# 11 Offline Computing and Software

## 11.1 Introduction

This section describes the LZ offline computing systems, including offline software for the LZ experiment, the definition of the computing environment, the provision of hardware and manpower resources, and the eventual operation of the offline computing systems.

The offline computing organization provides the software framework, computing infrastructure, data-management system, and analysis software as well as the hardware and networking required for offline processing and analysis of LZ data. The system will be designed to handle the data flow starting from the raw event data files (the so-called EVT files) on the SURF surface RAID array, all the way through to the data-analysis framework for physics analyses at collaborating institutions, as illustrated in Fig. 11.1.1.

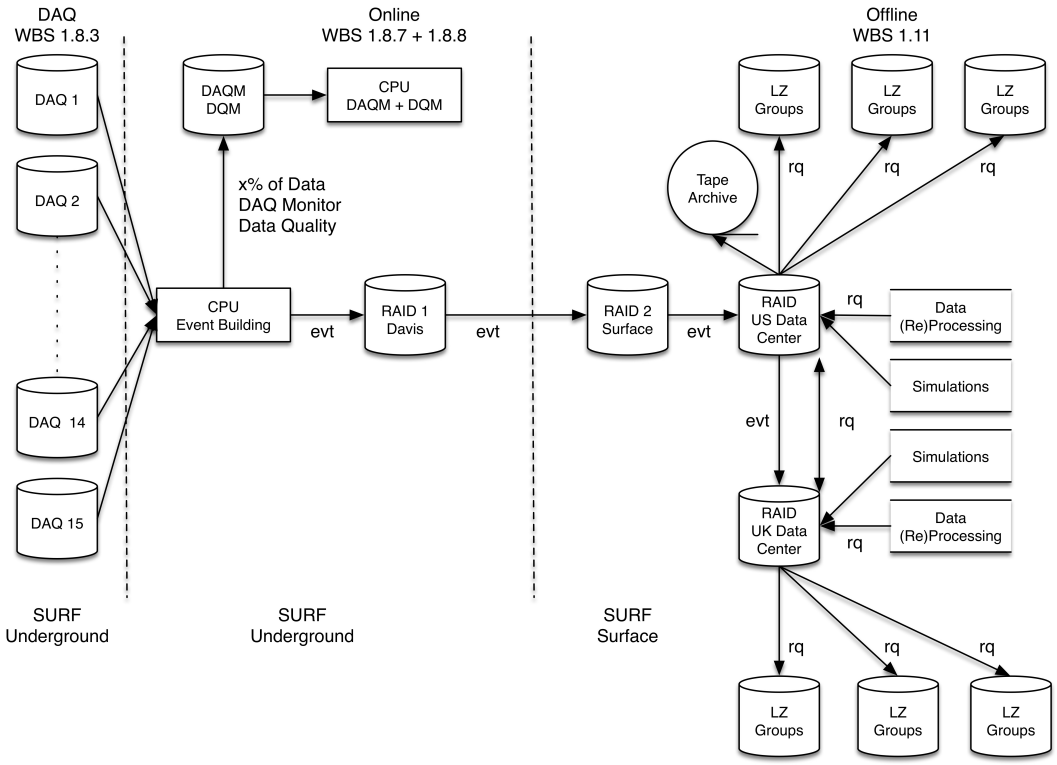


**Figure 11.1.1:** Schematic data-flow diagram for LZ.

## 11.2 Data Volume, Data Processing, and Data Centers

The LZ data will be stored, processed and distributed using two data centers, one in the U.S. and one in the U.K. Both data centers will be capable of storing, processing, simulating and analyzing the LZ data in near real-time. The SURF surface staging computer ships the raw data files (EVT files) to the U.S. data center, which is expected to have sufficient CPU resources for initial processing. The National Energy Research Scientific Computing (NERSC) center at LBNL will contain the resources to act as the LZ U.S. data center.

The run processing software (LZ analysis package, or LZap for short) extracts the PMT charge and time information from the digitized signals, applies the calibrations, looks for S1 and S2 candidate events, performs the event reconstruction, and produces the so-called reduced quantity (RQ) files. The RQ files will be accessible to all groups in the collaboration and represent the primary input for the physics analyses.

The EVT and the RQ files are also mirrored from the U.S. data center to the U.K. data center (UKDC, located at Imperial College London) partly as a backup, and partly to share the load of file access/processing, giving better use of resources for all LZ collaborators. The EVT file transfer to the U.K. data center is done from the U.S. data center as opposed to directly from SURF, in order to avoid any reduction of the bandwidth available to ship the raw data from the experiment. Subsequent reprocessing of the data (following new calibrations, reconstruction and identification algorithms, etc.) is expected to take place at one or both centers, with the newly generated RQ files copied to the other center and made available to the collaboration.

From the hardware point of view, the system must be able to deal with the LZ data volume in terms of storage capacity and processing. Based on the LUX experience and appropriate scaling for LZ (in terms of number of channels, single/dual gains, event rates, etc.), the amount of WIMP search data generated in 1.000 days of LZ running is estimated to be 2.8 PB. Including calibration runs, the total amount of LZ data produced per year is expected to be 1.3 PB to 1.4 PB, depending on the amount and type of calibration data collected during yearly operation. This estimate assumes that about three hours of calibration data are collected each week. The breakdown of the different sources of data in LUX and their scaling to LZ is given in Table 11.2.1, which clearly shows that the data volume is dominated by the S2 signals.

**Table 11.2.1:** Daily raw and compressed data rates in LZ based on scaled LUX data. The scaling factors have been computed as follows: (a) PMT surface area ratio (2) times number of channels ratio not including the low-gain channels (4), $2 \times 4 = 8$; (b) number of channels ratio (8) times rate ratio (13), $8 \times 13 = 104$; (c) liquid surface area ratio. The compression factor is taken to be 3, as described in the text. (Abbreviations: PE = photoelectron, SE = single electron.)

| Source | LUX (GB/d) | Scaling factor | LZ (GB/d) | LZ compressed (GB/d) |
|---|---|---|---|---|
| Single PE | 44.00 | $8^{(a)}$ | 352 | 117 |
| S1 | 0.24 | $104^{(b)}$ | 25 | 8 |
| S2 | 76.34 | $104^{(b)}$ | 7,939 | 2,646 |
| Uncorrelated SE | 20.00 | $9^{(c)}$ | 180 | 60 |
| Total | 140.58 | | 8,496 | 2,832 |

The SURF staging computer will have a disk capacity of 192 TB, enough storage for slightly more than two months (68 days) of LZ running in WIMP-search mode (at 2.8 TB/day), similar to its underground counterpart. The capacity of the staging arrays was based on the assumption that any network problems between SURF underground and the surface, or the surface to the outside, would take at most several weeks to be fully

resolved. The remaining storage capacity can be used to store additional calibration data. The anticipated data rates imply that the network must be able to sustain a transfer rate of about 33 MB/s. Such rates do not represent a particular challenge for the existing networks between SURF and LBNL or between LBNL and Imperial College. We note that the LUX experiment currently sends data from SURF to the primary data mirror at Brown University with an average throughput of 100 MB/s.

From the current LUX experience, we expect that processing one LZ event should take no more than one second on one core (using a conservative estimate based on an Intel Xeon ES-2670 at 2.6 GHz and 4 GB of RAM per core). Therefore, assuming a data-collection rate of 40 Hz, LZ needs 40 cores to keep up with the incoming data stream in WIMP-search mode. For reprocessing, as analysis software and/or calibrations are refined, a larger number of cores will be needed to keep the processing time within reasonable limits (e.g., a factor of 10 more CPU cores allows reprocessing of a years data in approximately one month).

Simulated data will also be created and stored. The top-level estimates in terms of storage capacity and CPU power for Monte Carlo simulations based on existing simulations are summarized in Table 11.4.1. They add up to a total data volume of about 85 to 100 TB and require approximately $10^6$ CPU hours per year.

### 11.2.1 The U.S. Data Center

The U.S. data center will be located at NERSC/LBNL. Currently NERSC has three main systems: the Parallel Distributed Systems Facility (PDSF) and two CRAY systems. PDSF provides approximately 3,200 cores running Scientific Linux and is a dedicated system for astrophysics, high-energy and nuclear physics projects. The CRAY systems, Cori Phase 1 and Edison, provide approximately 52,000 and 153,000 cores, respectively. All systems can access the Global Parallel File System (GPFS) with a current capacity of about 8 PB, which is coupled to the High Performance Storage System (HPSS) with a 240 PB tape robot archive.

The LZ resources will be incorporated within the PDSF cluster. Our planning assumes modest needs for data storage and processing power for simulations, as described in Section 11.4, a rapid growth in preparation for commissioning and first operation, and then a steady growth of resources during LZ operations. The planned evolution of data storage and processing power at the U.S. data center is given in Table 11.2.2. The amounts of raw and calibration data per year are assumed to be 1,120 TB and 320 TB, as described in the text, while the Monte Carlo data are ramped up to the maximum estimated capacity over the Project period (85 TB from Table 11.4.1, increased to 100 TB as a safety margin). Once the regular data-taking begins, the amount of simulations data is doubled in order to be able to accommodate both current and previous simulations. The processed data are assumed to be 50 % of the Monte Carlo simulations and 10 % of the data (assuming a slightly higher percentage of 10 % in the size of the RQ-files compared to the 7 % in LUX). The user data are assumed to be 50 % of the Monte Carlo simulations in the years prior to experimental data, and 5 % of the total data once LZ is running. The total disk space allocated includes a 20 % safety margin with respect to the total amount of calculated data.

The CPU power is ramped up to reach the maximum of 350 cores needed by the simulations in two years (2016 and 2017) and is increased by 40 cores to 390 cores in the commissioning year (2020) in order to be able to continue Monte Carlo production in parallel with real-time data processing (which assumes 1 s/event at 40 Hz). In the subsequent years of operation, the CPU power is increased by 440 cores per year, in order to be able to perform full data reprocessing in a reasonable time, in addition to the real-time data processing (400 + 40 cores). The CPU estimates for simulations are based on the total number of $10^6$ CPU hours from Table 11.4.1, which yields an average of about 115 cores/year in a steady state operation. Assuming that the simulations have a duty factor of about 1/6, i.e., run for 1 month and analyze/develop for 5 months, the total number of cores needed for simulations yields about 700, which is then equally divided between the U.S. and U.K. data centers.

Data flow from the surface data cache onsite to NERSC is automated by use of the Spade system. Spade will transfer raw data files from SURF to NERSC within 15 minutes of file close by the DAQ. At NERSC, data

**Table 11.2.2:** Planned storage (in TB) and processing power by U.S. fiscal year at the U.S. and U.K. data centers.

| FY | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024 | 2025 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Raw data | – | – | – | – | – | 560 | 1680 | 2800 | 3920 | 5040 | 6160 |
| Calibration data | – | – | – | – | – | 160 | 480 | 800 | 1120 | 1440 | 1760 |
| Simulation data | 40 | 80 | 80 | 100 | 100 | 200 | 200 | 200 | 200 | 200 | 200 |
| Processed data | 20 | 40 | 40 | 50 | 50 | 172 | 316 | 460 | 604 | 748 | 892 |
| User data | 20 | 40 | 40 | 50 | 50 | 55 | 134 | 213 | 292 | 371 | 451 |
| Total data | 80 | 160 | 160 | 200 | 200 | 1147 | 2810 | 4473 | 6136 | 7799 | 9463 |
| USDC: Disk space | 40 | 220 | 220 | 220 | 220 | 1360 | 3360 | 5360 | 7360 | 9360 | 11360 |
| USDC: CPU cores | – | – | 175 | 350 | 350 | 390 | 830 | 1270 | 1710 | 2150 | 2590 |
| UKDC: Disk space | 150 | 220 | 220 | 270 | 650 | 1597 | 3260 | 4923 | 6586 | 8249 | 9913 |
| UKDC: CPU cores | 150 | 175 | 350 | 350 | 350 | 390 | 830 | 1270 | 1710 | 2150 | 2590 |

will be written to NGF (NERSC Global File system) and automatically archived to HPSS (High Performance Storage System) tape. Data integrity will be verified by comparing checksums before and after transfer. From NERSC, data will automatically be transferred to the UKDC and verified by a second Spade data pipeline. Nominally (when network continuity is complete), all DAQ data files will be automatically replicated at the USDC and UKDC and backed up to HPSS within 30 minutes of DAQ file close. Although the U.S. data center will be located at NERSC, other U.S. computing resources are likely to be available to the collaboration. We will utilize resources available to the collaboration as appropriate.

### 11.2.2 The U.K. Data Center

The U.K. data center is led by Imperial College, and built on an infrastructure of GridPP [1, 2] hardware and software. It currently runs Monte Carlo production and analysis jobs, providing the data products to the collaboration via GridFTP and XROOTD. In the remainder of the construction phase, it will run the end-to-end-simulation, event-reconstruction and analysis applications described in Sections 11.4 and 11.3. The output will be background and signal models which capture the full physics of detection and measurement with the precision needed to predict dark-matter sensitivity. Tests with LUX data, system-test data, and simulation in the mock data challenges will assure that a validated pipeline is in place ready for first underground data. Timely commissioning, calibration and performance monitoring, as well as evaluation of final physics results, will all be helped by the high throughput, high availability and flexible scaling afforded by a Grid computing model.

UKDC will provide redundancy and parallel capacity for carrying out the first level, near real-time processing of all LZ raw data (when needed), and carrying out reprocessing of the entire data set on timescales of several weeks. The U.S. and U.K. data centers will use the same analysis framework, access the same central databases, and run identical software. The single, central installation is distributed via CVMFS and

the single, dedicated LZ virtual organization provides user authentication. Both are in place since June 2015, served from the University of Wisconsin.

Grid storage at UKDC was initiated in 2015 at 150 TB, to store the results of background simulations informing this Design Report. As of October 2016 the storage is 220 TB and a gradual increase is planned in 2018 and 2019 to reach 650 TB at the time of detector deployment (see Table 11.2.2). A further 1 PB to 1.5 PB per year, dependent on realized background rates and trigger strategy, will be added to maintain a complete copy of acquired LZ data.

Production using GridPP nodes at Imperial started during the simulation campaign of autumn 2015. Since December 2015, LZ has been using CPU resources at 5 GridPP sites with up to 2,000 jobs running simultaneously. This proves that the UKDC can meet its requirement of reprocessing WIMP search and calibration from scratch in one-tenth of the acquisition time. Transfer rates from UKDC to USDC were tested during the same simulation campaign. Performance using the globus data transfer toolkit with no optimization comfortably exceeded the 80 MBps required for a factor-of-two margin when mirroring acquired data.

The second large scale processing campaign was conducted in September 2016 for the validation of the background model. The simulations were performed for a modified geometry and for a larger number of detector components with increased statistics. A total of $5 \times 10^{10}$ events were generated within 411,000 CPU hours in one month. These files are using 85 TB of disk space. In addition, the UKDC has been used on request, for specific studies thorough the year using significant CPU hours (283,000) and disk storage (>50 TB). The requests are currently managed via a custom job submission system using Google sheet documents. This system will be soon superseded by a production job submission system currently in development.

The DIRAC system of middleware [3] is used to manage LZ data and computation on GridPP, providing command line, web portal, and Python API interfaces for submission, monitoring and accounting. Development has begun of tools to streamline building, editing, and merging the output of LZ jobs, using the object-oriented Python package Ganga [4] (the same approach as the LHCb collaboration, which has long experience running the Gaudi framework on the Grid). The LZ-specific aspects of UKDC are handled by a researcher at 50 % FTE and a programmer at 10 % FTE. Support of non-LZ-specific systems, which can include development of Grid infrastructure software as required, is provided by the GridPP team at Imperial.

## 11.3 Analysis Framework

The key element for the LZ data processing is the analysis framework (AF), which will contain some standard processing modules and will also allow users to put together modular code for data analysis to automatically take care of the basic data handling (I/O, event/run selection, etc.). A dedicated task force was created at the end of July 2014 to evaluate various options for LZ. In terms of existing frameworks, two `ROOT`-based frameworks were considered: Gaudi (developed at CERN and used by ATLAS, LHCb, MINERvA, Daya Bay, etc.) [5] and `art` (developed at Fermilab and used by MicroBooNE, NOvA, LBNE, DarkSide-50, etc.) [6]. In parallel, the possibility of evolving the framework developed for LUX, which is based on Python scripts and a MySQL database, and supports modules written in Python, C++/`ROOT` or MATLAB was also evaluated. For completeness, developing a new framework from scratch was also considered as another alternative. However, given the amount of effort this would have required (of the order of at least several FTE-years based on estimates from other experiments such as CMS, Double Chooz, MiniBooNE, T2K, etc.), this option was dismissed.

Input from the entire collaboration regarding the desired features for the LZ framework was collected and organized by the task force, and the three candidate frameworks were evaluated and ranked against this list. In addition, presentations and live demos for each of the three contenders were given during the regular task force meetings, while core frameworks were also installed on different test platforms to evaluate the respective installation processes.

The task force ranked unanimously Gaudi in first place, followed by a distant second by `art` and LUX in a close third position. The recommendation to adopt Gaudi as the LZ analysis framework was approved by the collaboration in April 2015. With the adoption of Gaudi as the foundation for the analysis framework, the first milestone (see Table 11.6.1) was the creation of a First Release, which was delivered to the collaboration in February 2016. This release was based on the Gaudi Hive branch of the Gaudi code, which is specially designed to support multi-threading within the framework itself. Goal of the release was to demonstrate end-to-end processing of mock (user-generated) events through the entire framework chain, and to support the development of different modules from a team of LZ collaborators.

A Physics Integration Release is currently under production and will be delivered to the collaboration in November 2016. The features of the framework needed for this release are:

- Definition of the transient data model for DAQ and Physics data;
- Creation of the necessary modules to read and write raw DAQ data based on the above transient model;
- Provision of a programming and build environment in which the physicists can adapt existing modules and develop new ones to run within the framework;
- Access to non-DAQ data, such as calibrations, slow control variables, etc., from a conditions database.

Before any of those can be achieved it is necessary to be able to build a version of Gaudi within the LZ infrastructure (see 11.5). This has been done by importing the Gaudi Hive codebase into our own GitLab in order to manage the impact of any changes made to Gaudi Hive in the future and building it from that source.

The definition of the transient data models are well underway. The DAQ one is a straightforward representation of the objects read out by the DAQ. The physics one requires more development as it has to capture the objects and relationships of derived quantities and these have not all yet been defined. The creation of the input and output modules has begun. Gaudi has a well established mechanism for extending its input and output modules to handle custom formats and we plan to use the experience of other experiments, such as Daya Bay, with this task to speed its development.

The Gaudi codebase presently supports two types of build systems, the legacy one based on CMT [7] and a new one base one CMake [8]. LZ has decided to use the CMake-based version for its build and is now using this as a model to develop the programming and build environment for developing modules for its framework. The creation of the input and output modules is being used as the test bed on which this mechanism can be developed.

Gaudi itself has a fully developed fault handling system so that any problems that arise during processing can be evaluated and either stop the processing altogether, stop the processing for a single event or allow the event to continue processing. This, together with its comprehensive logging system, means that is straight forward to decide which processings are successfully, which need to be redone once the cause of the fault has been addressed and which are irrecoverable.

Wherever possible, we anticipate that existing code from the successful LUX and ZEPLIN experiments will be adapted and optimized for use within the LZ analysis framework. The LZ processing and analysis codes will be written to be as portable as possible to ensure straightforward running on both Linux and OSX platforms for those groups who wish to do analysis in-house in addition to (or instead of) running codes on the data centers.

All non-DAQ data, i.e. any data recorded or developed that is not read out by the DAQ with each event, will be stored in a database (DB) known as the "conditions database". The challenges of this database are that it not only needs to understand the interval of validity for each piece of data, but also needs to support versioning of that data for instances such as when better calibrations are available. This problem is not unique to LZ and therefore it was decided to used the DBI package developed for the MINOS experiment. Not only

does this allow for intervals of validity and versioning, it also supports a hierarchy of data sources. This means that during development of code or calibrations it is possible to specify an alternate DB to be used for certain values which supersede the values in the main underlying DB. This also allows for the validation of new entries before they are inserted into the main conditions DB.

## 11.4 Simulations

Detailed, accurate simulations of the LZ detector response and backgrounds are necessary, both at the detector design phase and during data analysis. Current LZ simulations use the LZSim package, which in turn is based on the existing LUXSim software package [9], originally developed for the LUX experiment. The LZSim codebase is entirely separated from LUXSim, although several of the developers contribute to both. It is managed through the collaboration's git repository. This software provides object-oriented coding capability specifically tuned for noble liquid detectors, working on top of the Geant4 engine [10]. LZ intends to further update its simulation software in 2017 with two significant changes: first a switch to the most recent version of Geant4, 10.2, in order to take advantage of a number of critical improvements to the code and physics lists. Second, a recast of the LZSim framework into the more generalized BACCARAT framework, which is also based on LUXSim but not tied to any detector-specific or legacy code.

All LZ simulations will be integrated into the broader LZ analysis framework, from production to validation and analysis. The framework will in fact largely be developed using simulation output until detector data is available. Two output formats are supported, a raw simulation output at the GEANT interaction level, and a reduced tree format at the event level. Both use the `ROOT` format.

The simulations group is organized into several distinct areas of technical expertise, a structure reflected in the organization of this task. In addition to the computing-centric approach described here, physics output coordination is managed within a working group of the entire scientific collaboration. Tasks include:

(a) Simulation software packages maintenance, development, and collaboration support;

(b) Definition, maintenance, and implementation of an accurate detector geometry;

(c) Maintenance and continued improvement of the micro-physics model of particle interactions in liquid xenon, as captured in the NEST package [11];

(d) Detector response implementation which transforms the ensemble of individual GEANT4 photon hits at the PMTs to produce an event file of the same format and structure as in the data.

(e) Generators for relevant event sources in LZ for both backgrounds and signal;

Table 11.4.1 shows the computing needs estimates for all simulation tasks, based on recent production data. These needs are on the order of about 100 TB of disk storage and a total of $10^6$ CPU hours per year, mostly concentrated in short 1-to-4-week periods of burst activity. Both hard disk storage and CPU needs are dominated by background simulations, including a 50 % contingency to account for the need to repeat and/or compare some studies. There is also a provision for very-high statistics data sets for three major detector components, in order to produce a high-granularity map of the background spatial profile inside the TPC. These data sets are only kept in their most reduced format in order to conserve storage resources. For all other simulations, both reduced files and raw GEANT4 output files (translated to `ROOT` format) are kept. The total also includes optical simulations, which only provide photon collection results and are less demanding in terms of resources, calibration simulations, necessary to validate calibration data and early R&D efforts, and a number of ad-hoc small simulation tasks all grouped together. Resource time profiles for different simulation categories feed into the higher level computing estimates.

**Table 11.4.1:** High-level summary table of LZ simulations projections for computing power and storage needs, based on the simulation campaign of autumn 2015. The dominant part of the resources is required for detailed background simulations of detector components, including 50 % contingency for updates and comparative studies. Storage needs are contained within a 100 TB envelope for the duration of the project through commissioning (another 100 TB is allocated for processed and user data). Corresponding computing power is relatively modest as a yearly average, however most of the needs occur in short bursts with a 1-4 week timescale.

| Simulation type | Raw ROOT files (TB) | Reduced ROOT files (TB) | CPU needs (core-hours) |
|---|---|---|---|
| Background simulations | 30 | 2 | $1.0 \times 10^5$ |
| High-statistics background map | 0 | 17 | $8.0 \times 10^5$ |
| Contingency / repeats | 15 | 2 | $5.0 \times 10^4$ |
| Optical simulations | 2 | 0 | $5.0 \times 10^3$ |
| Calibration simulations | 10 | 1 | $3.3 \times 10^4$ |
| Other misc. simulations | 5 | 0.5 | $1.7 \times 10^4$ |
| Totals | 62 | 23 | $1.0 \times 10^6$ |

## 11.5 Software Infrastructure

All LZ software is centrally maintained through a software repository based on GitLab [12], which is currently operating at the University of Alabama. The repository is backed up daily and the snapshots are retained for 15 days. GitLab implements excellent tools for release management and code review. A GitLab snapshot from a recent developement cycle of LZSim is shown in Figure 11.5.1: different developers were working simultaneously on a round of updates to the detector geometry. After extensive testing, the updates were eventually merged into the master branch and folded into a tagged release. GitLab also offers a continuous integration tool, allowing for automatic testing and installation of the offline codebase on the U.S. and U.K. data center servers. Build automation is inherited from the Gaudi infrastructure and supported via CMake and cmt.

Release Management and Version Control standards were strictly enforced from a very early stage of the project to ensure sharing, verifiability and reproducibility of the results. Each code release undergoes a battery of tests before being deployed to production. For every update cycle, the code is inspected by a moderator, who checks its integrity and reports possible inconsistencies or conflicts. The package is then installed and executed on each data center and checked for functionality. If one test fails, the code update is rejected. Release management also acts as a bridge between development and production: this ensures that all the changes are properly communicated and documented, to achieve full reproducibility.

A comprehensive validation suite is being developed, and each class of software update has a well-defined set of tests, depending on the nature of the change and on its magnitude. For example, when the detector geometry in LZSim is updated, the standard overlap checks and geantino tests provided by Geant4 are always performed. If the update leads to a tagged release, a large statistics of photons is simulated in every part of the detector and the light collection carefully examined (light collection is the parameter most sensitive to geometry changes). Based on the successful experience from the Fermi-LAT software validation suite [13],

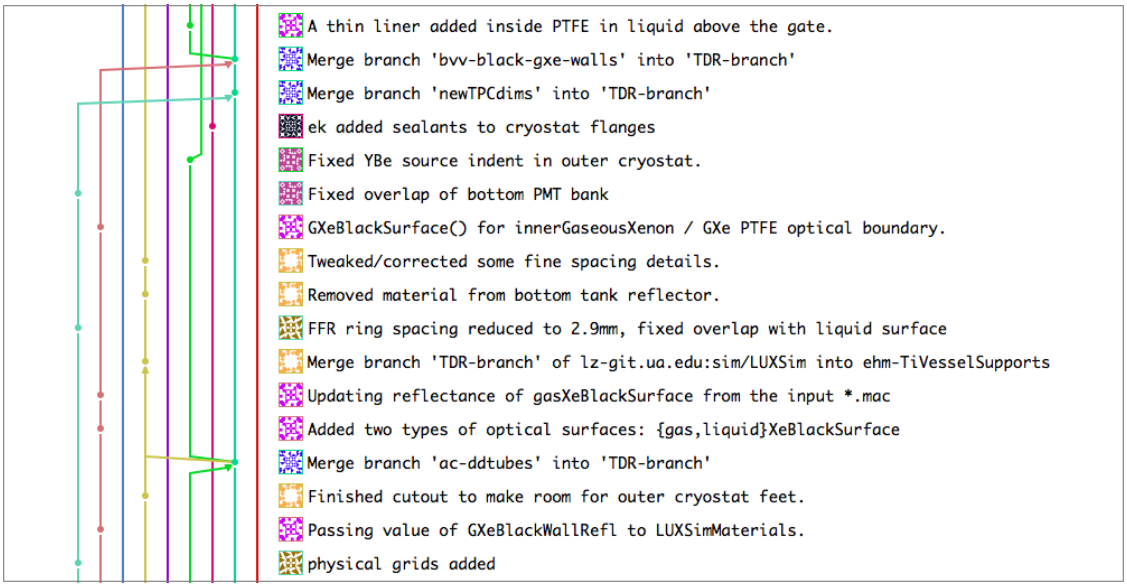


**Figure 11.5.1:** Parallel development workflow in GitLab: several contributors were updating the LZSim detector geometry simultaneously. Each vertical line corresponds to a different development branch. After extensive testing, every update was eventually merged to the master branch.

the results from these simulations are compared via a mono-dimensional KolmogorovSmirnov test. If discrepancies are found, they are reported back to the developers, who are tasked with explaining the changes or fixing the underlying errors before the code is deployed to production.

Software distribution is achieved via CernVM File System (CVMFS) [14]. CVMFS is a CERN-developed network file system based on HTTP and optimized to deliver experiment software in a fast, scalable, and reliable way. Files and file metadata are aggressively cached and downloaded on demand. Thereby the CernVM-FS decouples the life cycle management of the application software releases from the operating system. The LZ CVMFS server is hosted at University of Wisconsin, Madison and is visible to all the machines in the U.S. and U.K. data centers, and to most computing centers available to the collaboration (Wisconsin, SLAC, Edinburgh, Sheffield, etc.). It can be loaded to each collaborator's personal laptop by installing a FUSE client. All the LZ software releases and external packages are currently delivered via CVMFS: this ensures a unified data production and analysis stream, because the data centers access exactly the same versions of the same executables, removing possible dependencies on platform and configuration.

CERN software is also made available to the collaboration via CVMFS. Besides the above-mentioned Geant4 and Gaudi, we use several external packages, including CLHEP, ROOT and AIDA. The environment defining a specific combination of external packages is inherited via LCGCMT [15] and shares a build automation infrastructure with Gaudi. Again, delivering the external packages via CVMFS removes unwanted dependencies on architecture and environment. However, in order to maximize software availability to each collaborator, we also plan to deliver precompiled tarballs of LZ tagged releases and external packages for individual download.

Cybersecurity risks posed to the offline computing systems relate to the experiments data and information systems. Much of the LZ computing and data will be housed at major computing facilities in the United States (NERSC/LBNL) and U.K. (Imperial College), which have excellent cybersecurity experience and records. Specific risks posed to the LZ project relate to data transfer (in terms of data loss or corruption during transfer) and malicious code insertion. File checksums will mitigate the danger of loss or corruption of data

during transfer, while copies at both the U.S. and U.K. data centers provide added redundancy. Moreover, CVMFS features robust error handling and secure access over untrusted networks [16]. By requiring digital signatures and secure hashes on all distributed data, CVMFS also provides a strong security mechanisms for data integrity. Malicious code insertion can be mitigated by monitoring each commit to the code repository by the offline group and requiring username/password authentication unique to each contributor to the code repository. A comprehensive policy for release management and continuous testing will be the key factor in preventing malicious software from being deployed to production.

## 11.6 Schedule and Organization

Offline software by its nature is heavily front-loaded in the schedule. To enable the scientists to commission the LZ detector, the software for reading, assembling, transferring, and processing the data must be in place before detector installation. This implies, in particular, that the data transfer, offline framework, and analysis tools themselves will have been developed, tested, debugged, and deployed to the collaboration. We rely on the collaborations existing experience with the LUX experiment and others (Daya Bay, Double Chooz, Fermi, LAT, etc.), which routinely handled similar challenges. Key offline computing milestones are summarized in Table 11.6.1.

The decision on the choice of analysis framework for LZ has been taken in April 2015. The first version of the framework with a minimal number of modules was released in early March 2016. This will be followed by the first physics integration release planned for November 2016. This version includes all necessary modules for real-time processing (i.e., hit-finding algorithms, calibration constants modules, S1/S2 identification, event reconstruction), as well as a fully integrated simulations package (i.e., from event generation through photon hits, digitization, trigger, and data-format output).

**Table 11.6.1:** Key offline computing milestones.

| Date | Milestone | Status |
|---|---|---|
| Mar. 2015 | Analysis Framework decision | Done (Apr. 2015) |
| Feb. 2016 | First Analysis Framework release | Done (Mar. 2016) |
| Nov. 2016 | First physics integration release | Done (Nov. 2016) |
| Sep. 2017 | First mock data challenge | |
| Feb. 2018 | OCS FDR - Final Design Review | |
| Aug. 2018 | Second mock data challenge | |
| Nov. 2019 | Third mock data challenge | |
| Dec. 2019 | Full software release for commissioning | |

The first mock data challenge (September 2017) will test both the data flow (transfers, processing, distribution, and logging), as well as the full physics analysis functionality of the framework, separately. The first few weeks of LZ commissioning (calibrations included) will be simulated; participants should be able to quantify detector response and main backgrounds, based on simulated data. The second data challenge (July 2018) will be dedicated to testing the entire data chain. It will contain 6 months of simulated data with physics signals, including calibration data. Participants should be able to establish a detailed background

model and perform low-energy calibrations (ER/NR). The third data challenge (October 2019) will test the complete analysis strategy and is expected to validate the readiness of the offline system just before the LZ cool-down phase. It will include 1,000 days of simulated data including physics signals. No MC truth will be available to participants and physics signals will be known only to a subset of organizers. It will simulate the analysis for the first LZ science paper, including possible blinding/salting plans.

Offline computing will be led by physicists experienced in software development and use and a computing professional from LBNL. The software professional will also liaise with NERSC for collaboration on providing LZ compute resources in particular, provisioning and/or allocating of network, CPU, disk, and tape resources sufficient for LZ collaborators to transfer, manage, archive, and analyze all data for the experiment. The infrastructure software effort will also involve professional software engineering from LBNL. This person will provide technical leadership, oversight, and coordination of LZ collaboration efforts on infrastructure software as well as the design, implementation, testing, and deployment of critical LZ infrastructure components. LZ infrastructure software includes data management and processing, offline systems and monitoring, offline interfaces to LZ databases, and the analysis framework. The remainder, and bulk, of the software is a collaboration responsibility. Software for simulation, analysis, monitoring, and other tasks will be written and maintained by collaboration scientists.

## 11.7 Bibliography


[1] P. J. W. Faulkner *et al.* (GridPP), J. Phys. G: Nucl. Part. Phys. **32**, N1 (2006).

[2] D. Britton *et al.*, Philos. Trans. R. Soc. London, Ser. A **367**, 2447 (2009).

[3] A. Tsaregorodtsev *et al.*, Journal of Physics: Conference Series **119**, 062048 (2008).

[4] J. Mocicki *et al.*, Comput. Phys. Commun. **180**, 2303 (2009).

[5] G. Barrand *et al.*, Comput. Phys. Commun. **140**, 45 (2001).

[6] C. Green, J. Kowalkowski, M. Paterno, M. Fischler, L. Garren, and Q. Lu, *Proceedings, 19th International Conference on Computing in High Energy and Nuclear Physics (CHEP 2012)*, J. Phys. Conf. Ser. **396**, 022020 (2012).

[7] C. Arnault and V. Garonne, "Configuration and Management Tool," (2015), accessed: 2015-12-15.

[8] "CMake," (2015), accessed: 2015-12-15.

[9] D. S. Akerib *et al.* (LUX), Nucl. Instrum. Meth. **A675**, 63 (2012), arXiv:1111.2074 [physics.data-an].

[10] S. Agostinelli *et al.* (GEANT4), Nucl. Instrum. Meth. **A506**, 250 (2003).

[11] M. Szydagis, A. Fyhrie, D. Thorngren, and M. Tripathi (NEST), *Proceedings, LIght Detection In Noble Elements (LIDINE2013)*, J. Instrum. **8**, C10003 (2013), arXiv:1307.6601 [physics.ins-det].

[12] "Gitlab Inc." (2015), accessed: 2015-12-07.

[13] W. B. Atwood *et al.* (Fermi-LAT), Astrophys. J. **697**, 1071 (2009), arXiv:0902.1089 [astro-ph.IM].

[14] J. Blomer, C. Aguado Sanchez, P. Buncic, and A. Harutyunyan, *Proceedings, 18th International Conference on Computing in High Energuy and Nuclear Physics (CHEP 2010)*, J. Phys.: Conf. Ser. **331**, 042003 (2011).

[15] "LCG Software Elements," (2015), accessed: 2015-12-10.

[16] D. Dykstra and J. Blomer, *Proceedings, 20th International Conference on Computing in High Energy and Nuclear Physics (CHEP 2013)*, J. Phys. Conf. Ser. **513**, 042015 (2014).