Castor Planning

Castor development team

December 5, 2006

Contents

TG1 Known bugs

T1.1 Memory leak

Description The current stager suffers from severe memory leaks. There is currently a work around restarting the stager every 3 hours to avoid crashes due to lack of memory. The leaks need to be fixed.

Priority Medium

Manpower Sebastien, 10%

Duration 3 days

Estimated start date 1^{st} November 2006

Completion 100%

Spent Manpower 0 days

Risks and alternatives The risk is to keep running a cron job restarting the stager every 3 hours. It is actually a very reasonable alternative that is used in production for several months now.

T1.2 Bad states in the DB

Description There are regularly requests in the DB that are in WAIT_SUBREQ status and are waiting on a non-existent parent. We have now identified this situation as the result of the following sequence of events : first a PrepareToPut, then a Get that is set in WAIT_SUBREQ, waiting on the PrepareToPut, then a regular Put and finally a PutDone. The PutDone does not wake up the Get request but cleans up all the *Put, leaving the Get to wait on nothing. One has thus to wake up requests pending on all the sequence of *Put when a putDone occurs.

Priority High

Manpower Sebastien, 10%

Duration 4 days

Estimated start date 10^{th} June 2006

Completion 100%

Spent Manpower 4 days

Risks and alternatives Besides the requests that get stuck, the risk is to accumulate request with a bad status in the database, which could slow down the processing in case they accumulate too much. The alternative is to run regularly cleaning scripts recognizing this situation and curing

the stuck requests. Such a script already exists and is run manually from time to time.

T1.3 Memory leak follow-up

Description Check that the stager memory leak was an ORACLE bug and is actually gone with the last ORACLE update.

Priority Medium

Manpower Sebastien, 25%

Duration 1 day

Estimated start date 15^{th} December 2006

Risks and alternatives The risk is that the problem was somewhere else and comes back. We could anyway reuse the work-around of restarting regularly the stager.

T1.4 Databases Cleanup

Description Cleanup the databases from accumulated request in bad status.

Priority Medium

Manpower Sebastien, 10%

Duration 2 days

Estimated start date 1^{st} December 2006

Completion 5%

Risks and alternatives The risk is that old problems hide some remaining small ones and prevent us from debugging them. Another problem is that accounting cannot be used until this cleanup is done since stuck request may be taken into account depending on their status.

TG2 Testing

T2.1 Test recovery after DLF restart

Description All daemons should detect that the DLF connections are screwed and should reconnect and resume logging automatically.

Priority Medium

Manpower Dennis, 40%

Duration 4 days

Estimated start date 15^{th} August 2006

Completion 100%

Risks and alternatives The risk is to not store logging information in the DLF database after a DLF crash or restart. However, this information is available locally on the different nodes running the system. Only the centrally managed DB would be lost.

T2.2 Test DB hardware configurations

Description Different hardware configurations, including RAC and Dataguard tools should be tested to understand what is the most suitable hardware for the CASTOR stager DB

Priority Medium

Manpower

Duration 5 days

Estimated start date 1^{st} quarter 2007

Risks and alternatives The risk is that the current hardware is not able to support the load that will be needed for the LHC running. The only alternative then is to split the stager DB into several distinct DBs, managing less pools. The draw back is that the transfer from pool to pool across stagers is not supported.

Depends on Requires active collaboration from the Castor Operations and the DB support team.

T2.3 New DLF testing and deployment

Description Viktor and Dennis did a great job in tuning and improving DLF so that it can handle more logs per second. Dennis also developed the partitioning of the tables. All this needs to be tested and deployed on the production setup.

Priority High

Manpower Dennis, 40%

Duration 11 days

Estimated start date 1^{st} June 2006

Completion 100%

Spent Manpower 10 days

Risks and alternatives The risk of keeping the current DLF is that it cannot handle the logs generated by CASTOR fast enough which leads to a general slow down of the whole CASTOR system (Note that this "feature" disappeared in the new version, replaced by lost log). A work around is to reduce the amount of log in the DLF DB and replace it by log in local files.

TG3 Stager

T3.1 Signal handling on the client part

Description Be able to handle properly a Ctrl-c of the client and cancel the ongoing requests on the fly in case they were not yet fully processed (typically avoid unnecessary recalls).

Priority Low

Manpower Giulia, 50%

Duration 5 days

Estimated start date 15^{th} April 2007

Risks and alternatives The only risk here is to recall files from tape (and thus mount tapes) when it was not needed. This task is actually about recall optimization.

T3.2 Improvements on the stager_qry

Description For the stager_qry on directories, try to avoid the LIKE and use start_with or equivalent so that special characters don't cause troubles. Also try to evaluate the implications in terms of scalability of the regexps support.

Priority Medium

Manpower Sebastien, 25%

Duration 3 days

Estimated start date 2^{nd} quarter 2007

Completion 50%

Spent Manpower 2 days

Risks and alternatives The risk is that heavy usage of stager_qry overloads the stager database and slows down the whole CASTOR system. This can be avoided by reducing the number of threads dealing with queries or by disabling the regexps in stager_qry. Both of these work around can be done at any time without service interruption.

T3.3 Automatic cleanup at restart

Description Study the ways to clean the DB after the abrupt stop of a component (stager, rh or other DB related components). List the cases where a request is left in a bad state that will prevent further processing and see how these states could be detected (even if some other replica of the components may still be running) and fixed.

Priority Medium

Manpower Giulia, 50%

Duration 10 days

Estimated start date 3^{rd} quarter 2007

Risks and alternatives The risk of not cleanup up the DB after a crash is that some requests that were processed at the time of the crash will never be resumed and the user will way until the client timeout occurs.

T3.4 SvcClass specific selection policy for filesystems

Description The filesystem selection policy is CASTOR 2 is unique and is applied to all svcClass and diskpools. This should be modified in favor of a svcClass specific policy that could be changed on the fly.

Priority Medium

Manpower Dennis, 45%

Duration 5 days

Estimated start date 1^{st} May 2007

Risks and alternatives The risk is to not be able to tune 2 different diskpools with 2 different filesystem selection policies. It may not be a problem if a single policy can handle all kind of diskpools but this still has to be proved.

T3.5 Recall policies

Description The recall policies were foreseen in the CASTOR 2 architecture but never implemented. They should allow to execute various actions before the recall of a file such as adding more files for recall, wait some time before launching the recall, or wait until enough data have to be recalled.

Priority Medium

Manpower Dennis, 40%

Duration 5 days

Estimated start date 8^{th} January 2007

Risks and alternatives

T3.6 Filesystem GC trigger optimization

Description After some studies from Nilo, the trigger launching the garbage collection of a filesystem could be optimized.

Priority Medium

Manpower Giuseppe

Duration 2 days

Estimated start date 10^{th} May 2006

Completion 100%

Spent Manpower 3 days

Risks and alternatives

T3.7 Handle recovery after DB restart

Description All daemons should detect that the DB connections are screwed and should reconnect before resuming activity.

Priority High

Manpower Giulia, 80%

Duration 9 days

Estimated start date 15 July 2006

Completion 100%

Spent Manpower 8 days

Risks and alternatives The risk is to have daemons with bad database connection, thus unable to process any request, leading to a complete stop of CASTOR. An alternative to fixing the problem is to monitor the daemons activities and restart the daemons if they get stuck. Currently, a lemon monitoring is in place for that, issuing an alarm when things get stuck.

T3.8 Accounting

Description Build accounting information at the request handler level and use it to avoid denial of service attacks. This involved keeping statistics of the requests per user, per type and maybe per diskpool and set limits above which we refuse more requests.

Priority Medium

Manpower Rosa, 50%

Duration 10 days

Estimated start date 1^{st} September 2007

Risks and alternatives In the current situation, some looping scripts can kill the system by submitting too many requests. The system reacts quite well at the beginning : LSF queues jobs to a certain limit. Then the remaining ones stay in the stager DB but the rmmaster tries to submit them and gets stuck due to that. An alternative is to rewrite properly rmmaster (which is scheduled also) but then the requests will accumulate so much in the stager that it will take ages to process them once the original problem is fixed in the client.

TG4 Scheduler

T4.1 LSF limitations

Description Study the limitations of LSF and its castor plugin in terms of number of jobs scheduled per second. Propose and implement improvements if some can be done. This includes rewriting the LSF plugin as well as the RmMaster daemon for the monitoring part.

Priority High

Manpower Sebastien, 25%

Duration 20 days

Estimated start date 15^{rd} August 2006

Completion 50%

Spent Manpower 11 days

Risks and alternatives In case one LSF scheduler can not cope with the request rate, the only solution will be to multiply the CASTOR instances and partition the diskpools. An alternative is to try to reduce the number of calls to the CASTOR LSF plugin by not calling it at all when the is a single candidate machine (GET on a file with a single replica).

T4.2 LSF improvements

Description Once the limitations of LSF are removed, several improvements could be done to the CASTOR plugin. Among them (but not exhaustively), one can mention :

- a better error reporting when a job can not be scheduled, giving the precise reason in the LSF log
- a maximum number of retries for scheduling a job after which the job should fail and the client be warned. The limit should be at least service class specific so that disk only service classes try less times in case of no available space
- the usage of several queues with different limits could be envisaged

Priority Medium

Manpower Sebastien, 25%

Duration 8 days

Estimated start date 1^{st} quarter 2007

Risks and alternatives This is only an optimization, no risks here. However, it could generate some temporary instability on a working solution.

T4.3 Avoid scheduling prepareTo requests

Description For historical reasons, CASTOR 2 is scheduling all requests, including prepareToGet/Put/Update ones. This scheduling is actually probably useless and consumes a lot of resources. One should investigate if it could be avoided.

Priority Medium

Manpower Felix, 80%

Duration 5 days

Estimated start date 1^{st} December 2006

Completion 40%

Spent Manpower 2 days

Risks and alternatives This is only an optimization, no risks here. However, it could generate some temporary instability on the software.

TG5 Protocols

T5.1 RFIO common project

Description Launch and participate to a common RFIO project with DPM developers. DPM people are proposing to implement a new RFIO supporting plugins. The CASTOR plugin has to be extracted of the current code. Then the rest of CASTOR has to use the new RFIO.

Priority High

Manpower Giulia, 40%; Giuseppe 20%

Duration 45 days

Estimated start date 1^{st} August 2006

Completion 75%

Spent Manpower 41 days

Risks and alternatives The only alternative is to live with 2 concurrent RFIO versions like today which is very confusing for the end user.

Depends on The implementation, packaging and deployment of the new RFIO core part by the DPM team.

T5.2 GridFTP

Description GridFTP is still not an internal protocol of CASTOR. The move to GridFTP v2 should ease the port of GridFTP to CASTOR. At the same time, gridFTP should be made available for SLC4

Priority Medium

Manpower Viktor, 90%; Rosa, 60%

Duration 40 days

Estimated start date 1^{st} September 2006

Completion 90%

Risks and alternatives The only alternative is to stay with the current GrifFTP V2. It has 2 major drawbacks : its high footprint on the diskservers and the fact that it is not integrated to CASTOR, forcing us to export the physical locations of our files.

Depends on Deployment and support of GridFTP2 by LCG.

T5.3 Xrootd

Description Xrootd should become a CASTOR internal protocol. Most of the work will be done by the xrootd developers. The task is to support them and adapt the castor job so that it supports xrootd as transfer protocol. This is also a startup task for Rosa that will learn the CASTOR internals at the same time.

Priority High

Manpower Rosa, 20%

Duration 20 days

Estimated start date 1^{st} April 2006

Completion 90%

Spent Manpower 19 days

Risks and alternatives Xrootd can also be used in the same mode as GridFTP, where it calls RFIO internally. However, we would not benefit of the advantages of xrootd in terms of fast file openings and load balancing.

Depends on The completion of the xrootd CASTOR plugin by Andy Hanushevsky.

T5.4 RFIO API thread safety and TURL parsing

Description The former RFIO API was actually completely not thread safe, especially around the passing of external parameters like the stager name, the CASTOR version or the SvcClass to be used. In order to fix that, the API has to be slightly changed and the TURL parsong mechanism has to be extended to support the new parameters.

Priority High

Manpower Giulia, 80%

Duration 17 days

Estimated start date 1^{st} August 2006

Completion 100%

Spent Manpower 17 days

Risks and alternatives The major risk was around SRM v2.2. This component is multithreaded and accesses the different CASTOR stagers and svcClass concurrently. Not having a thread safe API would have

meant that CERN needed one SRM endpoint per couple stager/SvcClass (i.e. around 30...)

TG6 Security

T6.1 Strong Authentication

Description Implement a strong authentication in all CASTOR components and study the deployment of this secured CASTOR, including the handling of credentials and authentication to ORACLE.

Priority Medium

Manpower Rosa, 30%

Duration 40 days

Estimated start date 8^{th} January 2007

Completion 5%

Risks and alternatives The current situation is that CASTOR is quite insecure, allowing "hackers" to read and delete any file quite easily. However, this has never been a problem up to now.

Depends on CSEC security module now maintained by GD.

T6.2 VOMS support

Description Add support for VOMS roles/groups based authorization.

Priority Medium

Manpower Rosa, 15%

Duration 25 days

Estimated start date 1^{st} March 2007

Risks and alternatives The VOMS authorization model has not yet fully been clarified by LCG.

TG7 Components to improve

T7.1 DLF Improvements

Description The current DLF is unable to cope with the log rate generated by a loaded CASTOR 2 instance. Moreover, the slow down of DLF slows down the whole CASTOR 2 system. Another problem is that there is currently no way to archive DLF logs. This means that logs are accumulating in the database and that they need to be purged from time to time. In order to fix the situation, several improvements to DLF should be developed :

- the database needs to implement some partitioning mechanism, so that we can drop partitions and save them into CASTOR itself
- the way messages are sent from the client to the user application should be tuned (buffering, bulk sent, ...) to be able to handle more logs per second
- the call to the logging API in a component should be decoupled from the DLF internal message sending so that the client does not suffer from DLF slow down.

Priority High

Manpower Dennis, 40%

Duration 20 days

Estimated start date 1^{st} February 2006

Completion 100%

Risks and alternatives The risk is to not be able to use the database backend of DLF for logging and to stick to local logs distributed on the different nodes that are part of the CASTOR system.

T7.2 DLF archiving GUI

Description Partitioning was introduced in DLF by Dennis, allowing to archive old logs and retrieve them from archive whenever necessary. However, no tool is provided yet to do so easily. An archiving GUI also needs to be developed to ease the management of the archived logs.

Priority Low

Manpower Dennis, 20%

Duration 10 days

Estimated start date 4^{th} quarter 2007

Risks and alternatives A cron job can easily deal with archiving of logs. In case some debug session needs to use very old data, the old archives can be retrieved manually.

Depends on The availability of the ORACLE data pump at CERN, under investigation by the ORACLE team.

T7.3 SRM 2 updates and deployment

Description After the recent discussions on the future of SRM 2 in different conferences and meetings, the first SRM 2 implementation will have to be slightly modified before its deployment.

Priority High

Manpower Shaun, 30%; Giuseppe, 4%

Duration 40 days

Estimated start date 1^{st} May 2006

Completion 80%

Risks and alternatives The risk is to continue with SRM v1. After the different modifications we were forced to do on RFIO to cope with the decision of not deploying SRM v2, this situation is workable, although we have doubts on the scalability.

Depends on SRM collaboration to finally agree on some specification.

T7.4 New VDQM

Description Matthias rewrote VDQM but did not have time to put it in production, although some tests were performed. Before a second try is done, several improvements have to be developed, especially at the level of DB tuning and dedications. Stress tests should then be performed before it finally goes to production.

Priority Medium

Manpower Giulia, 50%

Duration 30 days

Estimated start date 1^{st} quarter 2007

Completion 10%

Risks and alternatives The risk is to stay longer with the old VDQM. It will work well but lacks some new features that are long awaited.

T7.5 Further DLF improvements

Description After the successful change to the new DLF on the production nodes, a number of improvements were requested to the new tool, e.g. optimization of the DB schema to speed up queries or addition of support for strong authentication data.

Priority Medium

Manpower Dennis, 40%

Duration 10 days

Estimated start date 1^{st} July 2006

Completion 100%

Spent Manpower 15 days

Risks and alternatives The risk is to not store logging information in the DLF database after a DLF crash or restart. However, this information is available locally on the different nodes running the system. Only the centrally managed DB would be lost.

T7.6 Adaptation and enhancement of the DLF GUI

Description The DLF GUI has to take into account the specificities of the new DLF implementation. On top of that, some little improvements have been requested.

Priority High

Manpower Dennis, 40%

Duration 20 days

Estimated start date 1^{st} September 2006

Completion 100%

Spent Manpower 20 days

Risks and alternatives The risk of keeping the current DLF is that it cannot handle the logs generated by CASTOR fast enough which leads to a general slow down of the whole CASTOR system (Note that this "feature" disappeared in the new version, replaced by lost log). A work around is to reduce the amount of log in the DLF DB and replace it by log in local files.

T7.7 Resolution of the Request Mixing problem

Description Under specific circumstances (among which request cancelation by the user), CASTOR was mixing request reponses, sending the response to the wrong client. This has been analyzed to be due to the lack on request id checking on the client side.

Priority High

Manpower Giulia, 50%

Duration 9 days

Estimated start date 1^{st} September 2006

Completion 100%

Spent Manpower 9 days

Risks and alternatives some experiments (e.g.) CMS were suffering a lot from this problem, making CASTOR unusable for them.

TG8 Components to change/rewrite

T8.1 Clips to perl transition

Description The clips policy engine is heavily used in CASTOR for taking decisions. However, this engine is quite heavy and far too complex powerful for our small usage. Since it takes energy to deploy and use it, it was decided to replace it by a little perl script. This script was already developed by Jean-Damien and needs to be tested before it goes to production. Clips can then be dropped.

Priority Medium

Manpower Sebastien, 20%

Duration 2 days

Estimated start date 1^{st} March 2007

Risks and alternatives There is no risk in keeping clips. This is only a deployment issue.

T8.2 JobDispatching

Description rmmaster stands for resource monitoring master. However, this component does far more than that. In particular, it is handling the interface to the scheduling system (maui or LSF). As part of the resolution of the LSF limitations, the monitoring part will be rewritten. However, the interface to the scheduling system also needs a rewrite. This interface suffers from many deficiencies :

- lack of readability of the code
- not stateless
- lack of consistency with the information hold by the stager database

The easiest solution is probably to rewrite the component in C++ from scratch.

Priority Low

Manpower Dennis, 40%

Duration 20 days

Estimated start date 1^{st} February 2007

Risks and alternatives We can live with the current rmmaster. However, it is sometimes a quite unstable.

T8.3 CUPV

Description CUPV is an old and simple component. There would be 2 main advantages in rewriting it : getting read of old code, not known by anybody and enhancing it with VOMS support.

Priority Low

Manpower

Duration 30 days

Estimated start date 2^{nd} quarter 2007

Risks and alternatives The risk is to continue using the old component that is basically unmaintained code.

Depends on Availability and suitability of a technical student.

TG9 Tools

T9.1 Repack for CASTOR 2

Description The repack tool is using internals of the CASTOR 1 API that are no more available in CASTOR 2. This means that we need to rewrite it completely, taking the opportunity to introduce many long awaited improvements proposed by Charles and co.

Priority High

Manpower Felix, 80%

Duration 136 days

Estimated start date 1^{st} February 2006

Completion 95%

Spent Manpower 143 days

Risks and alternatives The risk is to stay with the old repack and thus to be obliged to run a CASTOR 1 instance especially for this.

T9.2 Makefiles and packaging

Description The way we currently build the software and package it suffers from many problems :

- the Imakefiles are no more readable due to accumulated complexity with no clear organization
- the libraries are badly linked, leading quite often to problems at execution time
- there is no easy way to build only the selected packages
- the compilation flags are hardcoded in many places

We have to make a survey of the available solutions to this problem and move to a completely new building scheme.

Priority High

Manpower Giuseppe, 30%

Duration 50 days

Estimated start date 15^{th} July 2006

Completion 15%

Spent Manpower 9 days

Risks and alternatives Continue with the current Makefiles is a big loss of time for developers.

T9.3 Automation of releases and tests

Description There is currently no automatic way to build a release for all supported platforms and run a test suite on each of them. Some test suites are even missing. We plan to use here the new ETICS framework.

Priority Medium

Manpower Giuseppe, 20%

Duration 20 days

Estimated start date 15^{th} April 2007

Risks and alternatives Continue with the current manual system is a big loss of time for the developers.

Depends on The ETICS framework and the experience of Quattor with it

T9.4 Recover tool for Repack 2

Description A recovery tool should be written for Repack 2 that is able to take the name of an old tape that was just repacked and recover it. This allows to recover in case the new files are not fine. It should use the backup data stored by repack in its database.

Priority High

Manpower Giulia, 40%

Duration 8 days

Estimated start date 22^{nd} November 2006

Completion 50%

Risks and alternatives The risk is to have to deal with repack problems manually. This is not really working since a tape can have up to ¿10K files on it which have to be recovered one by one.

T9.5 Support of Repack 2

Description Although repack 2 is brand new, it will be used in 2007 to repack the entire set of 9940 tapes, that is 22K tapes worth 5PB of data. We thus expect that a lot of support and many fixes will be needed from the dev team.

Priority N/A
Manpower Giulia, 8%
Duration Ongoing
Estimated start date 8^{th} January 2007
Completion N/A

T9.6 CASTOR test suite

Description CASTOR needs a regression test suite e.g. based on unit test suite in order to be able to validate new releases easily. This test suite must be documented and easily extensible so that it's completed over time
Priority High
Manpower Giulia, 50%
Duration 10 days
Estimated start date 1^{st} July 2006
Completion 100%
Spent Manpower 10 days

TG10 Platform supports

T10.1 Port of servers to 64bits

Description Only the diskserver part of CASTOR 2 was ported to 64 bits architectures up to now. We foresee that the future servers may also run on 64 bit nodes and we thus need to port the server part too. This task also includes testing the new code with different combinations of hardware, including 32 and 64 bits head nodes, talking to both 32 and 64 bits diskservers and tapeservers.

Priority Medium

Manpower Rosa, 20%

Duration 40 days

Estimated start date 1^{st} September 2006

Completion 65%

Spent Manpower 29 days

Risks and alternatives If this is not done, we'll have to continue to maintain 32 bits machines for the CASTOR central servers while the next generation of machines will be only 64 bits. Many parts of Castor-1 are essentially unmaintained which difficults porting.

Depends on Rosa coordinates the porting but has to rely on developers for doing the porting of their code. For some unmaintained modules, porters need to be identified as Rosa will not be able to port all of them.

TG11 Documentation and publication

T11.1 New CASTOR web site

Description The CASTOR web site was basically never touched since CASTOR 2 developments are started, leading to a large fraction of inaccurate information. We want to develop a new web site, based on the ELFms web site look and feel

Priority High

Manpower Rosa, 50%

Duration 10 days

Estimated start date 1^{st} May 2006

Completion 100%

Spent Manpower 5 days

Risks and alternatives The current CASTOR web site is very confusing for users due to the fact that it contains mostly out of date information.

T11.2 Maintenance of the CASTOR web site

Description FAQ, wiki pages, latest news and so on have to be updated frequently on the CASTOR web pages to avoid them to decay like the old did

Priority N/A

Manpower Rosa, 3%

Duration Ongoing

Estimated start date 1^{st} June 2006

Risks and alternatives

T11.3 Guides

Description The CASTOR 2 documentation is quite poor for the moment. All the information are somewhere on the web but it's hard to find them. A Developer Guide is still missing.

Priority Low

Manpower

Duration 15 days

Estimated start date 4^{th} quarter 2007

Risks and alternatives

T11.4 Articles

Description We have never published anything in a conference concerning CASTOR 2 so far, except for early architecture drafts. We should publish at least 2 or 3 articles on "CASTOR 2 architecture", "CASTOR 2 deployment" or "High availability in CASTOR 2"

Priority Low

Manpower Giuseppe, 25%

Duration 10 days

Estimated start date 8^{th} January 2007

Completion 0%

Risks and alternatives Lack of visibility from CASTOR on one hand and difficulties for Giuseppe to find an academic position in a university without any publication on the other.

TG12 Tape Optimizations

T12.1 Optimization of file recalls

Description The current efficiency of file recalls is extremely low in terms of tape drive usage. There are several ways to improve it. One is to optimize the order of file recalls inside a tape to minimize the seek time. Another is to use recall policies in order to not mount tapes for only few files. Yet another is to read more files than needed on the tape, while it's rotating hoping that they may be used in the future.

Priority Medium

Manpower Miroslav, 30%

Duration 20 days

Estimated start date 8^{th} January 2007

Completion 5%

Risks and alternatives There is no big risks since we are running fine with the current setup, but the expectations are quite high. We consider that we can multiply by 2 the efficiency of the recalls that account for 80% of the drive usage.

T12.2 Support of small files in CASTOR

Description Small files are not very well supported in CASTOR in the sense that they kill the efficiency of the tape drives and robots. In order to improve the situation, we propose to add a layer of indirection by which the small files would be grouped in tar files before being written to tape. This would need to adapt the nameserver code so that it understands this situation as well as the rtcopy code to deal with this new step.

Priority Low

Manpower Miroslav, 30%

Duration 30 days

Estimated start date 1^{st} May 2007

Risks and alternatives There is no bug risks since we are running fine with the current setup, but the expectations are quite high. We consider that we can multiply by 2 the efficiency of the drives.

TG13 Management and support

T13.1 Project Management

Description Team management, project architecture, planning, and coordination.
Priority N/A
Manpower Sebastien, 20%
Duration Ongoing
Estimated start date 1^{st} January 2006
Completion N/A

T13.2 Tier 1 support

Description Support for Tier 1 centers
Priority N/A
Manpower Giuseppe, 20%
Duration Ongoing
Estimated start date 1^{st} April 2006
Completion N/A

T13.3 3rd level support

Description 3rd level support
Priority N/A
Manpower Giuseppe, 11%; Sebastien, 23%; Giulia, 7%; Rosa, 15%; Felix, 4%
Duration Ongoing
Estimated start date 1^{st} April 2006
Completion N/A

T13.4 Miscellaneous Bug fixes

Description Many small bug fixes discovered either by the 3rd level support or from internal tests.
Priority N/A
Manpower Sebastien, 20%; Giuseppe, 23%; Felix, 8%

Duration Ongoing
Estimated start date 1^{st} April 2006
Completion N/A

TG14 Ongoing activities

T14.1 Castor releases

Description Build and publish CASTOR releases. This also includes hotfix scripts and writing the release notes.

Priority N/A

Manpower Giuseppe, 7%; Rosa, 13%

Duration Ongoing

Estimated start date 1^{st} January 2006

Completion N/A

T14.2 Meetings and Workshops

Description Meetings and workshop (e.g. external operation meeting)

Priority N/A

Manpower Giuseppe, 4%; Sebastien, 8%; Giulia, 3%; Rosa, 4%; Felix 2%; Dennis, 2%

Duration Ongoing

Estimated start date 1^{st} January 2006

Completion N/A