

“Xoreax” Incredibuild For Linux Release notes

version: v2.5

date: February 22nd, 2013

Author:

Dmitry Kuzminov

[illegible]

Table of Contents

Introduction.....	4
Version control.....	4
Package information.....	4
Scope.....	5
The following items were planned for this release.....	5
Additional items.....	5
Tasks and status.....	6
SRS00001 - profile.xml: Parse Incredibuild profile XML file	6
SRS00002 - XgConsole: Application should launch instance of "Build System" service.....	6
SRS00003 - XgConsole: communicate with "Build System" service using socket communication ...	7
SRS00004 - XgConsole: run first process/batch file from command line.....	7
SRS00005 - XgConsole: Intercept process creation.....	7
SRS00006 - XgConsole: Run process on remote "Helper Machine"	7
SRS00007 - XgConsole: Print stdout/stderr on screen.....	7
SRS00009 - XgConsole: tasks submitted to "Build System" service instance would run in blocked mode.....	7
SRS00010 - XgConsole: end of work.....	7
profile.xml /tools/tool/"Allow remote" should have Arguments_Exception rule with substring.....	7
test_generate_flat_dummy_project.sh – script to generate dummy Make file based project for loading and other tests.....	7
Other known issues.....	7

Introduction

The goal of this document is to provide clear status for planned work, developed features and expose issues found during internal QA tests

Version control

- All sources are located in git propriety repository (the code is well secured, please check following <https://help.github.com/articles/github-security>)
- Version control is well defined, each binary package will have version that is represent its current git tag (freeze code)

Package information

- Package created by automatic “GNU Make” system and propriety build configuration
- Package as binary code is located on https://github.com/dimakuzminov/incredibuild_deployment.git
- Package structure
 - bin/ - all Incredibuild binary applications
 - web/ - Monitor scripts, sources and libraries
 - etc/ - system scripts that are used on “Helper” and “Initiator” machines
 - tests/ - all ATP and internal test scripts
 - usr/ - system files for “Helper” and “Initiator” machines
 - Documentation/ - all relevant Documents for this project (including this document)
 - grid_server_domain.conf.[domain_group_name] this file list DNS of computers in grid domain (it is manually defined files). For more information please read “Linux_Incredibuild_deployment_Instructions_v1.0.doc”
 - linux.pem – ssh access internal key (defined by amazon site, but used in all our test systems)
 - “prepare_helper_machine.sh” - script that install all relevant packages on helper machines. For more information please read “Linux_Incredibuild_deployment_Instructions_v1.0.doc”
 - “prepare_initiator_machine.sh” - script that install all relevant packages on helper machines. For more information please read “Linux_Incredibuild_deployment_Instructions_v1.0.doc”

Scope

The following items were planned for this release

1. SRS00001 - profile.xml: Parse Incredibuild profile XML file
2. SRS00002 - XgConsole: Application should launch instance of "Build System" service
3. SRS00003 - XgConsole: communicate with "Build System" service using socket communication
4. SRS00004 - XgConsole: run first process/batch file from command line
5. SRS00005 - XgConsole: Intercept process creation
6. SRS00006 - XgConsole: Run process on remote "Helper Machine"
7. SRS00007 - XgConsole: Print stdout/stderr on screen
8. SRS00009 - XgConsole: tasks submitted to "Build System" service instance would run in blocked mode
9. SRS00010 - XgConsole: end of work

Additional items

10. profile.xml /tools/tool/"Allow remote" should have Arguments_Exception rule with substring
11. test_generate_flat_dummy_project.sh – script to generate dummy Make file based project for loading and other tests

Tasks and status

SRS00001 - profile.xml: Parse Incredibuild profile XML file

- Using license free open source library libxml
- Parsing currently tags
 - Tools
 - Tool
- Elements
 - Filename
 - AllowRemote
 - AllowIntercept
- Handle no profile.xml
 - Report error
 - End execution of XgConsole process

SRS00002 - XgConsole: Application should launch instance of "Build System" service

- GridServer initially run Instance Server only
- Each XgConsole create new Instance of GridServer that is actually execute job process
- Currently only one XgConsole instance of GridServer can run
 - The previous version would be killed
- XgConsole creates new GridServer instance as handshaking mechanism using IP socket
 - GridServer instance must be created by System (our first GridServer daemon)
 - The kill could be done by System only as well
 - Automatically, by handshaking mechanism
 - Manually, by calling sudo service incredibuild stop
- XgConsole print all information to calling shell screen
- XgConsole receives back it's GridServer instance PID (process id)

SRS00003 - XgConsole: communicate with "Build System" service using socket communication

- Implemented in SRS00002

SRS00004 - XgConsole: run first process/batch file from command line

- XgConsole should be prefixed before standard command execution. Example:
 - make modules
 - XgConsole make modules
- XgConsole executes command with handling all command STDOUT outputs
- This command executions cannot be considered as grid command, only its childes

SRS00005 - XgConsole: Intercept process creation

- libincredibuildintr.so is responsible to catch execve requests
- system currently catch execve requests to intercept process execution
 - vfork or fork requests are not used currently, since no need
- XgConsole setup environment variables to force incredibuild system traps and set configuration for process execution procedures:
 - Intercept
 - Run on remote machine
- Intercept procedure using XgSubmit for process to be run on GridServer with options
 - silent (don't print XgConsole log information)
 - block (don't release XgSubmit till process is finished on GridServer)
 - XgSubmit return code that was generated on slot (helper machine)

SRS00006 - XgConsole: Run process on remote "Helper Machine"

- Remote process return code is added to process meta data for further return to Initiator machine parent process
- Slot writes STDOUT/STDERR to log files and also sends it's back to XgSubmit (its owner)

SRS00007 - XgConsole: Print stdout/stderr on screen

- XgSubmit prints all information returned back from GridServer before process finished to run on remote machine
- XgSubmit doesn't add prefix/suffix to original process output, just push it to XgSubmit STDOUT

SRS00009 - XgConsole: tasks submitted to "Build System" service instance would run in blocked mode

- GridServer and XgSubmit support block mode in process submit
- Each submitted process has meta data, it is using active communication between calling XgSubmit and GridServer
- GridServer is running simultaneously all submitted process, but keeping socket open for each XgSubmit that is running in "block" mode

SRS00010 - XgConsole: end of work

- XgConsole properly close GridServer instance when all work is finished.
- XgConsole, XgWait and GridServer supports option "WAIT AND EXIT" for GridServer instance. It will wait till all work is done, then it will exit GridServer from inside
 - It is done in order to resolve permission issues and also make closing process properly

profile.xml - /tools/tool/"Allow remote" should have Arguments_Exception rule with substring

- Using this release for make process, it was found that same process can make compilation and linkage
 - For linkage we want our process to run locally on machine
- Add "ArgumentsExclude" element for process
 - it will check hooked process if it has such substring in its arguments, if it exists the process will run on local machine

test_generate_flat_dummy_project.sh – script to generate dummy Make file based project for loading and other tests

- This script creates shared library project
- The project has one of use-cases to create Makefile
- it is also capable to create desired number of cpp files for compilation
 - each file can have desired number of functions
- All cpp files can run in parallel using “make -j” option

known issues

- XgConsole log information is printed on main screen (shell that executes XgConsole)
- XgConsole doesn't provide smart interface for GridServer instance creation, just each run kill (if exists) previous Instance and run new
- Only one XgConsole can run in system at same time
- Currently cannot build kernel project and it is under debugging
- Cannot run XgSubmit/XgWait outside from XgConsole environment
- GridServer Slot management:
 - Helper machine is stuck on NFS client (initiator changed its IP address)
 - Sometimes not all machines are connected
 - cached ssh keys (Fixed)
 - other issues (Not reproduced and committed as bugs)
- Interceptions and submit process command line
 - need to be redesigned for dynamic memory allocation based on demand
 - issues related to command line handling are reported to log file