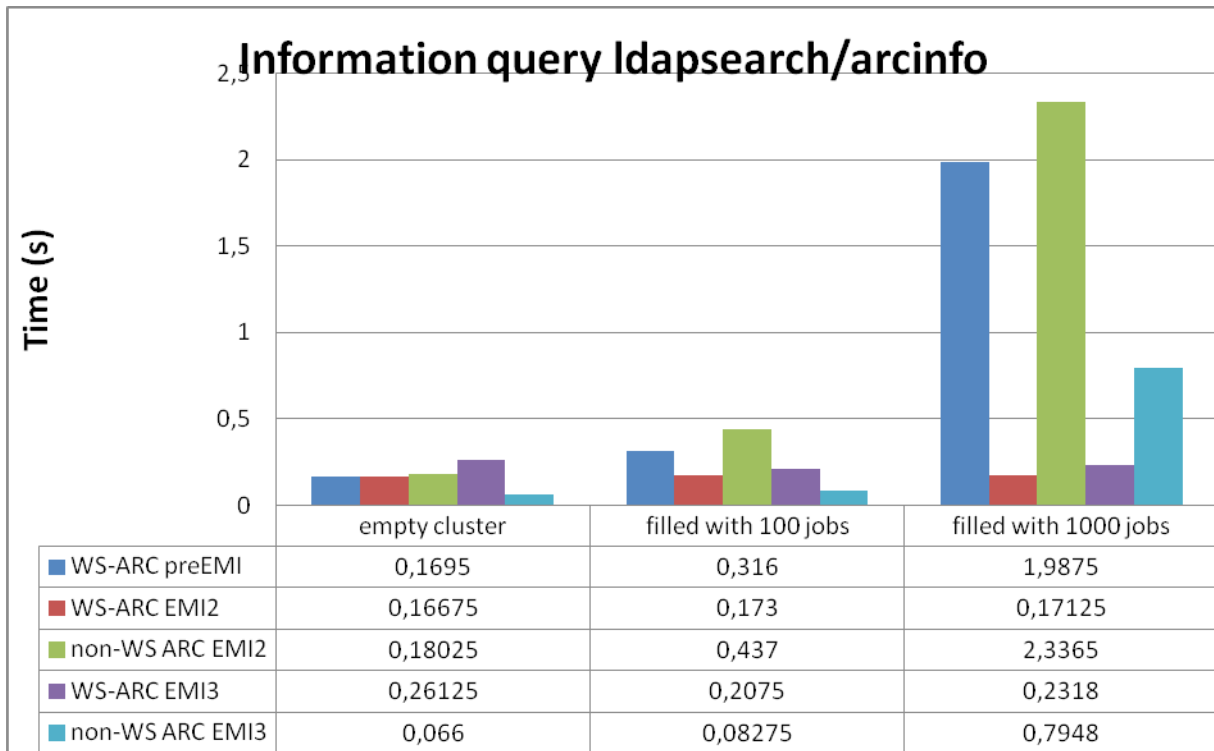


EUROPEAN MIDDLEWARE INITIATIVE

Submission of 1000 jobs to a full cluster



- We measure the time it takes to obtain the content of the complete local information tree of A-REX service (the information is not cached) when the information system is empty, contains only the minimal cluster/queue/user information; when it is filled with 100 and then 1000 finished jobs
- All queries are repeated multiple times



- We measure the time it takes to stage in 1, 10 and 30 files (each 300 MB large) from/to machine A to/from machine B
- Stage-in is performed in a way that data is directly uploaded by the client (machine B) to the A-REX's (machine A) session directory.
- The timing of data staging only includes the time spent on actual data movement (such factors as e.g. queuing and various grid layers are excluded).

Stage-in from client

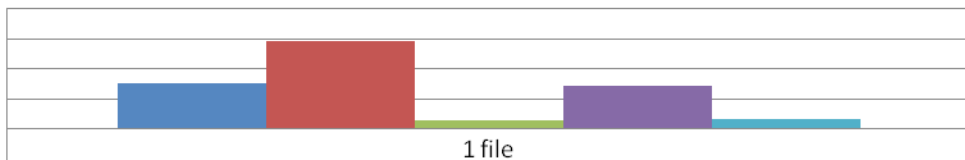


EUROPEAN MIDDLEWARE INITIATIVE

Stage-in from client - 1 file

Time (s)

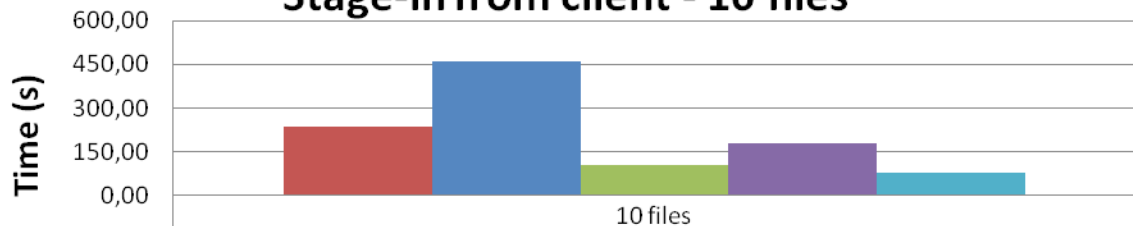
60,00
45,00
30,00
15,00
0,00



1 file

■ WS-ARC preEMI	22,92333341
■ WS-ARC EMI2	44,04646597
■ non-WS ARC EMI2	4,28326094
■ WS-ARC EMI3	21,43575
■ non-WS ARC EMI3	4,69925

Stage-in from client - 10 files



■ WS-ARC preEMI

235,66999990

■ WS-ARC EMI2

461,83715670

■ non-WS ARC EMI2

105,00217380

■ WS-ARC EMI3

178,4125

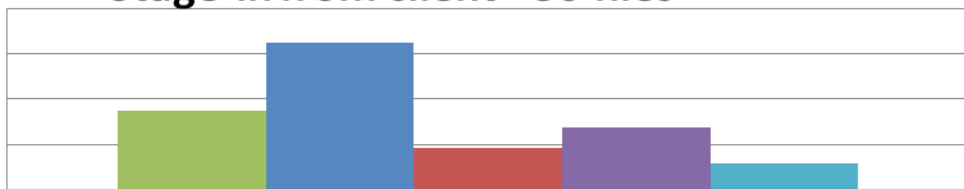
■ non-WS ARC EMI3

79,81475

Stage-in from client - 30 files

Time (s)

1600,00
1200,00
800,00
400,00
0,00



30 files

■ WS-ARC preEMI

699,61386970

■ WS-ARC EMI2

1293,74128300

■ non-WS ARC EMI2

368,52245700

■ WS-ARC EMI3

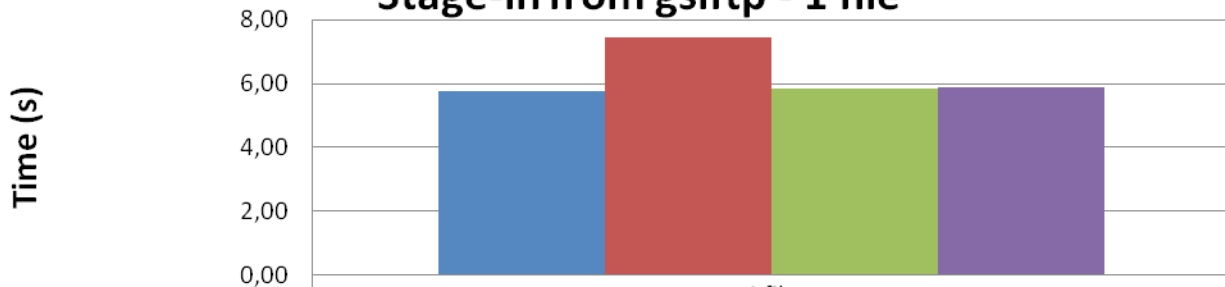
544,2135

■ non-WS ARC EMI3

228,0725

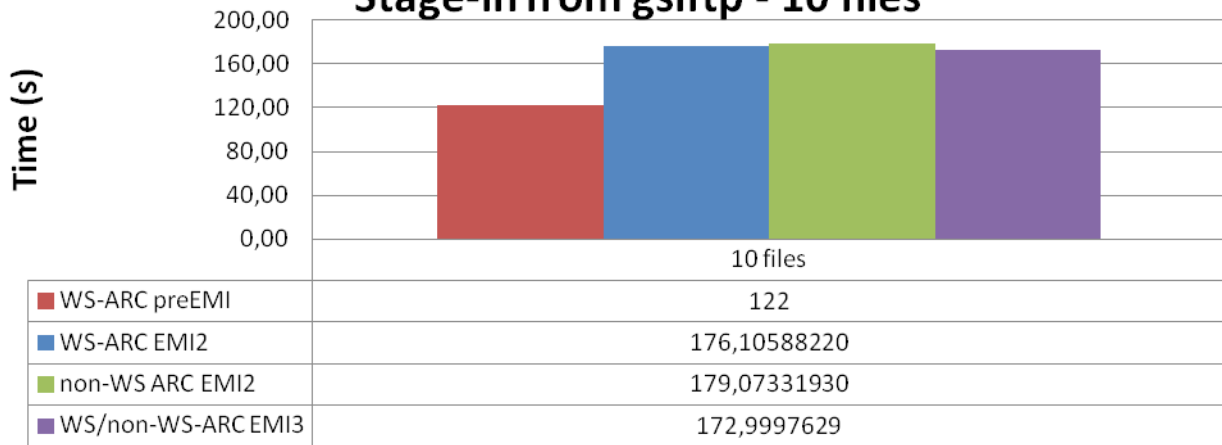
- We measure the time it takes to stage in 1, 10 and 30 files (each 300 MB large) from/to machine A to/from machine B.
- Stage-in is performed in a way that the downloader module of the A-REX service (machine A) fetches input data from the GridFTP server running on the submission machine B.
- The timing of data staging only includes the time spent on actual data movement (such factors as e.g. queuing and various grid layers are excluded).

Stage-in from gsiftp - 1 file



	1 file
■ WS-ARC preEMI	5,776666641
■ WS-ARC EMI2	7
■ non-WS ARC EMI2	6
■ WS/non-WS-ARC EMI3	5,885302633

Stage-in from gsiftp - 10 files



Stage-in from gsiftp - 30 files

Time (s)

600,00
400,00
200,00
0,00



30 files

■ WS-ARC preEMI

406,7866666

■ WS-ARC EMI2

505,87854220

■ non-WS ARC EMI2

495,72932900

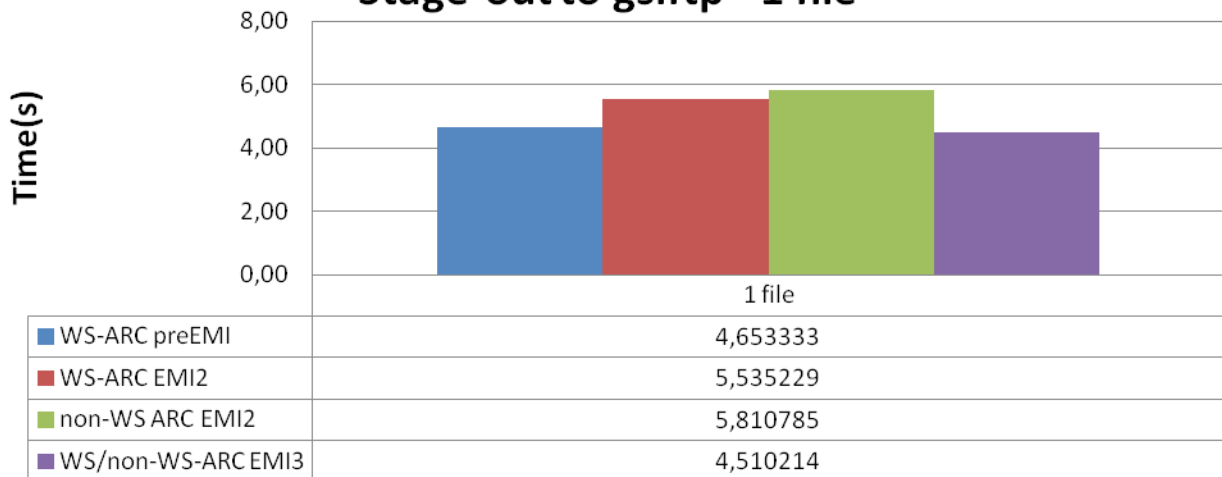
■ WS/non-WS-ARC EMI3

457,7140559

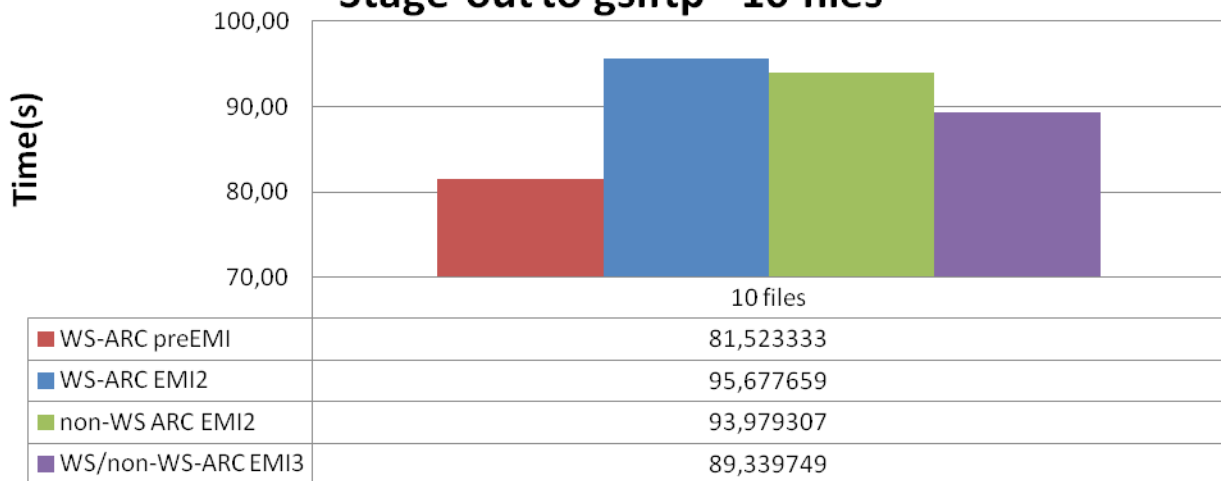
■

- We measure the time it takes to stage out 1, 10 and 30 files (each 300 MB large) from/to machine A to/from machine B.
- Stage-out is performed by the uploader module of the A-REX service (machine A) which sends data to the GridFTP server running on machine B.
- The timing of data staging only includes the time spent on actual data movement (such factors as e.g. queuing and various grid layers are excluded).

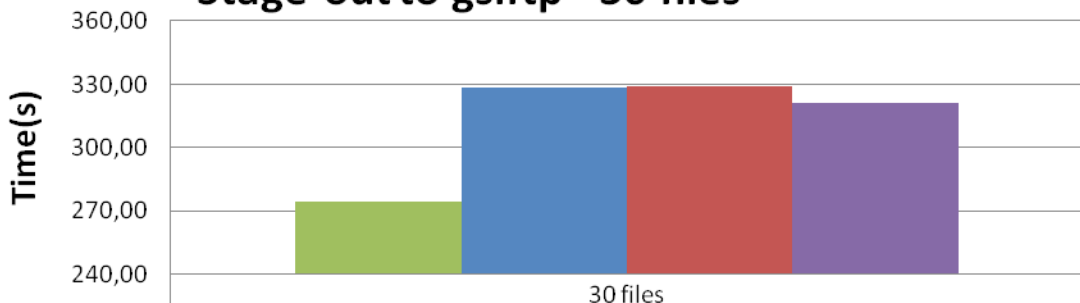
Stage-out to gsiftp - 1 file



Stage-out to gsiftp - 10 files



Stage-out to gsiftp - 30 files



■ WS-ARC preEMI

■ WS-ARC EMI2

■ non-WS ARC EMI2

■ WS/non-WS-ARC EMI3

30 files

274,730000

328,599223

328,751791

321,148129

Conclusions



- Submission of jobs to full cluster (1000 jobs) is little bit worse than for EMI 2 release
- Stage-in from client was improved for EMI-3 release
- Stage-out is worse than pre-EMI solution
- Remaining parameters are equal or better than in the case of EMI2 release
- Know issue - scalability of querying information system (a new solution exists-replacement of current method of querying local information tree with DB)
- We think that discussion about proposal of standard test cases is needed to compare distributed computing solutions (grid, clouds, etc...).

- [1] D5.4-1 RESOURCE CONSUMPTION PROFILE AND PERFORMANCE BENCHMARK STUDY OF THE EARLY PROTOTYPE RELEASE OF KNOWARC
http://www.knowarc.eu/documents/Knowarc_D5.4-1_07.pdf
- [2] D5.4-1 RESOURCE CONSUMPTION PROFILE AND PERFORMANCE BENCHMARK STUDY OF THE FINAL RELEASE OF KNOWARC,
http://www.knowarc.eu/documents/Knowarc_D5.4-1_09.pdf
- [3] http://wiki.nordugrid.org/index.php/Performance_testing
- [4] <http://www.eu-emi.eu/>
- [5] <http://node1.grid.upjs.sk/performance1.php>