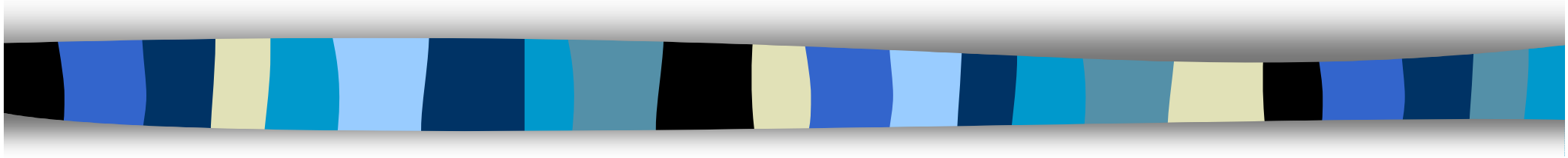
TSTG GeneralMeeting

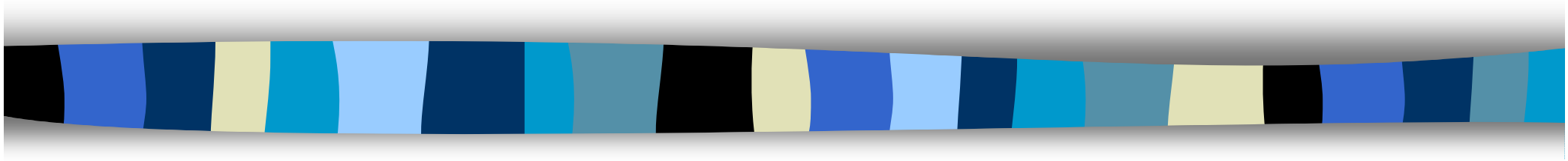


Gilbert Grosdidier

LAL-Orsay/IN2P3/CNRS

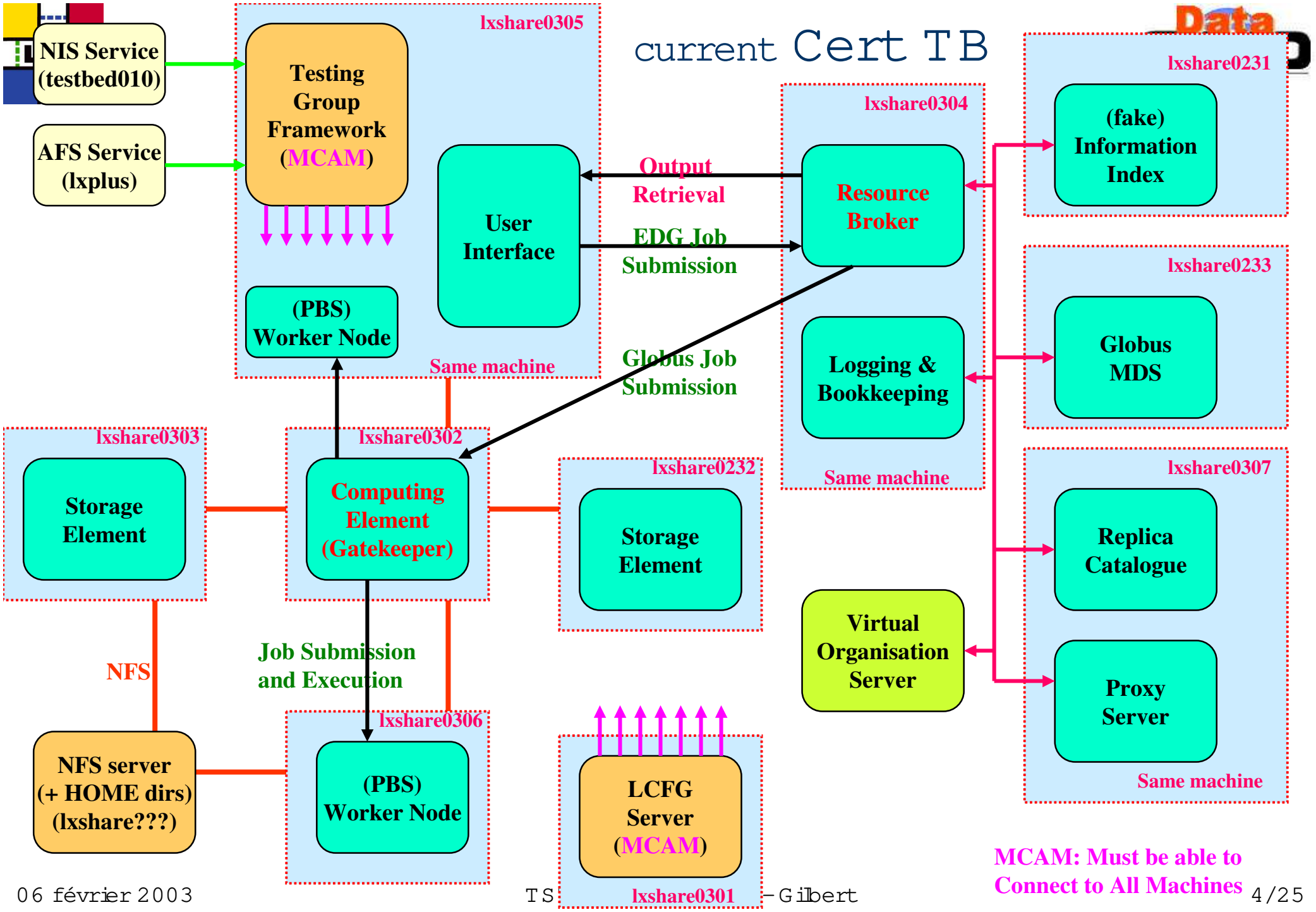
& LCG

Description of Testing Suites section



Testing Suites: Goals

- Testing Suites can be aiming at 4 kinds of use cases
  - Check that a GRID cluster is correctly installed and running
  - Check that a newly built M/W release is coherent and complete
  - Check that a Production cluster is fully operational
  - Check that a Testing Suite is adequate, and covers all bases
- The current goal for us is building/testing the Main Script
  - meaning the 4th use case
  - this Main Script will give access to all Suites thru switches
- To check our suites, we need
  - a well known/understood M/W version (currently edg-1.4.3)
  - a full GRID cluster with a controlled install
    - available at CERN
    - installed thru LCFG (reproducible)
    - UI, CE, 2xSEs, RB, LB, WN, MDS, (BDII), PX, RC
      - there are 2 SEs to allow for full data managements tests
  - this GRID cluster will evolve to something more realistic when needs arise


current Cert TB
NIS Service (testbed010)
AFS Service (lxplus)
lxshare0305
Testing Group Framework (MCAM)
(PBS) Worker Node
Same machine
User Interface
Output Retrieval
EDG Job Submission
Globus Job Submission
lxshare0304
Resource Broker
Logging & Bookkeeping
Same machine
lxshare0231
(fake) Information Index
lxshare0233
Globus MDS
lxshare0307
Replica Catalogue
Proxy Server
Same machine
Virtual Organisation Server
lxshare0303
Storage Element
lxshare0302
Computing Element (Gatekeeper)
lxshare0232
Storage Element
NFS
Job Submission and Execution
lxshare0306
NFS server (+ HOME dirs) (lxshare???)
(PBS) Worker Node
LCFG Server (MCAM)
lxshare0301
MCAM: Must be able to Connect to All Machines

Testing Suites: Structure

- These tools currently exist under 2 forms
  - GUI tool installed and running locally on each machine of the cluster
    - required by sysadmins when achieving manual or non-standards install
  - a large Main Script being run centrally from a single machine
    - covering however all the components of the cluster
    - implies some constraining requirements for accessing the remote nodes
    - allows for running in an automated framework, or thru a cron job
      - for Build checking or Production cluster checking
- They share basic scripts as much as possible
- I'll mainly cover the 2nd point (the large Main Script)
- Two languages used: **bash** and **Perl** (5.005003)
  - bash used for building basic scripts (mainly Install&Config level, HTML presenter)
  - But slowly becoming minority, because Perl offering more functionalities
    - while being a little bit more cryptic
  - Then it is strongly advised to use Perl to develop new tools
    - we will have to wrap bash scripts to plug them into OO Perl frame
  - We want however to preserve the base of already available/written tools
    - could be even advisable to re-use existing core functions to write new I&C modules

Testing Suites: Functionalities

- There are many areas to cover in these tests
  - even when running high level tests (M/W global functionality check, or cluster stress testing), one has to make sure that the lower levels are fine
- We currently distinguish these levels:
  - **Install & Configuration** level
    - checking existence of RPMs, config and log files, access priviledges, and so on
  - **Unit Testing** (and Daemon testing ?): single component functionality checks
    - ex: is the Globus gatekeeper running fine on this CE ?
  - **Global tests:** full functionality checks, involving several components, with single jobs, embedding different kinds of functionalities
    - ex: insertion of a new entry in the RC, + creation and replication of the file
    - ex: auto-coherency of the II, coherency with installed resources, is the II consistent with a smooth running of the GRID cluster ?
  - **Stress testing:** job storms and so on
    - ex: running several 100s of jobs, retrieving their outputs, checking it's OK
    - ex: same thing with copy storms
  - Stability and **Robustness** testing (standby right now)
    - is everything coming up fine and running after a reboot/shutdown/breakdown ?
    - probably aiming at Production cluster tests only
    - are **Security issues** part of this item ?

What is a Suite ?

- Proposal: a Suite is a bundle of individual tests that allow to check all the different aspects of a given component
  - meaning that, to be able to declare a component is operational, we will have to run a suite including all of I&C tests, Unit tests, Global tests and Stress tests for that specific component
  - it could obviously happen that several Suites could overlap
- What is the Main Script then ?
  - Simple answer: it should allow for running in an easy way several different Suites selected thru switches
    - the upper level switches should in turn switch on lower level tests to allow for more flexibility
  - So the Main Script will look like
    - a collection of high level Suites, if we look at switch level
    - or a collection of low level Suites, if one looks at individual component level
  - Whenever possible, building a new high level Suite should require to bundle only low level switches rather than combining high level switches
    - to avoid too much overlapping in the tests

About configuration files

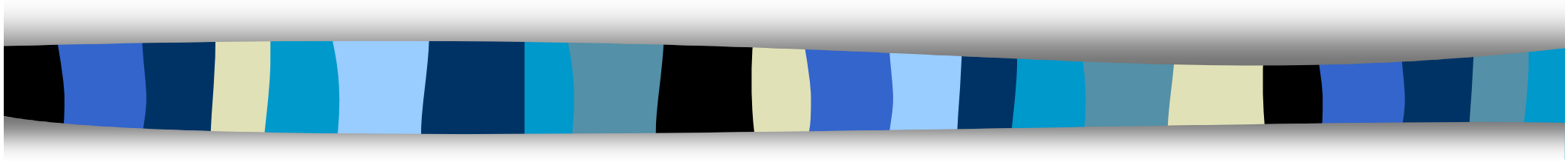
- It is foreseen to use 3 different configuration files
  - the testing framework configuration file
    - describing the path to the scripts, and other utility variables
  - the site configuration file
    - describing the cluster where the tests will be running
    - it is currently built from or extracted from the LCFG config file
  - the M/W configuration file
    - where are located the M/W version
    - containing pointers to version specific features
    - hope to have this one very small
- These config files are sourced by the Main Script upon starting
  - and every value is inherited from this
- We should stick to this model
  - and avoid having too many files to source to describe the different parts
  - no embedded sourcing: no file calling a file calling another file...
    - unless at the very beginning of the implementation process
  - clear sections into the main file are required
    - together with explanations :-)

Cert TB status (edg-1.4.3)

- Current machines
  - lxshare0301 - LCFG server
  - lxshare0302 - CE (+WN2)
  - lxshare0303 - SE1
  - lxshare0304 - RB
  - lxshare0305 - UI
  - lxshare0306 - WN1
    - PBS only
  - lxshare0307 - RC+PX
  - lxshare0231 - BDII
  - lxshare0232 - SE2
  - lxshare0233 - MDS

- Current « tstgrp » users
  - +aformic
  - +barcelo
  - +bettini
  - +cavalli
  - +dqing
  - +fchollet
  - +fede
  - +gmerino
  - +grodid
  - +legre
  - +louis
  - +mandjavi
  - +mdavid
  - +metery
  - +neyroud
  - +reale
  - +serra
  - +silvestre

Testing Suites Status section



Testing Suites: Status

- Install & Config level
  - These are mainly static tests
  - For CE, SE, RB, LB, UI, WN
    - Basics scripts: ready
    - Merging to Main Script: ready for all but SE
    - Migration towards edg-1.4.3, and testing: ready for all but SE
    - Presenter wrapping: ready for RB only
      - ideas available
  - For MDS, (BDII), PX, RC
    - nothing ready yet
    - looking for manpower
  - later: R-GMA, VO Server, RLS
    - looking for manpower as well

Testing Suites: Status (2)

- Daemon Testing
  - some of the daemon testing is currently part of previous level
    - status: fair
    - would rather have it here, but no serious harm
  - will have to integrate monitoring sensors developed elsewhere in EDG
    - no manpower assigned
- Unit Testing
  - many tools provided thru the ITeam already
    - Gatekeeper, GridFTP, II, MDS
    - status:
      - scripts under Perl OO ready
      - merged into the Main Script
      - Presenter feature not ready yet
        - » ideas exist, manpower assigned
  - most probably missing many areas, IMHO
    - will have to dig into material provided thru the WPs (WP1 Testing Plan ?)
    - will have to cross-check with input from loose cannons

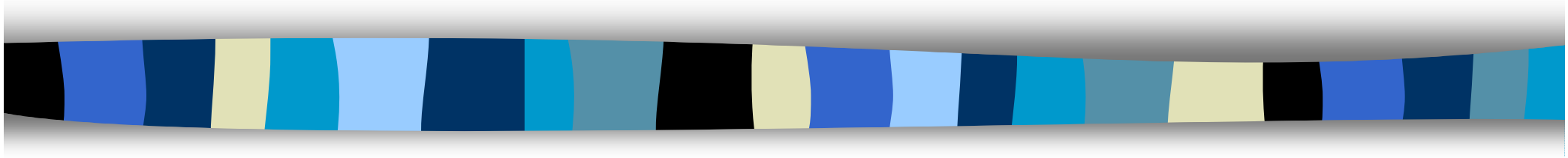
Testing Suites: Status (3)

- Global Testing
  - improved tools provided thru the IT team are available
    - status: fair, merged into the Main Script, Perl OO
    - no presenter wrapping yet
    - overall cluster testing with simple requests
      - full UI+RB+CE functionality
      - long job features
      - BrokerInfo functionality
  - II Consistency check (to allow for smooth running)
    - being developed, ideas exist, no manpower clearly assigned
- Stress Testing
  - framework for a job storm already exists
    - the script for submitting it also exists
    - easy to adapt to Perl OO
  - merged into the Main Script
  - submitting a simple JDL file
  - no presenter wrapping yet

Currently Available in the Main Script

- The Main Script is now just a little bit more than a skeleton
  - sourcing the 2 conf files (site & edg-test)
  - specifying the location of the Perl tools and libs (Cal's one)
  - calling some high level suites (mostly with no Presenter parsing)
  - neither clear documentation yet, nor options/comments
- Available Suites
  - CE: to exercise Globus Gatekeeper (currently: CE only)
  - Ses: exercising GridFTP+GridFTPUmask Cal's tests on both SE nodes
  - FTP: exercising GridFTP between all of CE, SE's, RB
  - I&C: running Security_ & RB_config_test
  - RB: running several Cal's tests
    - HelloWorld, HelloScript, "Sleep 30s", "CheckSum size=1Mb", BrokerInfo
  - MDS: running CheckConsistency & CheckGRIS Cal's tests, on BDII & MDS
  - GG: my Job Storm (mostly as a plugging exercise), running 100 requests
    - with a rather simple jdl script
  - DNS: running Cal's ReverseDNS test on all of CE, SE's, RB, PX

HTML Presenter section



Presenting the results (IFAE HTML parser)

- Choosen: the output of each test will be automatically parsed
  - they will be merged onto one or several HTML pages
  - under the form of a result matrix (or matrices)
  - several tests will be merged to allow for fast scanning
- Every entry should itself be a link
  - towards a more detailed page, allowing for in depth checking
  - titles and/or description of each test (hence its meaning) will also be available on the same page, thru links
- A (rather simple) example is available here:
  - http://grodid.home.cern.ch/grodid/test3101/

QuickTime™ et un décompresseur TIFF (LZW) sont requis pour visionner cette image.

About the PerlOO framework

- This is inherited and adapted from Cal
- Current OO methods for the BaseTest Class

• sub new	NoOver
• sub setContext	SubClass
• sub process	NoOver
• sub checkArguments	Utility
• sub setDefaults	Utility
• sub checkPrerequisites	SubClass
• sub setupTest	SubClass
• sub runTest	SubClass
• sub evaluateTest	SubClass
• sub cleanup	SubClass
• sub printHelp	NoOver
• sub printDesc	NoOver
• sub addOption	Utility
• sub setErrorMessage	Utility
• sub checkError	Utility

- Proposal: add new methods for
  - analyzing the output
  - formatting the results
    - sub buildConfFile SubClass
    - sub runParserScript NoOver
  - they willbe called from the evaluatetest or process methods

A Checking transfers to /tmp

Using prefix/value:/tmp,0.0660704630427063

- Check local—> lxshare0303.cern.ch: [OK] [OK] [OK]
- Check lxshare0303.cern.ch —> lxshare0232.cern.ch: [OK] [OK] [OK]
- Check lxshare0303.cern.ch —> lxshare0302.cern.ch: [OK] [OK] [OK]
- Check lxshare0303.cern.ch —> lxshare0304.cern.ch: [OK] [OK] [OK]

– Using prefix/value:/tmp,0.838301313109696

- Check local—> lxshare0232.cern.ch: [OK] [OK] [OK]
- Check lxshare0232.cern.ch —> lxshare0303.cern.ch: [OK] [OK] [OK]
- Check lxshare0232.cern.ch —> lxshare0302.cern.ch: [OK] [OK] [OK]
- Check lxshare0232.cern.ch —> lxshare0304.cern.ch: [OK] [OK] [OK]

– Using prefix/value:/tmp,0.649601413402706

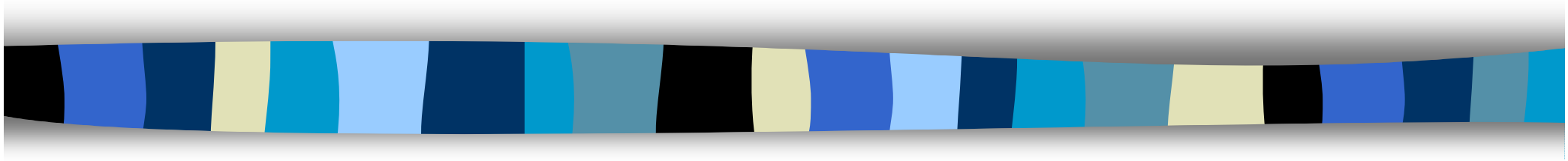
- Check local—> lxshare0302.cern.ch: [OK] [OK] [OK]
- Check lxshare0302.cern.ch —> lxshare0303.cern.ch: [OK] [OK] [OK]
- Check lxshare0302.cern.ch —> lxshare0232.cern.ch: [OK] [OK] [OK]
- Check lxshare0302.cern.ch —> lxshare0304.cern.ch: [OK] [OK] [OK]

– Using prefix/value:/tmp,0.318467741832137

- Check local—> lxshare0304.cern.ch: [OK] [OK] [OK]
- Check lxshare0304.cern.ch —> lxshare0303.cern.ch: [OK] [OK] [OK]
- Check lxshare0304.cern.ch —> lxshare0232.cern.ch: [OK] [OK] [OK]
- Check lxshare0304.cern.ch —> lxshare0302.cern.ch: [OK] [OK] [OK]

- 3 Checking transfers to ~
  - Using prefix/value:~,0.417699105571955
    - Check local—> lxshare0303.cern.ch: [OK] [OK] [OK]
    - Check lxshare0303.cern.ch —> lxshare0232.cern.ch: [OK] [OK] [OK]
    - Check lxshare0303.cern.ch —> lxshare0302.cern.ch: [OK] [OK] [OK]
    - Check lxshare0303.cern.ch —> lxshare0304.cern.ch: [OK] [OK] [OK]
  - Using prefix/value:~,0.446819024160504
    - Check local—> lxshare0232.cern.ch: [OK] [OK] [OK]
    - Check lxshare0232.cern.ch —> lxshare0303.cern.ch: [OK] [OK] [OK]
    - Check lxshare0232.cern.ch —> lxshare0302.cern.ch: [OK] [OK] [OK]
    - Check lxshare0232.cern.ch —> lxshare0304.cern.ch: [OK] [OK] [OK]
  - Using prefix/value:~,0.801964308600873
    - Check local—> lxshare0302.cern.ch: [OK] [OK] [OK]
    - Check lxshare0302.cern.ch —> lxshare0303.cern.ch: [OK] [OK] [OK]
    - Check lxshare0302.cern.ch —> lxshare0232.cern.ch: [OK] [OK] [OK]
    - Check lxshare0302.cern.ch —> lxshare0304.cern.ch: [OK] [OK] [OK]
  - Using prefix/value:~,0.546492491848767
    - Check local—> lxshare0304.cern.ch: [OK] [OK] [OK]
    - Check lxshare0304.cern.ch —> lxshare0303.cern.ch: [OK] [OK] [OK]
    - Check lxshare0304.cern.ch —> lxshare0232.cern.ch: [OK] [OK] [OK]
    - Check lxshare0304.cern.ch —> lxshare0302.cern.ch: [OK] [OK] [OK]

Manpower Issues section



Remaining Tasks & Manpower Issues

- This is a catalogue of non-assigned tasks
- Liaison man between EDG-IT team and LCG-TSTG ??
  - Should help monitoring closely and identify precisely the areas where the M/W was change dramatically between current and next versions
    - allows integrating in advance these changes in the Testing Scripts
    - and allows merging of new scripts provided by the EDG developers
    - He/She must be an experienced expert to be efficient, acting part time
- Of course, one liaison man is required between VDT and TSTG
- About the Install&Config suites now
  - Merging the Tests into the TSTG Framework RM
  - One developer to create the missing modules (MDS, PX, RC, BDII?)
    - and later R-GMA and VO Server MR+GT ??
  - One developer to help migrating existing suites from edg-1.2.2 to edg-1.4.3
    - and testing them on the Cert TB FC + MR+GT ??
  - We strongly suggested to extend the mandate of 2 INFN developers beyond the end of year 2002 to allow for achieving these tasks

Manpower Issues (2)

- One developer for gathering and implementing (if needs arise) Unit Testing (and may be Daemon Testing): this are may not be well covered, IMHO
  - cf WP1 Testing Plan to seek in details ??
- Robustness Testing development: no urgent need, but we will not drop this item ??
- Integration to TSTG Main Script and Display tools of Cal+GG+JJB GlobalFuntionality Tests & Job Storms
  - Integration GG + ??
  - Parser for the HTML Presenter GM+FC+MB
  - JJB globaltest is currently rewritten in Perl, including more functionalities, and more flexibility AS + ??
- Development of a "Copy Storm" following the same model as for the job storm above
  - reusing the frame developed by GG for storms ??

Manpower Issues (3)

- Checking Information Index coherence with requirements
  - check that the values returned for each II objectClass allow for a smooth running of the TB
    - we miss reference values for this !
    - some work already began around that, within a Perlframe ?? + GG
- Security and Control issues
  - this is rather vague right now : we probably have to find a way to check that every piece of sensitive info is accurately protected EF ??
    - proxies, certificates, access rights and so on

Expectations for VDT testing

- What to provide ?
  - (Perl) Scripts for Install&Config level, as well as for Unit Testing, ASAP
  - Later, Robustness tests
  - Global functionality tests, if they have to differ from the EDG tests for the same level
  - Care of Stress Testing should be taken with the existing framework, reusing the previous level scripts
- What will happen when the next version will be released ? As for our EDG partners:
  - the new version of the new tests scripts must be released altogether
  - or a specific advise should be sent, indicating how to update the testing scripts of the previous release