

A Multi-Level Parallel Pipeline for SPHERE-3 Detector Simulation: From EAS Generation to Image Approximation

V. A. Ivanov^{1,2*}, V. I. Galkin^{1,2**}, E. A. Bonvech^{1***}, O. V. Cherkasova^{1,3},
D. V. Chernov¹, T. A. Kolodkin^{1,2}, N. O. Ovcharenko^{1,2},
D. A. Podgrudkov^{1,2}, T. M. Roganova¹, and M. D. Ziva^{1,4}

(Submitted by E. E. Tyrtysnikov)

¹*Lomonosov Moscow State University, Skobeltsyn Institute of Nuclear Physics, Moscow, 119234 Russia*

²*Lomonosov Moscow State University, Faculty of Physics, Moscow, 119234 Russia*

³*Lomonosov Moscow State University, Department of Cosmic Research, Moscow, 119234 Russia*

⁴*Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics,
Moscow, 119234 Russia*

Received April 30, 2026; revised June 23, 2026; accepted June 26, 2026

Abstract—Optimization of the SPHERE-3 detector configuration, designed to study the mass composition of primary cosmic rays in the energy range 1–1000 PeV by registering Cherenkov light reflected from the snow surface, requires simulation of a large number of extensive air shower events. A software suite with a multi-step computational pipeline is presented: shower generation in CORSIKA, decoding and cloning of events (C++/OpenMP), ray-tracing of optical photons through the detector model (Geant4 MT), and approximation of images by a lateral distribution function (Python/multiprocessing, iminuit). The key property of the problem is its natural atomicity: each event is processed independently at all stages, which provides near-linear scaling confirmed empirically on up to 32 cores. Thread safety is achieved by architectural means—shared data are read-only, mutable state is isolated per-worker—without the use of locks on hot paths.

2010 Mathematics Subject Classification: 85A99

DOI: 10.1134/S1995080226619326

Keywords and phrases: *cosmic rays, extensive air showers, Cherenkov light, Monte Carlo simulation, Geant4, SPHERE-3*

1. INTRODUCTION

The problem of detecting primary cosmic radiation (PCR) has been actively addressed for many decades. Recent results [1] indicate that the majority of primary cosmic ray (PCR) events with energies of 1–1000 PeV may be of extragalactic origin. Thus, the composition of PCR in this energy range may be of crucial importance for constructing a model of the transition from galactic to extragalactic cosmic rays. Understanding the physics of this process is essential for describing the acceleration and propagation of cosmic rays.

The detection method based on registering Cherenkov light (CL) from extensive air showers (EAS) reflected off the snow surface was proposed by A.E. Chudakov [2] and became the basis for the series of SPHERE experiments. The results obtained with the SPHERE-2 detector [3] confirmed the promise of the method and provided the foundation for developing the next-generation detector—SPHERE-3 [4]. The SPHERE-3 reflected CL telescope will feature a substantially improved light-collecting power

*E-mail: ivanov.va18@physics.msu.ru

**E-mail: v_i_galkin@mail.ru

***E-mail: bonvech@yandex.ru

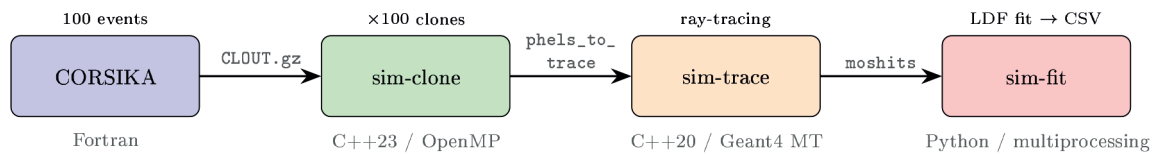


Fig. 1. Computational pipeline for the SPHERE-3 detector simulation. Intermediate data formats and parallelism technologies are indicated for each stage.

(effective entrance window area of at least 1 m^2) and an optical resolution of about 2000 pixels owing to the use of a modified Schmidt optical system with a SiPM photodetector mosaic [7].

Another novelty of the SPHERE-3 setup is the abandoning of the balloon carrier in favor of the UAV, thus, one can observe the upper hemisphere. Namely, a detector of the angular distribution of direct CL will be used that can help to accurately estimate the shower direction and contribute much to the primary nucleus mass definition.

Optimization of the detector configuration and development of criteria for shower classification by primary particle mass require accumulation of large statistics of simulated events. The full parameter space of the simulation includes 5 types of primary nuclei, 5 energy values, 5 zenith angles, several EAS models, and atmospheric models. Taking into account the cloning procedure — augmentation of statistics by parallel translation of the shower axis—and subsequent photon tracing, the total number of processed events reaches the order of 10^6 . The key property of the problem is its *natural atomicity*: each event at all processing stages is independent of the others, which allows efficient parallelization of computations without loss of physical meaning.

In this work, a software suite for the full simulation of the SPHERE-3 reflected CL telescope is presented (Fig. 1). The pipeline consists of four stages: generation of EAS events in CORSIKA [5] (Section 2), decoding of binary data and cloning of events using C++/OpenMP (Section 3), ray-tracing of optical photons through the geometric model of the detector based on Geant4 [6] (Section 4), and approximation of the registered images by a lateral distribution function using Python/multiprocessing (Section 5). In Section 6, the parallelism architecture of the entire suite is discussed. As the SPHERE-3 setup also includes a detector aimed at the direct CL angular images, Section 7 contains a few lines about its capabilities.

2. EAS EVENT GENERATION IN CORSIKA

The first stage of the computational pipeline (Fig. 1) is the simulation of extensive air showers in CORSIKA [5]. For each set of physical parameters (primary particle type, energy, arrival angles, EAS model, and atmospheric model), a base sample of 100 events is generated, which is sufficient for analyzing the main fluctuations of longitudinal shower development. CORSIKA takes into account the spectral sensitivity of photosensors and light absorption in the atmosphere, which makes it possible to optimize the calculations and track only those photons that can be registered by the detectors. Instead of using the standard CORSIKA output files, which possess excessive detail and large size, the data are stored as multidimensional arrays corresponding to the required level of detail.

2.1. Simulation Parameters

The simulation is performed on discrete sets of primary particle parameter shown in Table 1. Azimuthal angles are distributed randomly (uniformly) in the range $0 - 360^\circ$.

The full parameter space comprises: 5 energies $\times$ 5 zenith angles $\times$ 5 nucleus types $\times N_{\text{atm}}$ atmospheric models $\times N_{\text{mod}}$ EAS models. Taking into account the subsequent cloning (Section 3), each base event produces up to 100 images on the detector entrance window, yielding samples of the order of 10^4 events per grid point. The grid parameters can be adjusted and extended in the course of the work.

Table 1. CORSIKA simulation parameters

Parameter	Values
Primary energy	1, 3, 10, 30, 100 PeV
Zenith angle	5°, 10°, 15°, 20°, 25°
Nucleus type	H, He, N, S, Fe
EAS model	QGSJET01 [9], QGSJETII-04 [10]
Atmospheric model	3 CORSIKA models (MODATM 1,3,4): U.S. standard (Linsley); AT223, AT511 (Central European, 1993)
Base sample	100 events per set of parameters

2.2. Output Data Format

For each event, the spatio-temporal distribution of Cherenkov light at the snow surface level is stored. The data are represented as a three-dimensional array on a $3.2 \times 3.2 \text{ km}^2$ grid divided into 1280×1280 cells of size $2.5 \times 2.5 \text{ m}^2$. The time pulse of light in each cell is recorded as an array of 102 elements with a 5 ns step (100 data layers and 2 service layers). The data are stored in Fortran unformatted sequential access binary format (`CLOUT_*.gz`), compressed with gzip. The size of a single array before compression is

$$V = 1280 \times 1280 \times 102 \times 4 \text{ bytes} \approx 670 \text{ MB}.$$

In addition to the distribution on the snow surface, for prospective analysis of direct Cherenkov light, photon distributions at three observation levels ($h = 0.5, 1.0$, and 2.0 km above the snow) are stored in arrays of dimension $40 \times 40 \times 50 \times 50 \times 30$. The first two dimensions are spatial bins covering a $400 \times 400 \text{ m}^2$ area at each level; the next two are angular bins of the photon arrival direction (zenith θ_{ph} and azimuth φ_{ph} , spanning $50^\circ \times 50^\circ$ around the zenith); the last is the arrival time delay relative to the first photon reaching that level (0–60 ns in 30 bins of 2 ns). Joint registration of reflected and direct light makes it possible to substantially improve the accuracy of cosmic ray mass composition estimation.

Each parameter set is simulated independently, which allows running CORSIKA in parallel without inter-process coordination.

3. EAS EVENT DECODING AND CLONING (SIM-CLONE)

The second stage of the pipeline converts CORSIKA output data into a set of photoelectron files on the detector entrance window. This task is performed by the `sim-clone` application, implemented in C++23 with OpenMP for parallel computation.

3.1. Physical Meaning of Cloning

A single EAS event produces different Cherenkov light images on the detector entrance window depending on the relative position of the shower axis and the optical axis of the telescope. To increase the statistical sample without re-running the computationally expensive CORSIKA simulation, a cloning procedure is applied (Fig. 2): the original Cherenkov light distribution on the snow is translated by a random vector $(x_{\text{sh}}, y_{\text{sh}})$, generated uniformly within a circle of radius r_{clone} :

$$x_{\text{sh}} = r_{\text{clone}} \sqrt{\xi_1} \cos(2\pi\xi_2), \quad y_{\text{sh}} = r_{\text{clone}} \sqrt{\xi_1} \sin(2\pi\xi_2),$$

where $\xi_1, \xi_2 \sim U(0, 1)$ are uniformly distributed random numbers. The cloning radius is $r_{\text{clone}} = 500 \text{ m}$ for detector elevation heights $h \geq 600 \text{ m}$ and $r_{\text{clone}} = 300 \text{ m}$ for $h < 600 \text{ m}$.

Up to 100 clones are generated from each base event, yielding 10^4 images on the entrance window for a typical sample of 100 CORSIKA showers. Cloning does not produce statistically independent events in the strict sense, since clones share the underlying longitudinal shower development; however, the procedure preserves the statistical properties of the sample to the extent required for fluctuation analysis of the lateral image. Photons falling outside the detector field of view ($\sqrt{x^2 + y^2} > r_{\text{max}}$, where $r_{\text{max}} = 400 \text{ m}$ for $h \geq 600 \text{ m}$) are discarded at this stage.

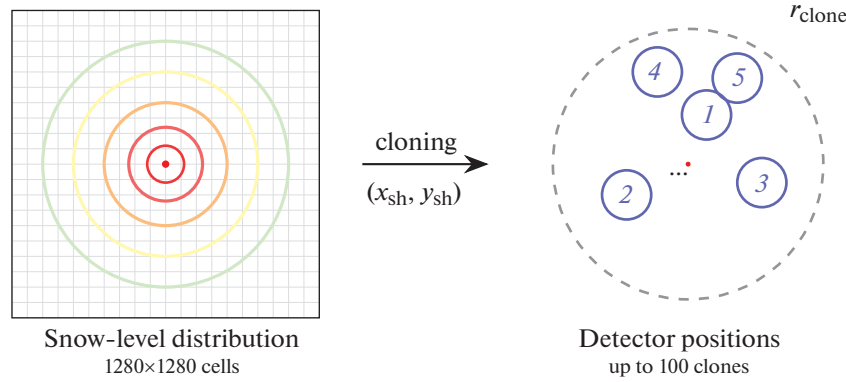


Fig. 2. Cloning scheme: a single Cherenkov light distribution on the snow surface (left) generates multiple images on the detector entrance window at various offsets (x_{sh}, y_{sh}) within a circle of radius r_{clone} (right).

3.2. CORSIKA Data Processing

The CORSIKA output files are read by a streaming gzip decoder (zlib) without prior decompression to disk. After loading the array, an index of nonzero elements is built in parallel; the number of such elements is substantially smaller than the full array size of $1280^2 \times 102 \approx 1.67 \times 10^8$. This allows processing only significant cells during photon generation.

3.3. Photoelectron Generation

For each nonzero cell (i, j, k) , the coordinates on the snow are computed as

$$x_{ij} = 2.5 \left(i - \frac{N_x}{2} - 0.5 \right) + x_{sh}, \quad y_{ij} = 2.5 \left(j - \frac{N_y}{2} - 0.5 \right) + y_{sh},$$

after which the distance from the detector axis is determined as $\rho = \sqrt{h^2 + x_{ij}^2 + y_{ij}^2}$. The observation angle $\alpha = \arccos(h/\rho)$ determines the angular efficiency coefficient, approximated by a third-degree polynomial [16]:

$$\eta(\alpha) = \frac{(a_0 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3) \cdot \cos^2 \alpha}{2.8167},$$

where the coefficients are $a_0 = 0.9952$, $a_1 = -3.571 \times 10^{-4}$, $a_2 = -1.309 \times 10^{-5}$, and $a_3 = -3.571 \times 10^{-7}$ (angle α in degrees)¹. The number of photoelectrons is generated according to the Poisson distribution

$$n_{ph} \sim \text{Pois} \left(\frac{k_{ph} \cdot 1.9 \cdot I_{ijk} \cdot \eta(\alpha)}{\rho^2} \right),$$

where $k_{ph} = 1.36848$ is the quantum efficiency coefficient and I_{ijk} is the intensity value in the CORSIKA array cell.

For each generated photon, the coordinates and time are subjected to uniform smearing within the cell: $\xi, \eta \sim U(-1.25, 1.25)$ m in space, $\tau \sim U(-2.5, 2.5)$ ns in time. The total photon arrival time is computed taking into account the propagation delay and shower inclination:

$$t = t_k + \tau + t_{big} + \frac{\rho}{c} + \frac{(x_i \cos \varphi + y_j \sin \varphi) \sin \theta}{c},$$

where $t_k = 5 \text{ ns} \cdot k$ is the time of the k th CORSIKA time bin, $t_{big} = (H_{\max} - 455)/(c \cos \theta)$ is the base delay determined by the shower maximum height $H_{\max}$, and θ and φ are the zenith and azimuthal angles of the primary particle.

¹The coefficients are reproduced from [16]; the higher-order terms contribute at a level below the Poisson photoelectron noise but are retained for reproducibility with the original parameterization.

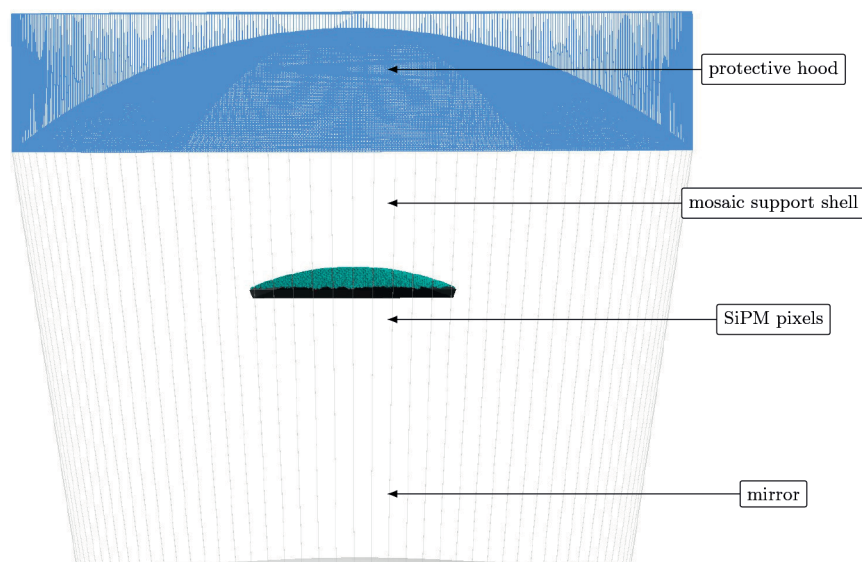


Fig. 3. Geant4 geometric model of the SPHERE-3 detector. Labeled elements (as discussed in §4.1): the spherical mirror, the mosaic support shell, the 2653-pixel SiPM mosaic, and the protective hood.

3.4. Parallelism and Data Format

Parallelism is implemented using OpenMP at the level of processing nonzero array cells within a single clone, with dynamic scheduling for load balancing. The atomic data unit of the pipeline is a single text file `phels_to_trace`, corresponding to one clone of one shower (up to 100 clones from each CLOUT file). Each file contains a header with shower parameters and records of the form (i, j, k, m, x, y, t) —cell indices, photon number, coordinates, and arrival time on the detector entrance window. Thread safety details are discussed in Section 6.

4. RAY-TRACING IN THE DETECTOR MODEL (SIM-TRACE)

The third stage of the pipeline is Monte Carlo ray-tracing of optical photons through the geometric model of the SPHERE-3 detector. The application is implemented in C++20 based on the Geant4 framework [6] using the built-in Geant4 MT multithreading support.

4.1. Detector Geometry Model

The detector model (Fig. 3) is placed inside a cylindrical world volume (radius 2 m, half-height 4 m, filled with air at $n = 1.00029$) and includes four main elements.

The primary optical element—the mirror—is imported from an STL mesh file using the CADMesh library [8]. A tessellated representation allows an accurate description of the aspherical surface without analytical parameterization. The mirror material is aluminum with specular reflection ($R = 88 - 85\%$ in the 1.5 – 2.0 eV range, modeled as a dielectric–metal boundary).

The mosaic support structure is a spherical shell ($r_{\text{in}} = 83.25$ cm, $r_{\text{out}} = 85.83$ cm, angular coverage $0 - 21^\circ$) made of aluminum with full absorption.

The photodetector array comprises 2653 pixels grouped in sets of 7 into 379 segments, with positions and orientations specified by a configuration file. Each pixel is modeled as a composite volume: the collector combines a hexagonal acrylic prism (radius 6 mm, height 4 mm) with a spherical drop lens (sphere radius 17 mm, $n = 1.5122$); the photocathode is a thin parallelepiped ($3 \times 3 \times 0.2$ mm) at the base of the collector.

A protective hood consisting of an aluminum ring and a conical wall connects to the mosaic; both surfaces act as full absorbers.

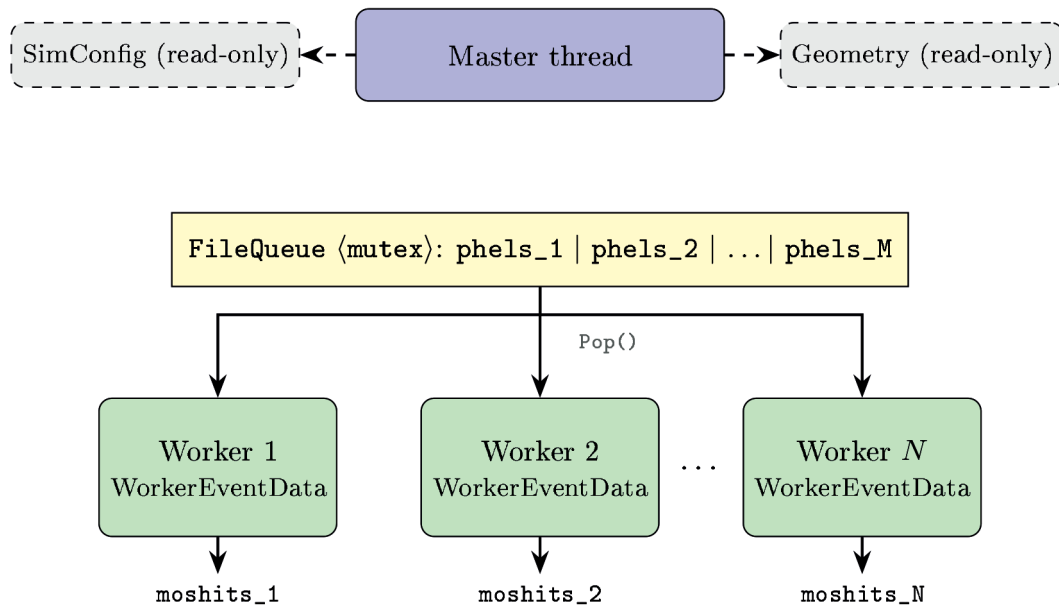


Fig. 4. Multithreading architecture of sim-trace. The master thread creates shared read-only geometry; worker threads dequeue files from the thread-safe **FileQueue** and write results to isolated output streams.

4.2. Ray-Tracing Physics

The list of physical processes is limited to optics: boundary interaction, bulk absorption, and Rayleigh scattering. All non-optical secondary particles are killed immediately.

Each primary photon is created with energy $E = 2$ eV and random linear polarization. The entry point on the detector entrance window is generated uniformly within a circle of radius $R = 85$ cm:

$$r = 85 \text{ cm} \cdot \sqrt{\xi_1}, \quad \varphi = 2\pi\xi_2,$$

after which the coordinates are rotated into the detector coordinate system. The photon momentum direction is computed as the unit vector from the source point on the snow to the entry point.

When a photon crosses the photocathode volume, the cosine of the incidence angle is computed as

$$\cos \xi = -(\mathbf{d} \cdot \hat{\mathbf{n}}),$$

where $\mathbf{d}$ is the momentum direction and $\hat{\mathbf{n}}$ is the pixel unit normal. If $\cos \xi > 0$, the angular sensitivity is

$$S(\xi) = (\cos \xi)^{p_1}, \quad p_1 = 1.093,$$

and the photon is registered stochastically with probability $S(\xi)$. In addition, at each detection event a spurious crosstalk hit is independently generated for each of the 6 neighboring pixels in the segment with probability $P_{\text{xt}} = 0.07$.

4.3. Multithreading Architecture

Multithreading is implemented using Geant4 MT (Fig. 4) following the “one event = one file” model: each worker thread dequeues a **phels_to_trace** filename from the thread-safe **FileQueue** (the only synchronization point), reads all photons as primary vertices, traces them through the geometry, and writes the detection results to its own output file.

The architecture separates data into three categories: immutable configuration **SimConfig** (orientation angles, paths), shared geometry and physics tables (read-only, guaranteed by Geant4 MT), and isolated per-worker state **WorkerEventData** (photon metadata, output stream, counters). The output text files **moshits/moshits_*** contain, for each registered photon: segment and pixel numbers, coordinates and time at the photocathode, momentum direction, origin code, and source metadata. Thread safety details are discussed in Section 6.

Table 2. LDF model parameters and Minuit bounds

Parameter	Description	Bounds
p_0	Core amplitude	$(0, 2I_{\max})$
p_1	Core linear coefficient (mm^{-1})	$(-1, 1)$
p_2	Core quadratic coefficient (mm^{-2})	$(-0.1, 0.1)$
p_3	Core cubic coefficient (mm^{-3})	$(-10^{-3}, 10^{-3})$
p_4	Tail amplitude	$(0, 10^6)$
p_5	Tail linear coefficient (mm^{-1})	$(-1, 3)$
p_6	Tail quadratic coefficient (mm^{-2})	$(-0.1, 0.1)$
R_{ch}	Crossover radius (mm)	$(10, 400)$
s_w	Sigmoid transition width (mm)	$(1, 100)$
x_0	Shower-center x coordinate (mm)	$(-300, 300)$
y_0	Shower-center y coordinate (mm)	$(-300, 300)$

Table 3. Parallelism models used at each pipeline stage

Stage	Technology	Level	Granularity	Shared state
CORSIKA	External script	Processes (OS)	One run	None
sim-clone	C++23/OpenMP	Threads (SHM)	Array cells	CL array (read-only)
sim-trace	C++20/Geant4 MT	Threads (Geant4)	One file/event	Geometry (read-only)
sim-fit	Python/multiprocessing	Processes	One file	Coordinates (read-only)

5. SHOWER IMAGE APPROXIMATION (SIM-FIT)

The final stage of the pipeline is the approximation of registered Cherenkov light images by a lateral distribution function (LDF) and the extraction of physical shower parameters. The application is implemented in Python 3.14 using `multiprocessing` for parallelization and the `iminuit` library [11] for nonlinear optimization.

5.1. Physical Model

The lateral distribution function (LDF) reduces a full mosaic image (2653 pixels) to a small set of physically interpretable parameters—total intensity, core and tail shape, transition radius, and shower-center coordinates—which serve as low-dimensional descriptors used downstream for energy reconstruction and primary mass-composition classification. The radial distribution of Cherenkov light intensity on the detector mosaic is described by a two-component LDF model with a sigmoid crossover:

$$F(r) = \omega_1(r) \frac{p_0}{1 + p_1 r + p_2 r^2 + p_3 r^3} + \omega_2(r) \frac{p_4}{1 + p_5 r + p_6 r^2},$$

$$\omega_1(r) = \frac{1}{1 + \exp((r - R_{\text{ch}})/s_w)}, \quad \omega_2(r) = 1 - \omega_1(r).$$

where $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ is the distance from the shower axis in focal plane coordinates. The model parameters and their bounds are listed in Table 2.

The detector pixel coordinates are converted from hardware (x, y, z) to a focal plane projection

$$x' = \frac{x}{z} f, \quad y' = \frac{y}{z} f,$$

Table 4. Thread-safety mechanisms in the three pipeline applications

Application	Mutexes	Read-only shared data	Per-worker mutable state	Aggregation
sim-clone	0	CL array	Per-thread RNG, output buffer	Reduction
sim-trace	1	Geometry, physics tables	Per-worker event data	Accumulables
sim-fit	0	Pixel coordinates	Full process state	Async collection

Table 5. Speedup S and parallel efficiency $\eta = S/N$ for the three stages at selected thread counts N

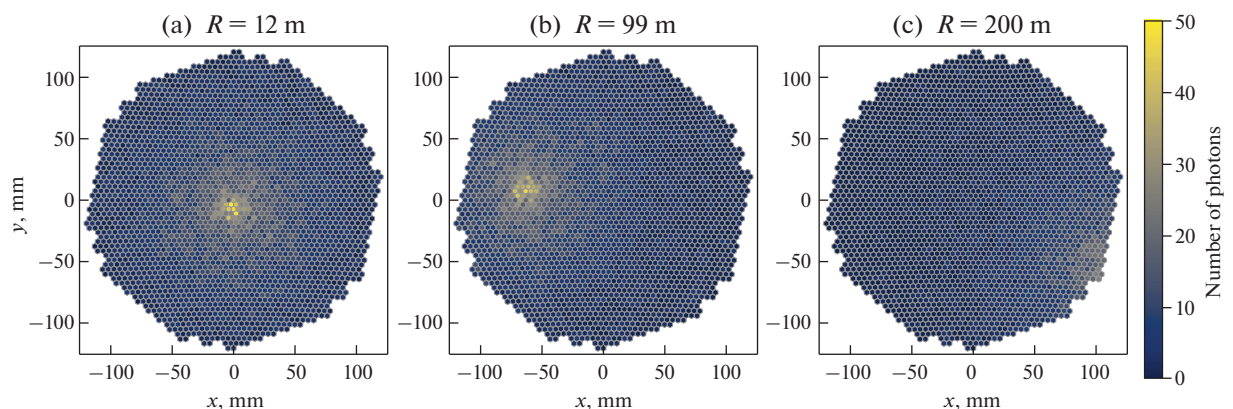
N	S_{clone}	S_{trace}	S_{fit}	η_{clone}	η_{trace}	η_{fit}
1	1.0	1.0	1.0	1.00	1.00	1.00
4	3.7	2.5	3.1	0.92	0.63	0.79
8	5.8	3.7	3.9	0.73	0.46	0.48
16	8.4	4.8	4.0	0.52	0.30	0.25
24	9.1	4.8	3.5	0.38	0.20	0.15
32	8.9	4.7	3.1	0.28	0.15	0.10

where $f = 330$ mm is the focal length of the optical system. Before approximation, the median background level b , estimated from a sample of background events, is subtracted element-wise from the data: $I'_k = \max(I_k - b, 0)$. Typical Cherenkov light images on the mosaic are shown in Fig. 5.

5.2. Optimization Strategy

Fitting is performed by the `iminuit` library (a wrapper over CERN Minuit2 [12]). Initial parameter values are determined adaptively from the radial profile. The core amplitude is initialized as $p_0^{(0)} = I_{\text{max}}$, the crossover radius $R_{\text{ch}}^{(0)}$ is estimated from the intensity-weighted mean radius, the core and tail coefficients are derived from the half-maximum radius and tail width, and $s_w^{(0)} = 8$ mm. The shower center coordinates are initialized as the mean coordinates of the $N_{\text{top}} = 10$ brightest pixels:

$$x_0^{(0)} = \frac{1}{N_{\text{top}}} \sum_{k=1}^{N_{\text{top}}} x_k, \quad y_0^{(0)} = \frac{1}{N_{\text{top}}} \sum_{k=1}^{N_{\text{top}}} y_k,$$

**Fig. 5.** Cherenkov light images on the SPHERE-3 detector mosaic for proton showers (10 PeV) at different distances R from the detector axis. The color scale indicates the number of registered photons per pixel.

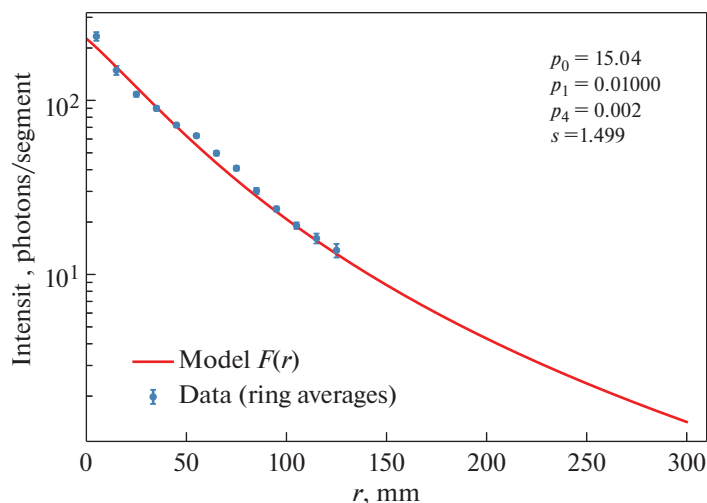


Fig. 6. Example LDF approximation: ring-averaged pixel intensities (points with error bars) and the fitted model $F(r)$ (solid curve).

where the summation runs over pixels with the highest intensity I_k .

To improve robustness, the optimization is performed with up to three restarts. The first Minuit pass keeps secondary curvature and transition-width parameters fixed and fits the primary shape parameters; the second pass releases all parameters and repeats Simplex $\rightarrow$ Migrad. The result with the lowest reduced χ^2 is selected. For N fitted pixels and $k = 11$ free parameters the minimized statistic is

$$\chi_{\text{ndf}}^2 = \frac{1}{\max(N - k, 1)} \sum_i \frac{(F_i - I_i)^2}{\max(I_i, 1)}.$$

An example of an LDF approximation is shown in Fig. 6.

5.3. LDF Integration

After fitting, the radial integral of the LDF is computed to estimate the total shower intensity

$$I_{\text{total}} = 2\pi \int_0^{330} F(r) r dr,$$

using SciPy adaptive quadrature. The upper integration limit (330 mm) corresponds to the focal length of the optical system.

5.4. Parallelism and Data Format

Parallelism is implemented at the process level (`multiprocessing.Pool`), where the atomic unit of work is a single `moshits` file. When the pool is created, each worker loads the detector geometry (2653 pixels) once; thereafter, only the file path is passed between the master and workers. Processing is performed via `imap_unordered` for maximum core utilization. Optionally, pixel coordinates are placed in `shared_memory` to eliminate geometry duplication across processes.

Events are filtered by three criteria: the distance of the shower axis from the detector center (≤ 400 mm), the total intensity after background subtraction ($\geq I_{\text{min}}$), and the distance of the peak from the mosaic center (≤ 270 mm). Results are written to CSV files (separately by particle type) containing the approximated LDF parameters, fit quality metrics, and the radial integral. Thread safety details are discussed in Section 6.

6. DISCUSSION OF THE PARALLEL ARCHITECTURE

The applications described in Sections 3–5 form a unified computational pipeline in which each stage employs its own parallelism model. This section discusses the common architectural principles that ensure the efficiency and correctness of parallel computations throughout the entire suite.

6.1. End-to-End Atomicity

The key property of the simulation problem is its natural atomicity at all levels of the pipeline. Each CORSIKA shower produces an independent Cherenkov light distribution; each clone of that shower corresponds to a statistically independent position of the shower axis; each image on the detector mosaic results from an independent ray-tracing; each LDF approximation is determined solely by the data of a single image. This chain of independence

shower $\rightarrow$ clone $\rightarrow$ mosaic image $\rightarrow$ approximation

guarantees that no processing step for one event requires information about any other event. Consequently, within each stage there is no need for inter-process (or inter-thread) communication during data processing.

The only synchronization points in the entire pipeline are:

- the file system between stages, where the output of one stage serves as the input to the next;
- a thread-safe queue inside the ray-tracing application, requiring a single mutex acquisition per event;
- result accumulation in the fitting stage, where the master process collects outputs as workers complete.

The workload naturally classifies as high-throughput computing (HTC): a large number of independent Monte Carlo events processed without inter-task communication, distinct from tightly-coupled HPC workloads such as MPI-based simulations that require global collective operations.

6.2. Hierarchy of Parallelism Models

Each pipeline stage employs a parallelism model tailored to the nature of its computational workload (Table 3).

The choice of technology at each stage is determined by the structure of the problem. Shower simulation in CORSIKA is completely independent between runs; parallelization is trivial and is realized by an external script that distributes the jobs.

In **sim-clone**, each CLOUT file contains $\sim 10^5 - 10^6$ nonzero cells of the three-dimensional array. The computations are uniform in structure (one formula per cell with Poisson generation), which corresponds to a data-parallel workload at the array-cell level. OpenMP with dynamic scheduling effectively balances the nonuniformity arising from cells with higher intensity producing more photons.

In **sim-trace**, tracing a single photon through the detector geometry requires substantial computation (intersections with tessellated surfaces, multiple reflections and refractions). Geant4 MT natively supports event-level parallelism, where each event is processed by a separate thread. The geometry and physics tables are shared among threads in read-only mode, as guaranteed by the Geant4 architecture.

In **sim-fit**, nonlinear optimization involves calls to the Python interpreter and NumPy operations. The use of processes instead of threads is dictated by the GIL (Global Interpreter Lock) in CPython, which prevents threads from executing Python bytecode in parallel. Process isolation also simplifies state management, as each process has its own address space.

6.3. Thread Safety by Design

In all three applications, thread safety is ensured primarily by architectural means rather than by synchronization primitives: shared data are read-only, and mutable state is isolated per worker.

In **sim-clone**, the Cherenkov light array (670 MB) is loaded before the parallel section begins and is available to all threads as read-only. Each thread generates photons using its own random number generator and writes the result to a private output buffer. The only aggregation—computing the global minimum and maximum arrival times—is performed by the OpenMP runtime via built-in reduction operations, without explicit locks.

In **sim-trace**, the detector geometry and physics process tables are created by the master thread and are available to workers in read-only mode, as guaranteed by Geant4 MT. All mutable state is encapsulated in a per-worker data structure created at worker initialization. The only mutex—protecting the file dequeue operation—is acquired once per event. For a typical event with $\sim 10^5$ photons, the tracing time exceeds the lock acquisition time by orders of magnitude, so contention is negligible.

In **sim-fit**, process isolation guarantees the absence of shared mutable state by construction. When a shared read-only memory segment is used, the detector pixel coordinates are accessible to all workers but are never modified. The background level is computed once in the master process and passed to workers as an initialization argument.

A summary of the thread-safety mechanisms employed by each application is given in Table 4.

6.4. Physical Correctness Under Parallelism

Parallel execution must not violate the statistical properties of the simulation. Correctness is ensured at several levels.

Random number generators are independent across workers. In **sim-clone**, each thread uses its own generator seeded from a combination of hardware entropy and a hash of the thread identifier, guaranteeing statistical independence of the sequences. In **sim-trace**, the CLHEP generator is managed by Geant4, which provides per-thread instances. In **sim-fit**, each process has its own generator state by definition of process isolation.

The simulation result—the set of approximation parameters—does not depend on the number of worker threads or processes, up to stochastic fluctuations caused by differences in random number sequences. Deterministic bit-for-bit reproducibility under changes in the degree of parallelism is not guaranteed and is not the goal; only statistical equivalence of the results matters.

The statistical properties of the output sample are likewise preserved. The Poisson distribution of photoelectron counts, the uniform generation of clone positions, and the stochastic photon acceptance all rely on thread-local generators and do not depend on the order in which elements are processed. Permutation of the processing order — whether of array cells due to dynamic scheduling or of files due to asynchronous collection — does not affect the statistical properties of the sample.

6.5. Scalability

The described architecture possesses the following scalability properties:

- Each stage scales linearly with the number of cores, as a consequence of the absence (or minimality) of inter-thread synchronization. The overhead consists of one-time data loading (the Cherenkov light file, the detector geometry, or the pixel coordinates) and minimal synchronization operations (one mutex acquisition per event in the ray-tracing stage).
- Scaling with data volume is straightforward: increasing the number of showers, clones, or particle types does not require changes to the architecture. Each new event is processed independently, and adding computational nodes proportionally accelerates the processing.
- The pipeline is heterogeneous in its use of parallelism technologies (OpenMP, Geant4 MT, multiprocessing), which allows optimal resource utilization at each step. The file system as the interface between stages provides loose coupling: stages can be executed on different computational nodes sharing a common storage.

The main scalability limitation is I/O: with a sufficiently large number of threads, performance may be bounded by the file system throughput (especially during the writing of intermediate text files). Switching to binary formats or using a RAM disk can alleviate this limitation if needed.

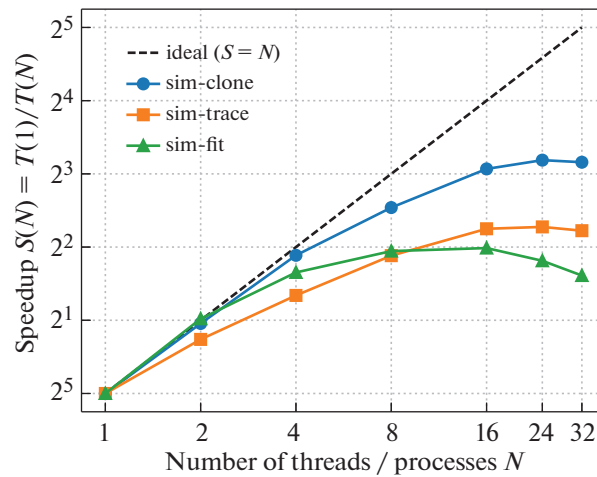


Fig. 7. Measured strong-scaling speedup $S(N) = T(1)/T(N)$ of the three pipeline stages on a 32-thread AMD Ryzen 9 server. Dashed line shows the ideal linear speedup. Each data point is the mean of three trials.

6.6. Measured Scalability

To quantify these claims we ran independent strong-scaling sweeps of the three stages on a 32-thread AMD Ryzen 9 5950x server running Fedora, with $N \in \{1, 2, 4, 8, 16, 24, 32\}$ worker threads or processes and three trials per point. The *sim-clone* workload was one CLOUT4w file of 10 PeV proton showers producing 100 cloned photoelectron drops; *sim-trace* traced 20 such drops against the detector geometry; and *sim-fit* approximated 100 detector-level images with the LDF model described above. Figure 7 shows the measured speedup $S(N) = T(1)/T(N)$, and Table 5 collects the parallel efficiency $\eta(N) = S(N)/N$.

The *sim-clone* stage scales efficiently up to roughly sixteen threads, reaching $S \approx 9.1$ at $N = 24$ before the curve flattens. This plateau is not an artefact of the chosen sample size: a control sweep with two CLOUT files (200 clones) at $N \in \{16, 24, 32\}$ yielded wall times that doubled almost exactly (55, 50, 51 s), matching the single-file values within the run-to-run scatter. The residual floor is a genuine serial overhead, dominated by the single-threaded decompression and parsing of each CORSIKA output that precedes the parallel clone loop. Production workloads processing many CLOUT files amortize this overhead across the pipeline and continue to benefit from additional threads, whereas a single-file run saturates near $S \approx 9$ regardless of core count.

The *sim-trace* stage plateaus earlier, at $S \approx 4.8$ for $N \geq 16$. The cost of initializing the Geant4 detector geometry, the physics list, and the per-thread random number stream is paid once per invocation and does not scale with thread count, so on the 20-shower benchmark this initialization dominates. For production workloads of thousands of showers per invocation the amortized per-shower cost continues to scale closer to linearly.

The *sim-fit* stage shows a soft peak around $N = 16$ ($S \approx 4$) and a slight decline at higher N . Each process handles a relatively short approximation task (hundreds of milliseconds per image), so the relative cost of Python process startup and shared-memory geometry setup grows at high N and eventually offsets the per-task gain. Batching more images per worker or reusing a long-lived pool would push the peak further out; this optimization is orthogonal to the parallel structure and is left to future tuning.

Taken together, the three stages confirm the qualitative claims of Section 6.6.5: each stage scales near-linearly within its parallel fraction, and the plateaus observed here originate from fixed per-invocation overheads rather than from inter-thread contention or architectural serialization.

6.7. Vectorization

Within this work, explicit OpenMP SIMD directives (`#pragma omp simd`, available since OpenMP 4.0) are not employed. The arithmetic segments of the *sim-clone* hot loop—coordinate transformations, radial distance, and the angular efficiency polynomial—are auto-vectorized by

GCC/Clang under `-O3 -ffast-math`. The computational cost of the stage is however dominated by Poisson photoelectron sampling and the associated MT19937 random number generator, both scalar by construction: benefiting from SIMD in this part would require replacing the generator (for example with a SIMD-friendly Philox or xoshiro family) and restructuring the sampling into batched form, which we leave for future work.

7. CONCLUSION

In this work, a software suite with a multi-level parallel architecture for the simulation of the SPHERE-3 experiment is presented—from the generation of extensive air showers to the approximation of registered Cherenkov light images on the detector mosaic.

The key property of the problem is its natural atomicity: each event at all pipeline stages is processed independently of the others, which allows the computations to scale to an arbitrary number of cores without loss of physical meaning. Each stage employs a parallelism model tailored to the nature of the computational workload: OpenMP for data-parallel processing of Cherenkov light cells, Geant4 MT for event-level parallelism during photon tracing, and `multiprocessing` for process isolation during nonlinear optimization in Python. Thread safety is ensured primarily by architectural means—shared data are read-only, mutable state is isolated per-worker—which minimizes synchronization overhead.

The approximation stage implements a robust fitting strategy for the lateral distribution function with multiple restarts and adaptive choice of statistic, which provides stable extraction of shower parameters over a wide range of intensities.

Beyond the reflected-CL pipeline presented here, an analogous processing chain for direct Cherenkov angular images is under development as part of the dual-detector design of SPHERE-3. Joint analysis of reflected and direct CL is expected to substantially improve mass-composition sensitivity; preliminary joint criteria are reported in [13, 14] and will be detailed in a separate publication.

Further development of the suite envisions the integration of the processing stages into a single automated pipeline, as well as the application of the suite to the systematic optimization of the SPHERE-3 detector configuration and the development of criteria for shower classification by primary particle mass.

FUNDING

This work was performed using the equipment of the Shared Research Facilities for Supercomputing at Lomonosov Moscow State University [15]. The authors acknowledge support from the Russian Science Foundation (RSF grant No. 23-72-00006). The work of V.A. Ivanov was supported by Theoretical Physics and Mathematics Advancement Foundation “BASIS” (grant no. 24-2-10-53-1).

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. S. Thoudam, J. P. Rachen, A. van Vliet, A. Achterberg, et al., “Cosmic-ray energy spectrum and composition up to the ankle: The case for a second Galactic component,” *Astron. Astrophys.* **595**, A33 (2016). <https://doi.org/10.1051/0004-6361/201628894>
2. A. E. Chudakov, “Probable method for EAS registration using Cherenkov light reflected from snow surface” in *Proceedings of the All-Union Symposium* (1972), pp. 69–74.
3. R. A. Antonov, S. P. Beschapov, E. A. Bonvech, et al., “Results on the primary CR spectrum and composition reconstructed with the SPHERE-2 detector,” *J. Phys.: Conf. Ser.* **409**, 012088 (2013). <https://doi.org/10.1088/1742-6596/409/1/012088>
4. D. V. Chernov, C. Azra, E. A. Bonvech, V. I. Galkin, V. A. Ivanov, V. S. Latypova, D. A. Podgrudkov, and T. M. Roganova, “SPHERE-3 project for studying the composition of primary cosmic rays in the energy range between 1 and 1000 PeV,” *Phys. At. Nucl.* **85**, 641–652 (2022). <https://doi.org/10.1134/S1063778822060059>

5. D. Heck, J. Knapp, J. N. Capdevielle, G. Schatz, and T. Thouw, “CORSIKA: A Monte Carlo code to simulate extensive air showers,” Report FZKA No. 6019 (Forschungszentrum Karlsruhe, 1998). <https://doi.org/10.5445/IR/270043064>
6. S. Agostinelli, J. Allison, K. Amako, et al., “Geant4—a simulation toolkit,” *Nucl. Instrum. Methods Phys. Res., Sect. A* **506**, 250–303 (2003). [https://doi.org/10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8)
7. F. Acerbi and S. Gundacker, “Understanding and simulating SiPMs,” *Nucl. Instrum. Methods Phys. Res., Sect. A* **926**, 16–35 (2019). <https://doi.org/10.1016/j.nima.2018.11.118>
8. C. J. Poole, I. Cornelius, J. Trapp, and C. M. Langton, “A CAD interface for GEANT4,” *Australas. Phys. Eng. Sci. Med.* **35**, 329–334 (2012). <https://doi.org/10.1007/s13246-012-0159-8>
9. N. N. Kalmykov, S. S. Ostapchenko, and A. I. Pavlov, “Quark–gluon–string model and EAS simulation problems at ultra-high energies,” *Nucl. Phys. B (Proc. Suppl.)* **52B** (3), 17–28 (1997). [https://doi.org/10.1016/S0920-5632\(96\)00846-8](https://doi.org/10.1016/S0920-5632(96)00846-8)
10. S. Ostapchenko, “LHC data on inelastic diffraction and uncertainties in the predictions for longitudinal extensive air shower development,” *Phys. Rev. D* **89**, 074009 (2014). <https://doi.org/10.1103/PhysRevD.89.074009>
11. H. P. Dembinski, P. Ongmongkolkul, et al., “scikit-hep/iminuit v2.32.0,” Zenodo (2025). <https://doi.org/10.5281/zenodo.3949207>
12. F. James and M. Roos, “Minuit—a system for function minimization and analysis of the parameter errors and correlations,” *Comput. Phys. Commun.* **10**, 343–367 (1975). [https://doi.org/10.1016/0010-4655\(75\)90039-9](https://doi.org/10.1016/0010-4655(75)90039-9)
13. E. A. Bonvech, N. O. Ovcharenko, O. V. Cherksova, et al., “Estimation of extensive air shower primary parameters using Cherenkov light angular distribution in the SPHERE-3 experiment,” *Bull. Russ. Acad. Sci.: Phys.* **89**, 1007–1010 (2025). <https://doi.org/10.1134/S1062873825711572>
14. T. A. Dzhataoev, E. A. Bonvech, O. V. Cherksova, et al., “A combined approach to study the composition of 1–1000 PeV cosmic rays by EAS direct and reflected Cherenkov light detection,” *PoS (ICRC2025)*, 246 (2025). <https://doi.org/10.22323/1.501.0246>
15. V. V. Voevodin et al., “Supercomputer Lomonosov-2: Large scale, deep monitoring and fine analytics for the user community,” *Supercomput. Front. Innov.* **6** (2), 4–11 (2019). <https://doi.org/10.14529/jsfi190201>
16. R. A. Antonov, T. V. Aulova, E. A. Bonvech, V. I. Galkin, T. A. Dzhataoev, D. A. Podgrudkov, T. M. Roganova, and D. V. Chernov, “Detection of reflected Cherenkov light from extensive air showers in the SPHERE experiment as a method of studying superhigh energy cosmic rays,” *Phys. Part. Nucl.* **46**, 60–93 (2015). <https://doi.org/10.1134/S1063779615010025>

Publisher’s Note. Pleiades Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

AI tools may have been used in the translation or editing of this article.