2024 年度 卒業論文

ALICE 実験グリッド・コンピューティング 広島拠点の高度化を伴う再構築

2025 年 2 月 7 日

指導教員 志垣賢太 教授
主査 志垣賢太 教授
副査 樋口克彦 准教授

広島大学
クォーク物理学研究室

学籍番号 B210228
氏名 松本拓磨

概要

広島大学では ALICE 実験のグリッド・コンピューティング拠点として広島 Tier2 サイトを 2008 年に設立し運用してきた。実験による実データやシミュレーションによるデータの統計量は常に増加し続けており計算リソースは絶えず増強が必要である。そのため、各研究機関が持つノードを活用するグリッド・コンピューティングが重要である。このうち、Tier2 は国レベルのグリッド計算拠点であり、現在は主に物理を理解するために必要なモンテカルロ・シミュレーションによるデータを生成する。私は、グリッドミドルウェアの進展への対応のため 2024 年頭より停止していた広島サイトの OS を入れ替え新たなミドルウェアを入れ再稼働させた。また、1,280 コアへの計算ノードの拡張に加え、新しいストレージシステムに対応し高可用性を確保するためにストレージシステム管理用のノードの追加とシステムストレージ容量を 2 倍へ拡張を実施している。将来的に長崎にもグリッドサイトを構築し、日本における計算リソースの拡張を見据えている。この卒業論文では、広島 Tier2 サイトの高度化を伴う再構築と再稼働における主要な取り組みについて記述し、今後の発展について議論する。

目 次

図 目 次

表 目 次

第1章　序章

1.1　量子色力学

素粒子間に働く基本相互作用としては「電磁相互作用」、「弱い相互作用」、「強い相互作用」、「重力相互作用」の４つの相互作用がある。基本相互作用の中で、クォークとグルーオン間に働く強い相互作用は量子色力学（QCD) によって記述される。強い相互作用はグルーオンによって媒介される。

クォークはカラーと呼ばれる内部自由度を持っている。クォークのカラーは光の三原色にちなんで赤、青、緑の 3 種類あり、反クォークは反カラーを持つ。クォークやグルーオンは単体では存在できないが、粒子が複合しカラーを無色にすることで安定し存在することができる。通常の物質状態ではクォークやグルーオンは核子内に閉じ込められている。高温または高バリオン数密度ではクォークやグルーオンが核子内に閉じ込められなくなり自由に動き回ることができる、クォーク・グルーオン・プラズマ（QGP）という物質相になる。理論的に予想される QCD 物質の相図 (図 1.1) を示した。横軸が正味バリオン数密度、縦軸が温度であり、通常のハドロンからなる物質状態は図の左下のエリアの「ハドロンガス」である。ハドロンガスから高温もしくは高バリオン数密度状態のところに QGP 相が存在する [1]。

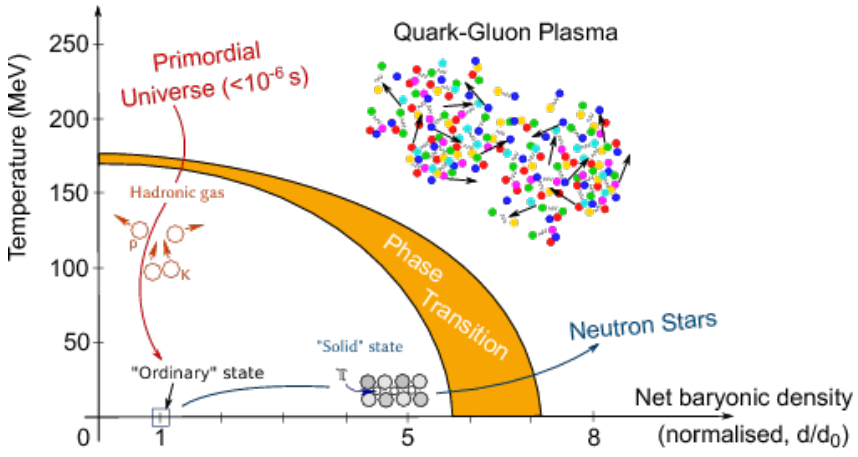


図 1.1: QCD 相図 [8]

1.2　高エネルギー重イオン衝突実験

高エネルギー重イオン衝突実験は、粒子加速器で重イオンを加速し衝突させることにより、ビッグバン直後の宇宙初期に存在したとされるクォーク・グルーオン・プラズマを人工的に生成し研究する。ブルックヘブン国立研究所の相対論的重イオン型加速器 Relativistic Heavy Ion Collider

（RHIC）や、欧州原子核研究機構（CERN）の Large Hadron Collider (LHC) で実験が行われている。

1.2.1 LHC

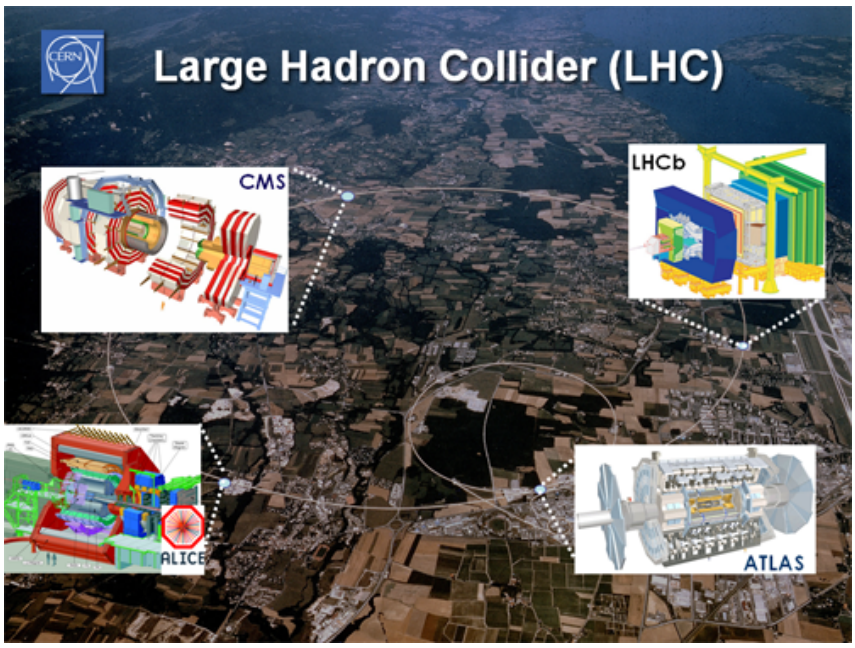


図 1.2: LHC の全体図 [10]

スイスとフランスの国境に位置する LHC は CERN によって運用される世界最大のハドロン衝突型円形加速器である。LHC は全長 27 km の円形トンネル内に設置され超電導磁石リングと多数の加速構造から構成され、高エネルギーの陽子や鉛原子核を衝突させる。LHC のリング状に位置する 4 つの実験、ALICE、ATLAS、CMS、LHCb (図 1.2) の検出器内で LHC で加速された粒子は衝突し、衝突点における重心衝突エネルギーは最大 13.6 TeV にも及ぶ。LHC の各実験では、それぞれ着目している物理が異なる。

1.2.2 ALICE 実験

ALICE 実験（A Large Ion Collider Experiment）は LHC の中で重イオン衝突に特化した実験である。この実験の主な目的は LHC を用いて高温状態を作り出し、QGP の性質を解明することを目的としている。ALICE 検出器は、重イオン衝突によって生成される膨大な数の粒子の運動量やエネルギーを観測するための設計である (図 1.3)。実験によるデータの統計量は増え続けるので、計算リソースは絶えず増強が求められる。生成されるデータの処理には、グリッド・コンピューティングが重要な役割を果たしている。

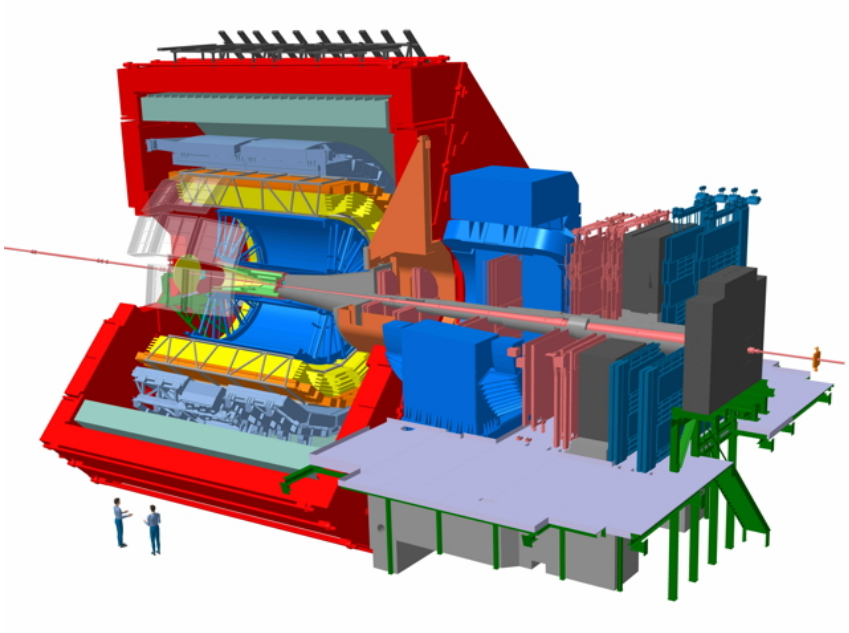


図 1.3: ALICE 検出器 [11]

1.3 グリッド・コンピューティング

1.3.1 グリッド・コンピューティング

グリッド・コンピューティングとは、複数のコンピュータをネットワーク上で組み合わせて、広域分散型スーパーコンピュータを構築する、分散型コンピューティング・システムである。グリッド・コンピューティングは処理負荷を複数のシステムに分散し、コンピューティングプロセスの速度を大幅に向上させ、効率的に大規模な計算を処理することが可能となる。グリッド・コンピューティングは、地理的に分散したハードウェアリソースをインターネットや仮想プライベートネットワークを介して接続し、各ノードが大きなタスクの一部を処理することで機能する。また、必要に応じてノードを追加することで、計算能力を拡張することができる。
また、以下の要因により高い信頼性も確保している [13]。

1. 計算処理の冗長性確保

グリッド・コンピューティングでは、複数のコンピュータを地理的に分散して配置している。そのため特定のノードが故障したりネットワーク障害が起こった場合、他のノードがタスクを処理するためシステムを継続的に運用することができる。この冗長性によりシステムの可用性は高まる。

2.Fault Tolerance

障害発生時、タスクが別のノードに移り再実行される。これにより計算の中断を最小限に抑えることができシステム全体の安定性を確保する。

3. 負荷分散

グリッド環境では、計算タスクが複数のノードに分散する。これにより特定のノードへ過度な負荷が集中することを防ぎ、障害のリスクを減少させる。負荷分散により、全体的な処理能力が向上し安全性が高まる。

4. ディスクデータの複製

複数のノード間で重要なデータは複製されるため、特定のノードが故障してもデータの損失リスクが減少する。この仕組みにより、システム全体の耐障害性が向上する。

1.3.2 高エネルギー重イオン加速器実験とグリッド・コンピューティング

高エネルギー重イオン加速器実験では、大量のデータを取得して解析する。また、解析による物理理解のために用いる膨大なモンテカルロシミュレーション（MC）データの生成も必要なため、非常に多くの計算リソースを必要とする。計算リソースの確保には巨大な計算機施設を作ることがあげられるが限界がある。そこで、グリッド・コンピューティングを用いて、世界中の大学や研究所などにあるコンピュータを使うことにより計算リソースを確保している。グリッド・コンピューティングは高エネルギー重イオン加速器実験において重要な役割を担っている。

1.3.3 ALICE 実験のグリッド・コンピューティング

ALICE 実験では、毎秒 3.5 TB/s の生データが ALICE 検出器より取得する。生データを研究者が解析に用いる AOD（Analysis Object Data）にする必要がある。毎秒数 PB の実験データや、MC を処理するため、世界中の計算機資源を組み合わせ計算グリッドを構成している。

グリッド・コンピューティング構造

ALICE 実験のグリッド・コンピューティング環境は Tier という階層型構造に基づき構成している。Tier は Tier0、Tier1、Tier2、Tier3 と 4 種類あり、各 Tier の役割は異なる。以下に各 Tier について記す。

・Tier0

CERN にあり、ALICE 実験の検出器で取得した実データの初期処理、保存、分配の中枢的な役割を持つ。実データの Tier1 サイトへ分配、コピーの保存、運用管理を行う。

・Tier1

各地域の主要なデータセンターで、Tier0 から受け取ったデータを保存し、Tier2 に分配する。実データの再処理やシミュレーションデータの生成も行う。

・Tier2

地域拠点級のサイトである。Tier1 からデータを受け取り、必要に応じて Tier1 へデータを返送する。データ解析の主要な実行拠点で、物理解析に必要な計算リソースを提供する。シミュレーションデータの生成も行う。

	拠点	使用 CPU コア数	ディスク容量	使用用途
Tier0	CERN	～75K	363.8 PB	イベント再構成＋ MC ＋解析
Tier1	地域の主要拠点	～30K	154.5 PB	イベント再構成＋ MC ＋解析
Tier2	地域拠点	～70K	74.63 PB	MC ＋解析

表 1.1: ALICE 実験グリッドサイトリソース

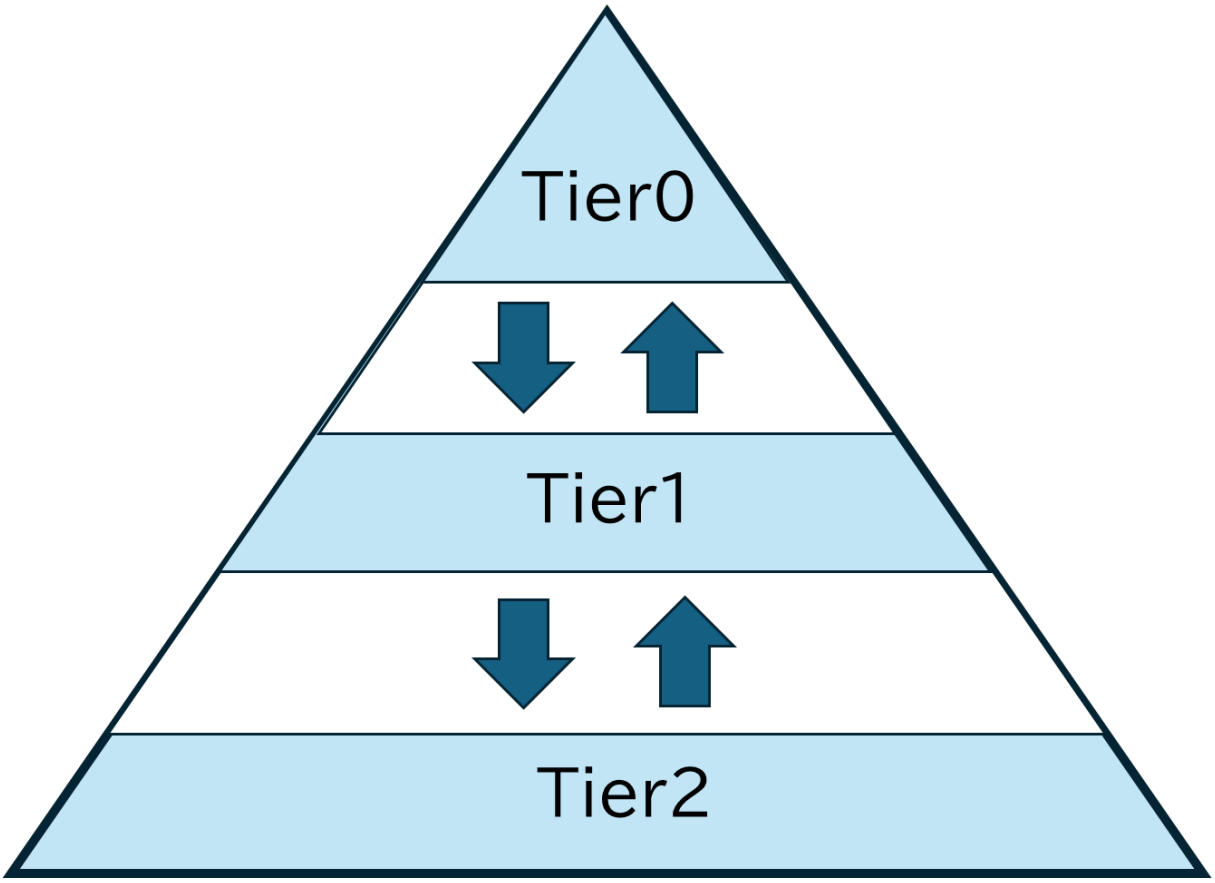


図 1.4: Tier 構造

広島サイト

ALICE 実験のグリッドサイトは現在 Tier0 が CERN に、Tier1 が 7 個、Tier2 が 60 個程度存在する (図 1.5)。広島サイトは Tier2 の一つである。広島大学では 2008 年に ALICE 実験のグリッド・コンピューティングサイトとして、広島サイトを設立し、運用してきた。現在、日本にある

ALICE 実験のグリッドサイトは広島サイトのみである。広島サイトでは主にモンテカルロシミュレーションによるデータを生成している。

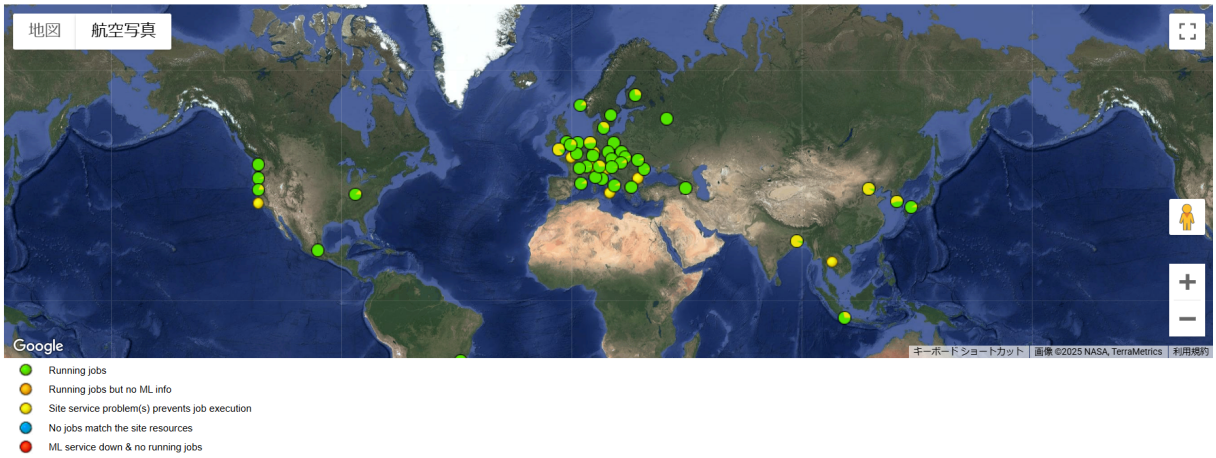


図 1.5: ALICE 実験のグリッドサイト [5]

1.4 研究目的

ALICE 実験で取得する実データや、シミュレーションによるデータの統計量は常に増加し続けており、計算リソースは絶えず増強が必要である。本研究では、ALICE 実験の計算リソース増強のため、2024 年 1 月より停止していた、広島サイトの再構築を行った。再構築では ALICE 実験の現在のグリッドフレームワークに対応した管理ノードの構築、計算用ノードの構築、ストレージシステムの構築が求められ、本研究では管理ノード、計算ノードの構築を行った。

1.5 本論文の構成

本論文の構成を示す。第 2 章に広島サイトに用いたノードの性能、サイト設計と各ノードの基本設定、簡潔なサイトの構成手順を記す。第 3 章で実際に行ったサイト構築の主要な作業について記す。主要な作業は第 2 章の構成手順をより詳細に記す。第 4 章で、再稼働した広島サイトの現状と今後の構築課題について記す。

第2章　グリッドサイトの設計

2.1　広島サイトの各ノード性能

広島大学 理学部 D 棟 303 にある (図 2.1) ノードの一部を用いて広島サイトを構築した。広島サイトは grid00 から grid07 と、qx301 から qx320、nfs11 から nfs13 の計 31 台のノードを用いて構築する。grid00 から grid07 を Core Server、qx301 から qx320 を Worker Node、nfs11 から nfs13 を Storage Node と呼ぶ。Core Server の内 grid01 はグリッドサイトの管理用ノード、grid03 をキャッシュプロキシ、grid04 から grid06 はストレージの管理用ノードとして構築する。Worker Node である qx301 から qx320 は計算用ノードとして構築する。nfs11 から nfs13 はストレージとして構築する。

図 2.1: サーバールーム

2.1.1 各ノード性能

各ノードの性能を記載する（表 2.1）。
Worker Node はそれぞれ 64 コア持っているので、広島サイトは 1,280 コアとなる予定である。

表 2.1: 各ノードの仕様

項目	仕様
Core Server grid00～07	
プロセッサ	AMD EPYC 7302P 16 Core
メモリ	64 GB
ストレージ	2 TB HDD × 2（ミラーリング）
Worker Node qx301～320	
プロセッサ	AMD EPYC 7302 16 Core × 2（64 Core、Hyper Threading）
メモリ	192 GB
ストレージ	1 TB HDD × 1
Storage Element nfs11	
プロセッサ	AMD EPYC 7302P 16 Core
メモリ	64 GB
ストレージ	2 TB HDD × 2（ミラーリング）、16 TB HDD × 57（RAID6）
Storage Node nfs12,13	
プロセッサ	AMD EPYC 7262 8 Core
メモリ	64 GB
ストレージ	1 TB SSD × 2、12 TB HDD × 36（ZFS RAIDZ3）

また、Storage Node は停止前は nfs11 のみ使用しストレージ容量は 864 TB であった。再構築に伴い nfs12 と nfs13 を追加しストレージ容量は 1,656 TB にする。

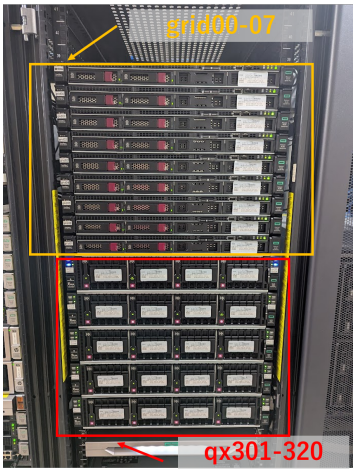


図 2.2: grid00-07 と qx301-320

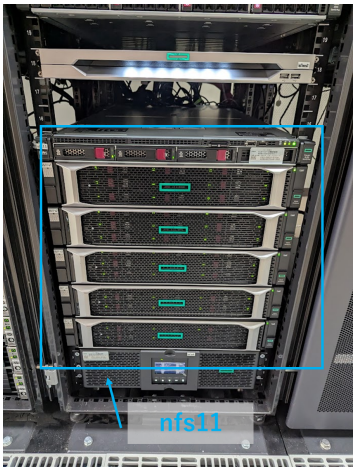


図 2.3: nfs11

図 2.4: nfs12、nfs13

RAID と RAIDZ

ストレージ用のノードである nfs11 から nfs13 のストレージは RIAD 構成である。RIAD（Redundant Array of Inexpensive Disks）は複数のドライブを組み合せ、データの高速化や、冗長性を確保する技術である。nfs11 が該当する。一方 RAIDZ は ZFS（Zettabyte File System）で RAID の仕組みを基盤に実装された。高い耐障害性を持つ。nfs12、nfs13 が該当する。
RAID 構成により、ディスクの破損時にデータの損失を防ぐことができる。RAID6 は 2 つのディスクの破損まで許容され、RAIDZ3 は 3 つのディスクの損傷まで許容される。

2.2 基本設定

2.2.1 OS

ALICE 実験のソフトウェアを実行するため、使用するノードの OS はすべて AlmaLinux 9 にする。

2.2.2 ネットワーク構成

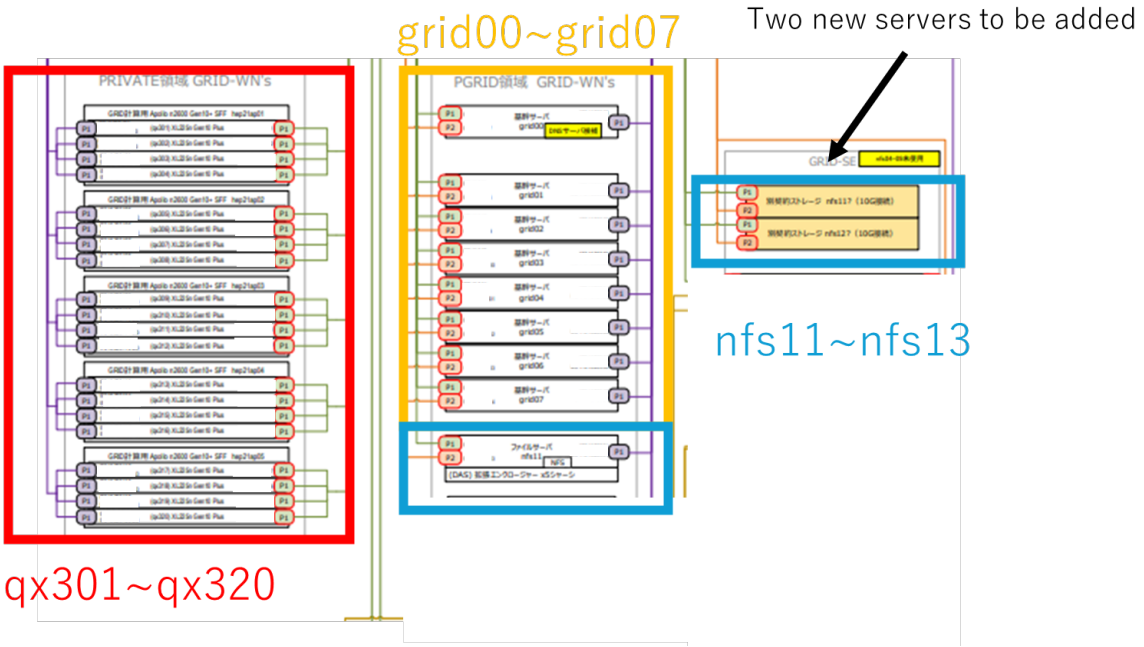


図 2.5: 広島サイト構成図

ネットワーク設定を行う。図 2.5 に記載している。grid00 から grid07 と nfs11 から nfs13 はグローバル IP とローカル IP を持っているので、両方とも設定し接続を行う。qx301 から qx320 はローカル IP しか持っていないので、ローカル IP のみ接続を行う。

2.3 サイトの設計

サイトは 3 種類の構成ノードからなる。Core Server、Worker Node、Storage Node の三つである。Core Server はサイト管理ノードやキャッシングプロキシ、ストレージの管理ノードの役割を持つ。サイト管理ノードは Central Manager と呼び広島サイト全体を管理する。Worker Node は広島サイトに振り分けられたジョブを処理する。Storage Node は広島サイトにデータを保存するストレージとしての役割がある。

2.3.1 サイト構築手順

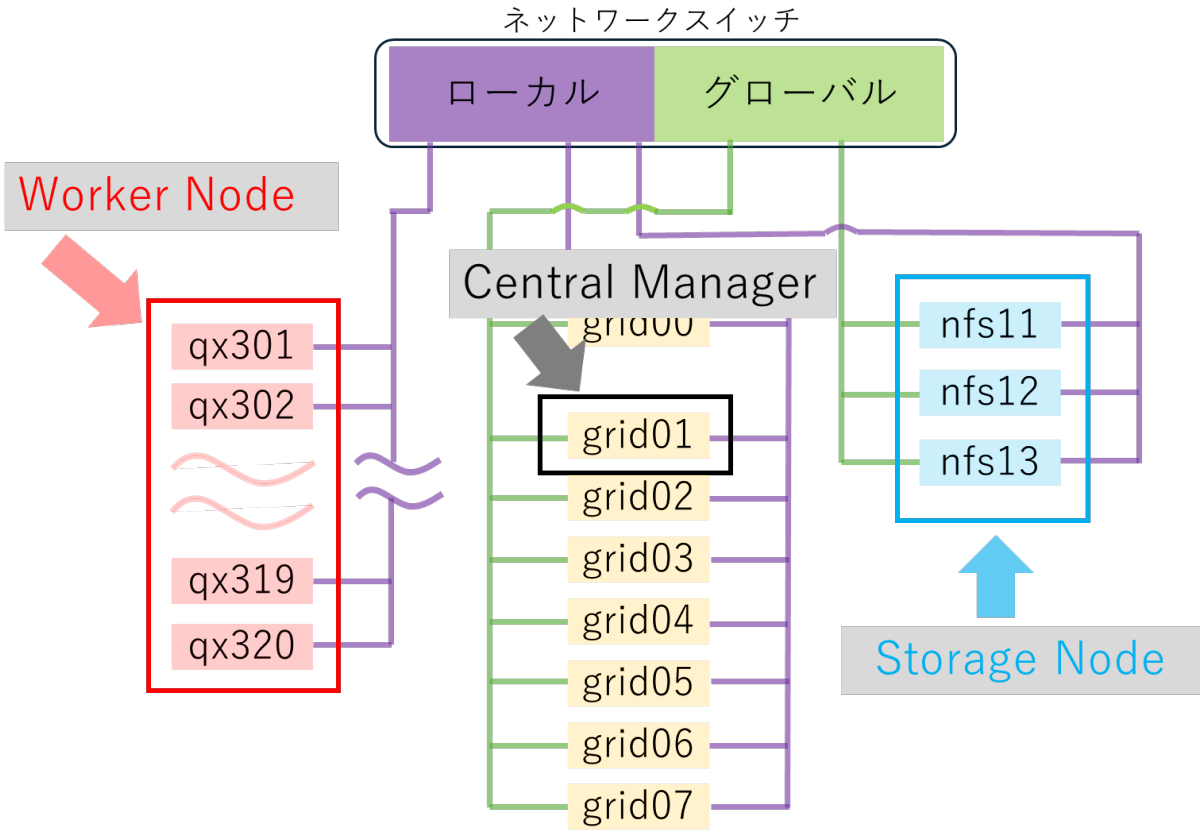


図 2.6: 各ノードの役割

図 2.6 に 3 種類のノードと使用するものを示した。Central Manager として grid01 を使用する。Worker Node として qx301 から qx320 を使用する。そして、ストレージノードとして nfs11 から nfs13 を使用する。また、grid04 から grid06 はストレージノードを管理するノードとして使用し、grid03 は広島サイトのキャッシングプロキシとして使用する。

初めに、キャッシングプロキシとして使用する grid03 の構築を行う。その後、Central Manager を構築し、Worker Node、Storage Node の順で構築を行う。

・構築手順

キャッシングプロキシ → Central Manager → Worker Node → Storage Node

現在再構築が完了したキャッシングプロキシと Central Manager、Worker Node についてそれぞれ簡潔に構成手順順を記す。詳細な構築作業は第 3 章に示した。

キャッシングプロキシの構築手順

grid03 を広島サイトのキャッシングプロキシとして構築する。広島サイトのネットワーク負荷軽減のため、プロキシサーバーの機能を持つソフトウェアである Squid を利用する。grid03 に Squid をインストールする。grid03 は Central Manager と Worker Node の構築時に、CVFMS のキャッシングプロキシとして利用する。

Central Manager の構築手順

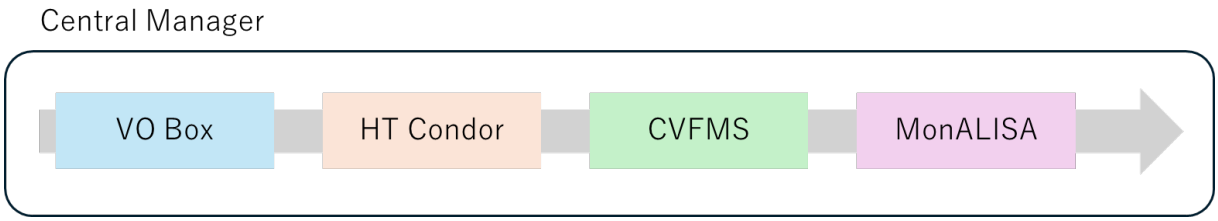


図 2.7: Central Server の構築手順

grid01 を Central Manager として構築する。グリッドサイトの運用に重要な Virtual Organization の管理、運用を行うミドルウェアである VO Box をインストールする。その後、ジョブのスケジューリングや実行にかかわる HT Condor のインストールをする。また、解析やシミュレーションに必要なライブラリを CVFMS を通じて入手する。その後、MonALISA を起動しグリッドサイトの運行状態を MonALISA を通じて把握する。以上の手順で、Central Manager の構築を行い広島サイトを再稼働する。

表 2.2: Grid Computing における主要ソフトウェアの役割

名称	説明
VO Box	VO Box はグリッドジョブとデータ転送を管理するために使用。
HT Condor	ジョブスケジューラーおよびワークロード管理システム。
CVMFS	グリッドサイトに実験用ソフトウェアやライブラリを配布するために使用。

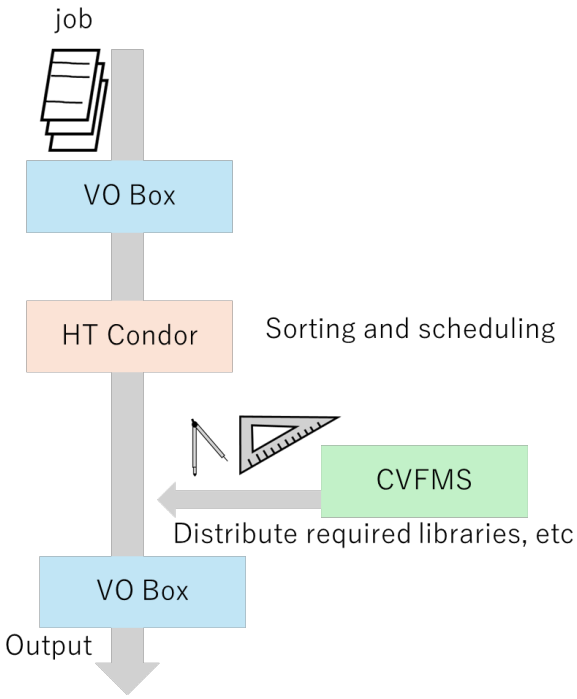


図 2.8: Central Manager における各パッケージの役割

Central Manager における各ソフトウェアの役割を図 2.8 に表す。広島サイトにジョブが来ると初めに VO Box で処理が行われる。ここでは、 HT Condor では、ジョブを効率的にスケジューリングし実行する。この際ジョブは分散されるが、CVFMS が分散環境でも一貫性を維持するための役割を担っている。そして、処理が終わったら VO Box と介してデータは渡される。

Worker Node の構築手順

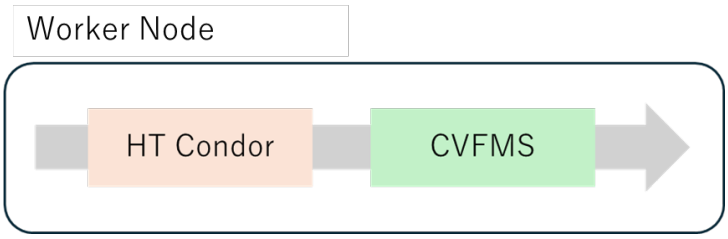


図 2.9: Worker Noder の構築手順

qx301 から qx320 を Worker Node として構築する。Worker Node はジョブの実行のみを行うので HT Condor と CVFMS のインストールを行う。Central Manager が Worker Node を認識しジョブを振り分ける。

Storage Node の構築手順

現在検討中である。

第3章　グリッドサイトの構築作業

OS の入れ替えやネットワーク設定は主に広島で行った。Central Manager や Worker Node の構築は CERN で ALICE 実験のグリッドサイトを確認しながら行った。以降、サイトの構築作業について示す。

3.1　基本設定作業

3.1.1　OS

Core server は grid01 から grid07 までの OS を AlmaLinux 9 に入れ替えた。Worker Node は qx301 から qx320 まですべての OS を入れ替えた。Storage Element は nfs11、nfs12 の OS を入れ替えた。

3.1.2　ネットワーク構成

Core server はローカル IP とグルーバル IP を持っており、すべて接続を完了した。Storage Element もローカル IP とグローバル IP を持っている。nfs11 と nfs12 は接続を行ったが、nfs13 はネットワークスイッチが現在足りていないため、グローバル IP しか接続できていない。Worker Node はローカル IP のみ持っており、すべて接続を完了した。

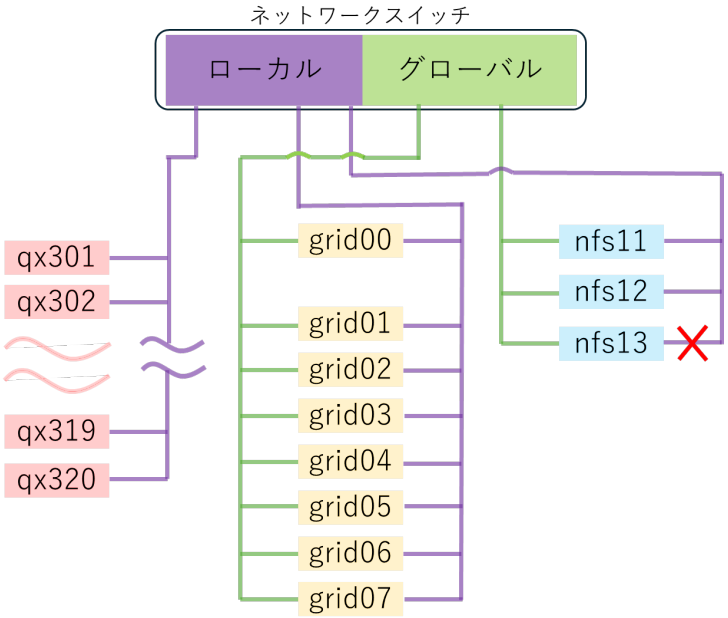


図 3.1: ネットワーク接続状況

3.2 キャッシングプロキシの構築

Squid は Web 用のキャッシングプロキシである。頻繁にアクセスされる Web コンテンツのコピーを保存することで、同一データをサーバーから繰り返し取得する必要性を軽減し、ネットワークトラフィックを最適化を行う。grid03 に Squid をインストールした。grid03 を広島サイトのネットワーク負荷軽減のために利用する。

Central Manager と Worker Node において、grid03 をソフトウェアの配布を行う CVFMS のキャッシングプロキシとして利用する。

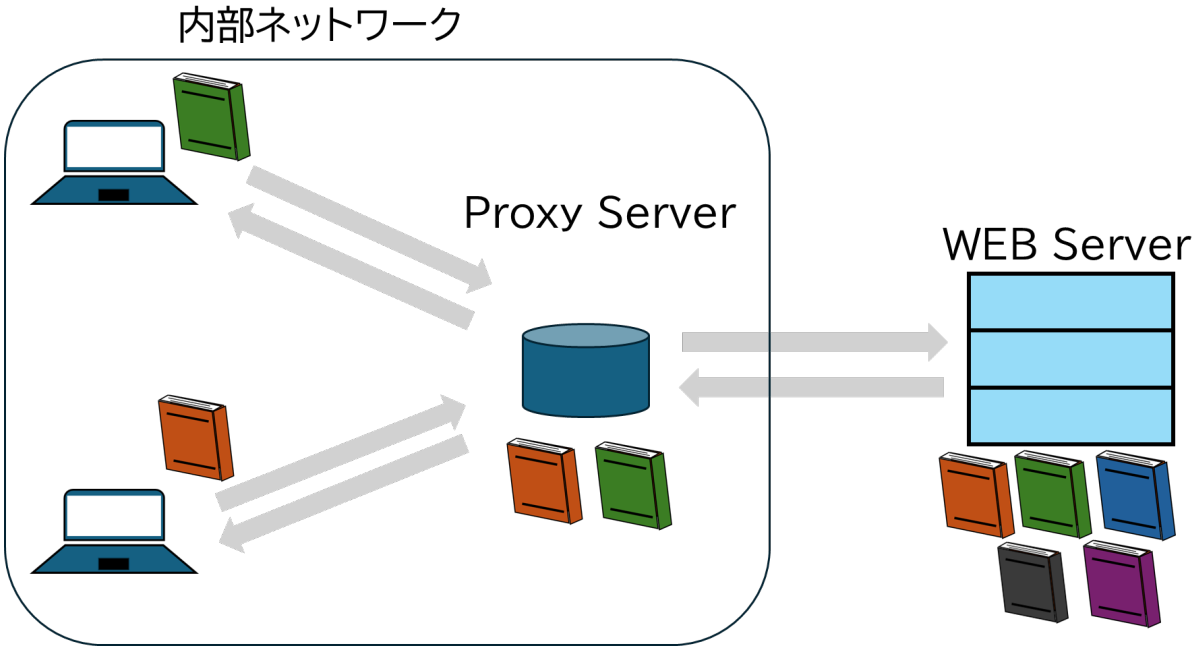


図 3.2: キャッシングプロキシの概念図

3.3 Central Manager の構築

grid01 を広島サイトの Central Manager として構築を行った。図 2.7 に従い構築を行った。以下その詳細を記す。

3.3.1 VO Box (Virtual Organization Box) のインストール

VO Box は Grid コンピューティングにおいて、Virtual Organization（VO）（図 3.3）の管理と運用をサポートするミドルウェアである。VO Box を通して、ジョブやデータ転送の管理を行う。VO Box は VO に特化したツールやソフトウェアをホストし、管理者が Grid ジョブを提出・監視するための中間レイヤーとして機能する。ALICE 実験ではこの VO Box を用いる。VO Box はグリッドサイトとしての玄関口のようなものである。

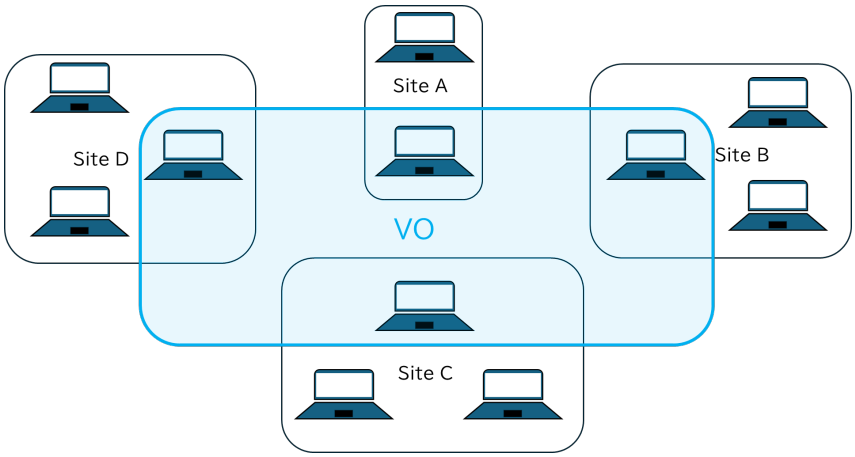


図 3.3: VO の概念図

Central Manager の配置のため、grid01 へ VO Box のインストールを行った。VO Box の設定ファイルである A.“site-info.def” の作成をし、B.gsi 認証、ミドルウェアの設定を行う C.YAIM の起動を行った。それぞれ簡単に記す。

A.“site-info.def” の作成

VO Box の設定ファイルを ALICE 実験のデータにアクセスするために変更する必要がある。“site-info.def” をルートディレクトリに作成した。“site-info.def” は広島サイトが ALICE 実験のグリッドサイトであることを記述した。その後 “groups.conf” と “users.conf” をルートディレクトリに作成した。“groups.conf” と “users.conf” は grid01 にグリッド環境用のグループとユーザーを定義した。

B.gsi 認証

ALICE 実験のグリッド・コンピューティング拠点となるには、証明書が必要である。証明書を KEK（高エネルギー加速器研究機構）から取得した。取得した証明書を使い認証を行った。
この設定により YAIM を実行できるようになる。

C.YAIM (Yet Another Installation Manager) の起動

Central Manager の Certificate のインストール完了後、YAIM を起動した。YAIM (Yet Another Installation Manager) は、分散コンピューティング環境である Grid システムにおいて、効率的なミドルウェア設定や管理を行う。

3.3.2 HT condor のインストール

Central Manager である grid01 に HT Condor をインストールした。Central Manager における HT Condor はホストとして構築した。
HT Condor はジョブのキューイングやスケジューリングを行うため、計算リソースを効率的に活用する。HT Condor は grid01 ではジョブの実行を行わず、Worker Node へのジョブの割り振

りを行う（図 3.4）。HT Condor は計算クラスタを管理する役割を持つ。Central Manager の構築にあたり、Worker Node を追加するまでは HT Condor は grid01 でジョブを実行する。クラスタを管理するため、パスワードを生成し grid01 のルートディレクトリに配置した。

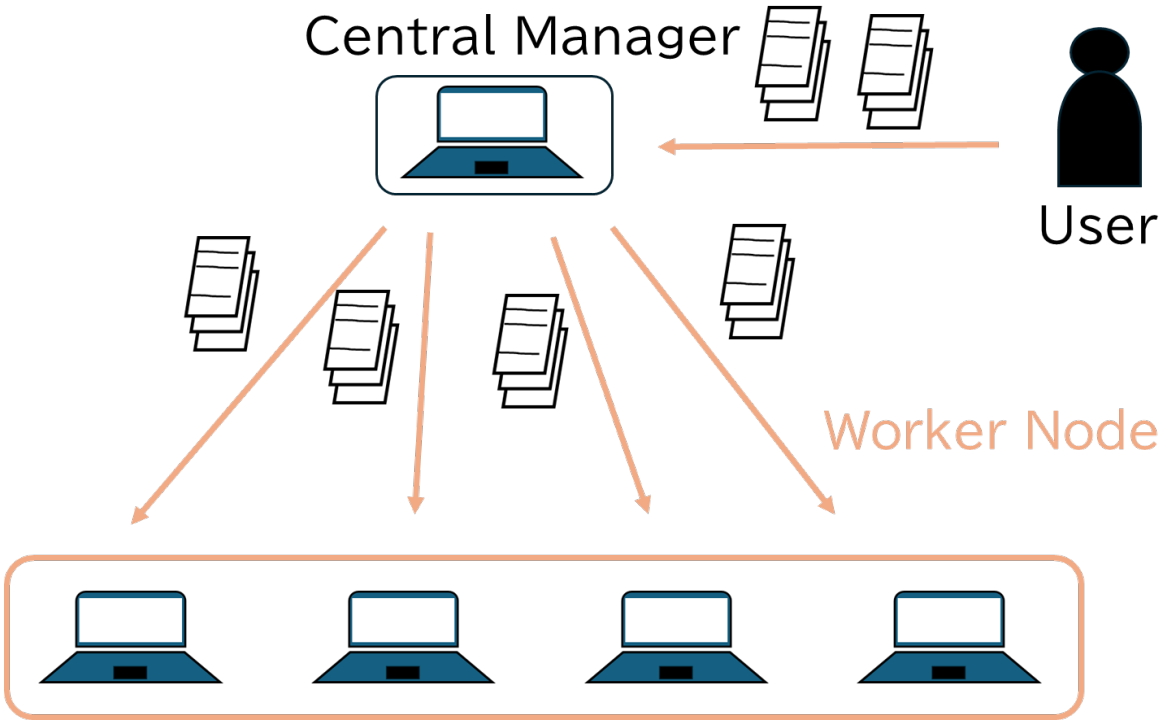


図 3.4: HT Condor によるジョブ管理

HT Condor のインストールにより、広島サイトの各ノードの状況を確認できる。図 3.5 は grid01 が condor に認識されており、Worker Node を追加すると qx301 から qx320 が追加される。図 3.5 では広島サイトにジョブを流していないため、Idle 状態となっている。ジョブを実行すると Busy 状態になる。

```
condor_status
Name                                OpSys      Arch   State     Activity

slot1@grid01                        LINUX      X86_64 Unclaimed Idle

               Total Owner Claimed Unclaimed Matched Preempting  Drain Backfill BkIdle

  X86_64/LINUX     1     0       0         1       0          0      0        0      0

         Total     1     0       0         1       0          0      0        0      0
```

図 3.5: grid01

3.3.3 CVFMS（CERN Virtual Machine File System）のインストール

CVMFS (CERN Virtual Machine File System) を grid01 にインストールした。CVMFS はグリッド・コンピューティング環境におけるソフトウェア配布システムとして利用する。CVMFS を用いて、実験ソフトウェアやライブラリをサイトに配布し、サイト全体で一貫したソフトウェア環境を構築する。また、CVFMS は必要となるまで、データの読み込みを行わない Lazy Loading 方式（図 3.6）を用いているため、データ転送の効率化が図られる。また、ネットワーク負荷軽減のため grid03 を CVFMS のキャッシングプロキシとして設定した。

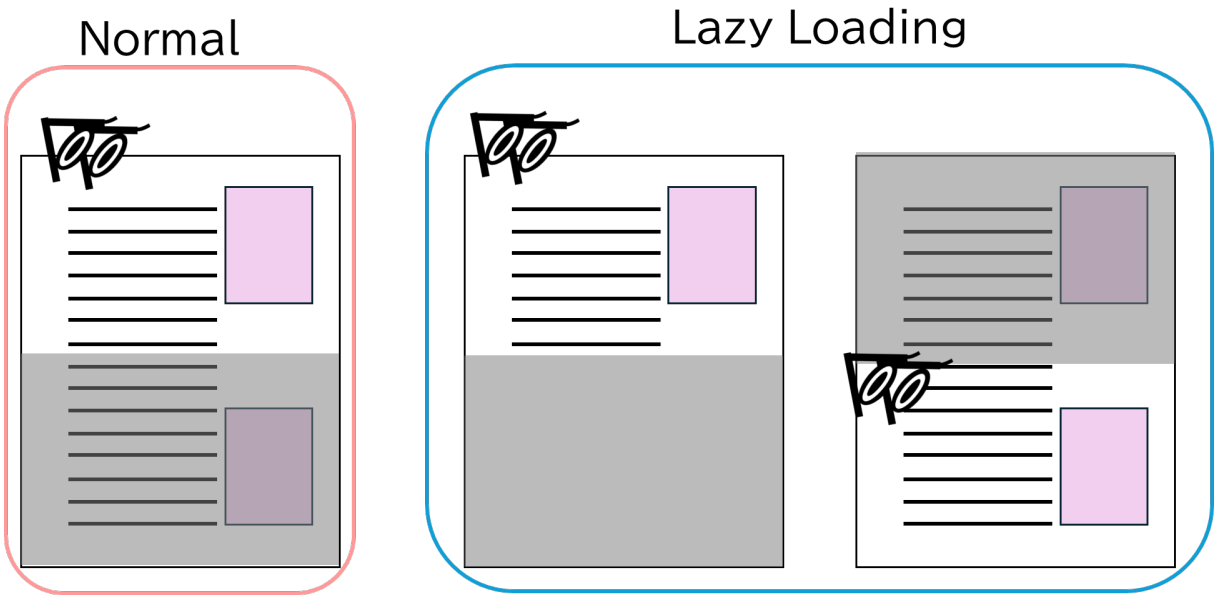


図 3.6: Lazy Loading 方式

3.3.4 MonALISA(Monitoring Agents in A Large Integrated Services Architecture) の起動

Central Manager に各ソフトウェアとミドルウェアのインストール完了後に、MonALISA を起動した。MonALISA を起動することによって、サイトの運行状況を視覚的に確認できる ALIMonitor で広島サイトを確認できる。MonALISA はグリッド環境をリアルタイムで監視、管理を行うフレームワークであり、ALICE 実験では MonALISA で得られた情報を ALIMonitor を通じて可視化している。広島サイトの停止時は ALIMonitor に広島サイトのプロットは存在しなかったが、MonALISA の起動に伴い広島サイトのプロットを再点灯することに成功した。図 3.7 は起動直後に確認した ALIMonitor の画像である。緑のプロットはジョブを実行できていることを表し、黄色のプロットはジョブが実行できていないことを表す。まだ、広島サイトでジョブが実行できていないことがわかる。

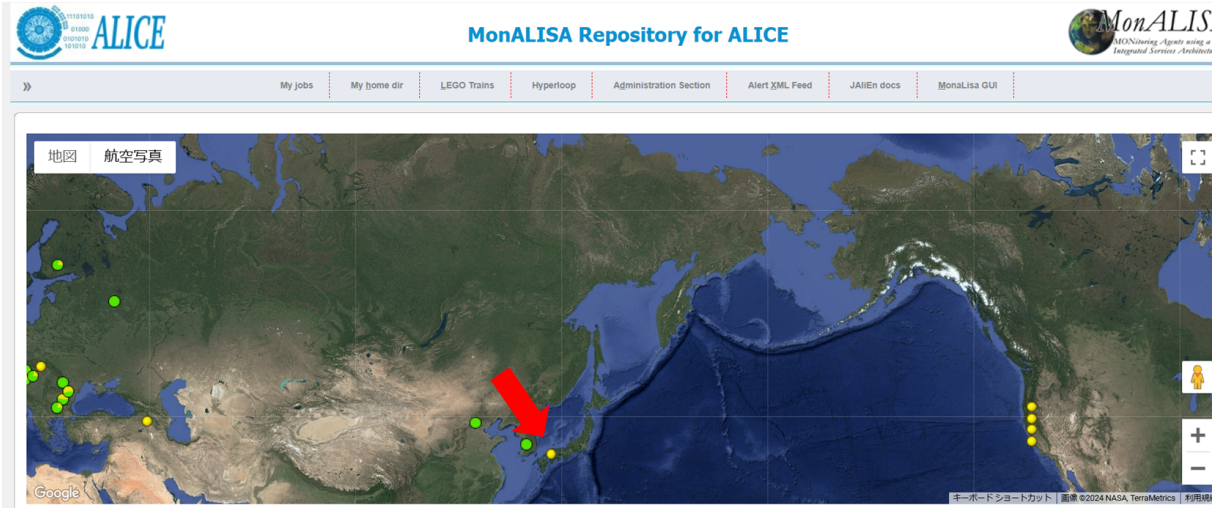


図 3.7: ALIMonitor で広島サイトの確認 [4]

ユーザー認証

ジョブを実行するため、広島サイトが ALICE 実験のグリッドサイトである証明を行った。広島サイトにおける計算処理の認証を実現するため、ALICE 実験よりサーバー証明書と秘密鍵（Key）を取得した。取得した証明書により、Central Manager 内のユーザー”alicegm”が ALICE 実験のグリッドサイトとして認証されたユーザーであることを証明した。

広島サイトにおける Central Manager の構築を完了させ、認証機構が適切に機能するように設定をおこなった。これにより、グリッド・コンピューティング環境における広島サイトが、ジョブを受け入れ実行する拠点として再稼働した。

3.4 Worker Node の構築

計算リソースを拡張するため、Worker Node を構築した。Worker Node は計算用の Node で Central Manager が割り振ったジョブを実行する。広島サイトは Worker Node として構築するノードを 20 台有している。これらを Worker Node として構築する方法（図 2.9）について記述する。

3.4.1 Worker Node の HT condor のインストール

初めに Worker Node に HT Condor をインストールした。Worker Node では HTCondor がジョブの処理を管理する。Central Manager の構築にも HT Condor を使用したが、Worker Node では使用するプロセスが異なる。Central Manager の利用した `shedd` ではなく、Worker Node では `startd` を利用しジョブを処理する。Central Manager の HT Condor 構成時に作成した、広島サイト内での共通のパスワード（3.3.2 節）を Worker Node のルートディレクトリに複製する。また、Central Manager と Worker Node は違うノードなので、サーバーの認証情報を共有しデータの受け渡しを行うため同じ POOL に入れる（図 3.8）。Worker Node で Central Manager から振り分け割れたジョブを実行することが可能となる。

設定が完了すると Worker Node が認識される。condor で Central Manager に加えて新たな Node が追加されたことを確認する。初めに qx301 を追加した（図 3.9）。 Central Manager はジョブを Worker Node に振り分けることが可能となった。

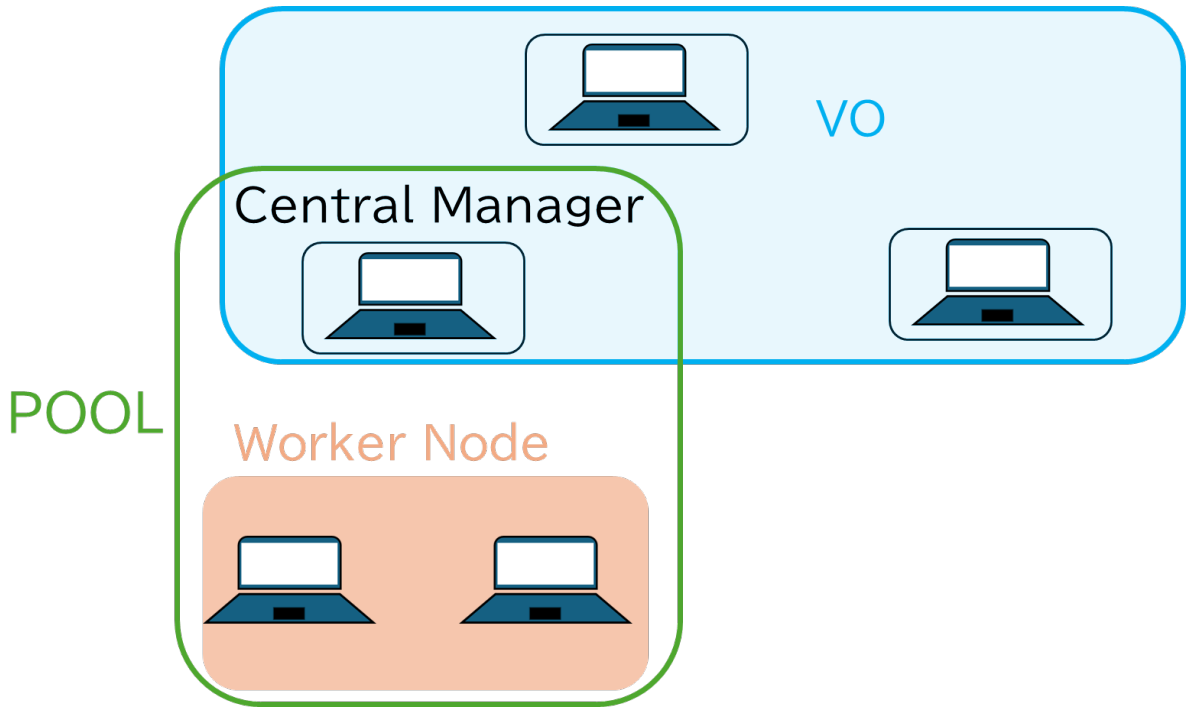


図 3.8: POOL の概念図

```
                     condor_status
Name                                OpSys      Arch   State     Activity LoadAv Mem
A

slot1@grid01                        LINUX      X86_64 Unclaimed Idle      0.000  63781
0
slot1@qx301                         LINUX      X86_64 Unclaimed Idle      0.000 192634
0

               Total Owner Claimed Unclaimed Matched Preempting  Drain Backfill BkIdle

  X86_64/LINUX     2     0       0         2       0          0      0        0      0

         Total     2     0       0         2       0          0      0        0      0
```

図 3.9: qx301 を認識させた様子

HT Condor のホスト構成

Central Manager と Worker Node では使用する HT Condor のプロセスが異なる。ホスト構成の Central Manager ではジョブの割り振りを行う、schedd を用いる。一方、ジョブを実行する Worker

Node では startd を用いる。Worker Node 追加後 (図 3.10) の condor_startd は Cemtral Manager で実行しなくする。

```
pstree | grep condor
|-condor_master-+-condor_collecto
|               |-condor_negotiat
|               |-condor_procd
|               -condor_schedd
|               |-condor_shared_p
|               -condor_startd
```

図 3.10: HT Condor のプロセス

Worker Node の追加が完了し、HT Condor の設定を変更した。Central Manager 構築時は広島サイトが稼働できたか確認するためジョブを自身でも処理できるようにしていた。しかし、Worker Node を構築したため、ジョブを Central Manager が処理する必要はない。また、サイトのリスク管理においてノードごとの役割を分散させたいため、Worker Node でのみタスクを処理するように設定を変更した。grid01 内の HT Condor の設定ファイルである”00-minicondor”を HT Condor のホスト構成用の設定にする。設定を反映させ、認識されているノードが Worker Node のみとなる (図 3.11)。広島サイトは Central Manager ではジョブを実行せず、Worker Node のみにジョブを振り分ける。

```
condor_status
Name                                OpSys      Arch   State     Activity

slot1@qx301                         LINUX      X86_64 Unclaimed Idle
slot1@qx302                         LINUX      X86_64 Unclaimed Idle

               Total Owner Claimed Unclaimed Matched Preempting  Drain Backfill BkIdle

  X86_64/LINUX     2     0       0         2       0          0      0        0      0

         Total     2     0       0         2       0          0      0        0      0
```

図 3.11: 実行ノードの変更

グリッド環境統一のための設定

複数のノードで構成するグリッド環境の場合、すべてのノードで同一の環境とすることが重要となる。ジョブを実行する環境を統一するため、ジョブ実行用のユーザーとグループを作成した。

ユーザーを “alicegm” 、グループを “alice” とした。広島サイトは複数のノードで構成するが、すべてのノードで同一のユーザーとグループを設定することにより、HT Condor でジョブ管理が可能となる。Worker Node にジョブを実行するユーザーを追加した。HT Condor がジョブを行う

“slot”（スロット）ごとに、“slot1_1”、“slot1_2” とユーザーを作成するよう設定を行った。各スロットごとに専用のユーザーで実行し、ジョブの分離、他のスロットへの干渉を防ぐ。

```
condor_status
Name                                OpSys      Arch   State     Activity LoadAv Mem
ActvtyTime

slot1@qx301                         LINUX      X86_64 Unclaimed Idle      0.000   255
4  0+00:11:40
slot1_1@qx301                       LINUX      X86_64 Claimed   Busy      0.000 19008
0  0+00:00:00
slot1@qx302                         LINUX      X86_64 Unclaimed Idle      0.000   255
4  0+00:04:39
slot1_1@qx302                       LINUX      X86_64 Claimed   Busy      0.110 19008
0  0+00:02:15

               Total Owner Claimed Unclaimed Matched Preempting  Drain Backfill BkIdle

  X86_64/LINUX     4     0       2         2       0          0      0        0      0

         Total     4     0       2         2       0          0      0        0      0
```

図 3.12: slot の使用

ジョブ “slot1_1” で実行されていることがわかる。各ジョブは専用のユーザーで実行される。

ジョブ実行用ディレクトリの設定

ジョブを実行する際、HT Condor ではプロセスを分離するためコンテナを用いる。コンピュータの中にジョブ用の環境が整ったコンピュータを用意するようなものである。ジョブの実行には一番ストレージ容量の大きいディレクトリを使用する。

```
df -h
Filesystem                  Size  Used Avail Use% Mounted on
devtmpfs                    4.0M     0  4.0M   0% /dev
tmpfs                        95G     0   95G   0% /dev/shm
tmpfs                        38G   18M   38G   1% /run
efivarfs                    176K   68K  104K  40% /sys/firmware/efi/efivars
/dev/mapper/almalinux-root   70G  3.6G   67G   6% /
/dev/sda2                   960M  282M  679M  30% /boot
/dev/sda1                  1022M   17M 1006M   2% /boot/efi
/dev/mapper/almalinux-home  856G  6.0G  850G   1% /home
tmpfs                        19G     0   19G   0% /run/user/1000
cvmfs2                      4.0G  632M  3.3G  16% /cvmfs/cvmfs-config.cern.ch
cvmfs2                      4.0G  632M  3.3G  16% /cvmfs/alice.cern.ch
```

図 3.13: ディスクの空き容量

Worker Node で一番容量が多いディレクトリをホームディレクトリとしたので (図 3.13)、ホームディレクトリの下にジョブ実行用のディレクトリを作成した。HT Condor の設定ファイルで作成したディレクトリをジョブ実行ディレクトリと指定した。

3.4.2 Worker Node の CVFMS のインストール

続いて CVFMS をインストールした。ジョブの実行にあたり、ALICE 実験用のソフトウェアやライブラリが随時必要となる。CVFMS を Worker Node にインストールすることによりジョブの処理が可能となる。CVFMS の設定ではキャッシュの上限を 5 GB に設定した。キャッシュがメモリを圧迫することを防ぐ。また、ネットワークの効率化のため、grid03 を Central Manager の構築時に squid として利用しているので同様に利用する。

第4章　現状と今後の課題

4.1　現状

4.1.1　Central Manager

グリッドサイトの中枢的な管理ノードとしての役割を持つ、Central Manager の構成は完了した。広島 Tier2 サイトは ALICE 実験のグリッド・コンピューティング拠点として再稼働した。また、Central Manager の起動により、広島サイトの現状は ALIMonitor で確認することが可能となった。今後はサイトに問題があった場合は ALIMonotor で確認できるため、サイトの管理体制も安定した。

図 4.1: 広島サイトの再稼働 [4]

図左側の 2024 年 1 月までは、RUNNING のプロットがあるが、それ以降サイトの停止によりプロットが無い。今回、広島サイトを再構築したことにより 2024 年 12 月から再び RUNNING のプロットがみることができる。

稼働に伴い、広島サイトの運行状況は ALIMonitor で確認が行える。CPU 稼働率やジョブ実行数、合計コア数など様々な情報の確認が行える。

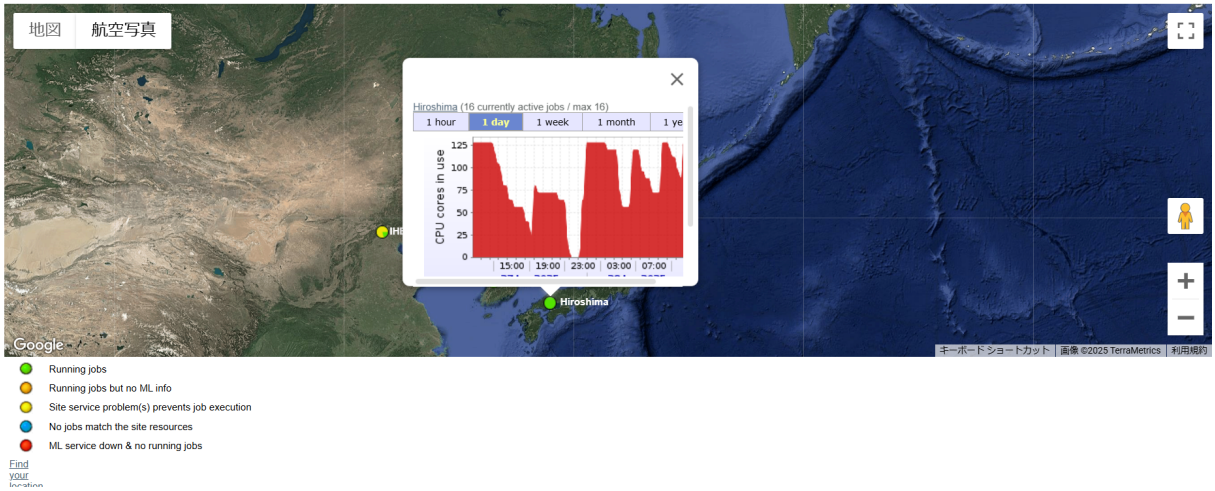
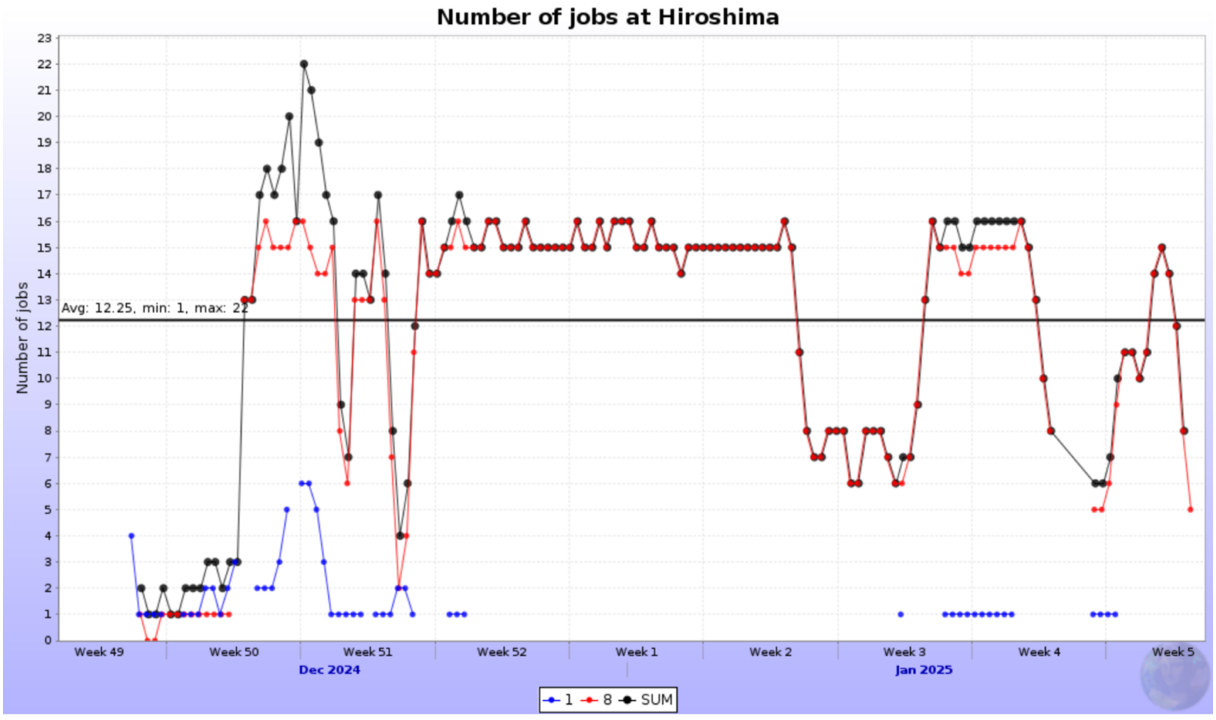


図 4.2: 広島サイトの CPU 使用率 (2025 年 1 月 29 日時点）[5]

ALIMonitor で広島サイトの CPU 使用率を見た。Central Manager の稼働により、ALIMonitor で運行状況を可視的に確認できる。サイトに異常があったり停止した場合はプロットが赤色になる。

4.1.2 Worker Node



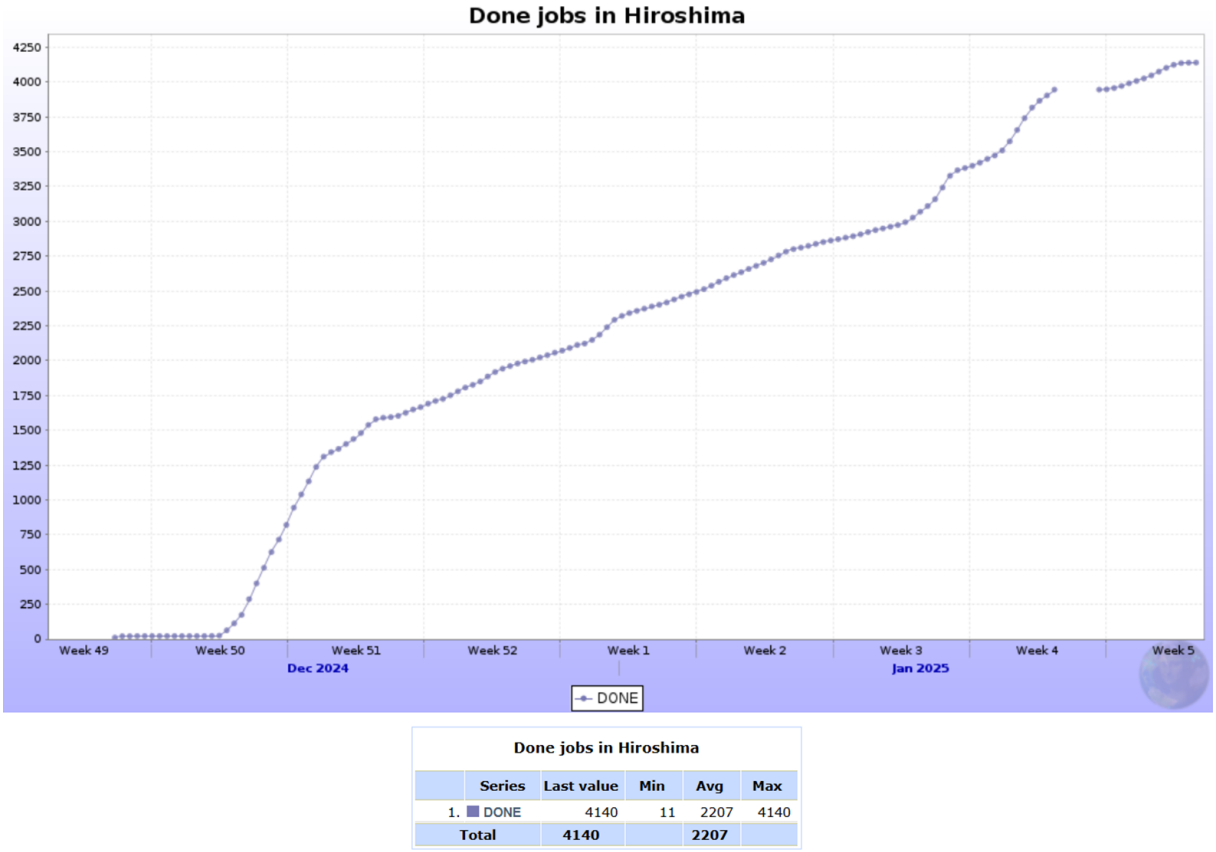
Number of jobs at Hiroshima

	Series	Last value	Min	Avg	Max
1.	1	1	0	1.686	68
2.	8	5	0	11.69	24

図 4.3: ジョブあたりの使用コア数 [4]

ジョブを実行する Worker Node は 2 ノード活動していて、ALICE 実験に 128 コアの計算リソースを提供している。広島サイトでは 1 コア使用するジョブと 8 コア使用するジョブを実行する (図 4.3)。2025 年 1 月 23 日から 26 日まで Central Manager の HT Condor の設定ファイルにエラーが生じ停止したが、復旧させ稼働している。

再稼働から 2025 年 1 月 30 日 15:00 まで、合計 4,140 個のジョブを処理した（図 4.4)。



Done jobs in Hiroshima

	Series	Last value	Min	Avg	Max
1.	DONE	4140	11	2207	4140
Total		4140		2207	

図 4.4: 再稼働後のジョブ実行数 [4]

また、すべての Worker Node の OS を AlmaLinux 9 へ入れ替えを行い、qx305 まではすべてのセットアップを完了した。広島サイトとして qx301 から qx305 までの Worker Node は追加した (図 4.5)。

```
                     condor_status
Name                                OpSys      Arch   State     Activity LoadAv Mem     ActvtyTime

slot1@qx301                         LINUX      X86_64 Unclaimed Idle      0.000 192634 10+04:50:41
slot1@qx302                         LINUX      X86_64 Unclaimed Idle      0.000   2554 48+21:01:00
slot1_1@qx302                       LINUX      X86_64 Claimed   Busy     67.430 190080  0+01:29:01
slot1@qx303                         LINUX      X86_64 Unclaimed Idle      0.000 192634 14+02:45:29
slot1@qx304                         LINUX      X86_64 Unclaimed Idle      0.000 192634  8+23:29:54
slot1@qx305                         LINUX      X86_64 Unclaimed Idle      0.000 192634  6+05:50:21

               Total Owner Claimed Unclaimed Matched Preempting  Drain Backfill BkIdle

 X86_64/LINUX      6     0       1         5       0          0      0        0      0

        Total      6     0       1         5       0          0      0        0      0
```

図 4.5: Worker Node の追加状況

4.1.3 Storage Element

nfs11 と nfs12 の OS を AlmaLinux 9 に入れ替えネットワーク接続も完了している。一方、nfs13 はネットワークスイッチが不足し、セットアップが完了していない。

4.2 今後の課題

4.2.1 Worker Node の増築

qx306 から qx320 の Worker Node を追加し 20 台の Worker Node を持つサイトにする。すべての Worker Node に HT Condor と CVFMS のインストールを行い広島サイトの Worker Node を追加し、現在の 128 コアから 1,280 コアへの計算リソースの拡張を行う。

4.2.2 Storage Node の構築

ネットワークの接続を行い、nfs13 のセットアップを完了する。その後、EOS を用いたストレージシステムを構築する。広島サイトで生成したシミュレーションデータを他のサイトへ送信せずに、サイト内で保存できるようになる。

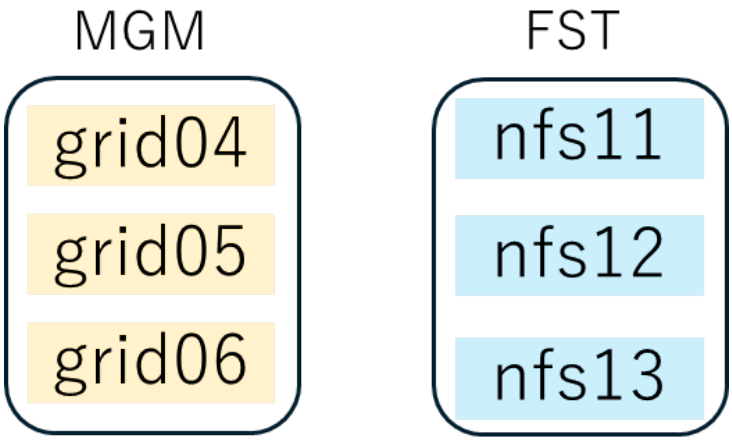


図 4.6: ストレージノードの構成

EOS において、grid04、grid05、grid06 をファイルの管理を行う MGM とし、ストレージ（FST）として nfs11、nfs12、nfs13 を扱う予定である。広島サイトの停止前からストレージ容量を 2 倍へ拡張する。

4.2.3 モニタリングツールの導入

グリッドサイトには安定した稼働が必要となる。広島サイト内のノードを監視するため、サイト内のモニタリングシステムを構築する。各ノードの稼働状況を把握し、問題発生時に対応できるシステムを構築する。

謝辞

志垣賢太教授、卒業論文の構成のアドバイスや添削をしていただきありがとうございました。伝わりやすさを意識して構成を組みなおすことができました。また、ミーティングではさまざまなアドバイスをいただきました。ありがとうございました。

Maarten Litmaath, thank you for teaching me ALICE experiment grid computing from scratch during my stay at CERN for the construction of the Hiroshima site. Without you, I would not have been able to restart the Hiroshima site.

大山健教授、構築にあたりコンピューティングについて教えていただきありがとうございました。また、サイトの構成についてアドバイスをいただきありがとうございました。CERN 滞在時に手続き等のサポートもしていただきありがとうございました。

Latchezar Betev, thank you for providing me with opportunities to communicate with computing experts during my stay at CERN.

Costin Grigoras, thank you for teaching me how to use iLO. Thanks to you, I was able to reinstall the OS from CERN, which allowed the work to proceed smoothly.

Maksim Melnik Storetvedt, thank you for running test jobs during the reactivation of the Hiroshima site.

八野哲助教、KEK からグリッドの Certificate を取ってきてくださりありがとうございます。また、サーバー室の使い方やアドバイスをしていただきありがとうございます。

荻野雅紀さん、再構築前の情報を教えていただきありがとうございました。

三好隆博助教、B4 ゼミでご指導いただきお世話になりました。また、コピー室の鍵を貸していただいたりと気を利かせていただきありがとうございます。

本間謙輔准教授、ラボエクササイズでは実験のやり方を教えていただきありがとうございました。

山口頼人准教授、夜中によく大部屋に入ってきて雑談をしていただきありがとうございました。

小山内博基さん、宮本祥さん、広島サイトの再構築にあたりお世話になりました。ありがとうございました。残りの構築も共に行いましょう。

木村健斗さん、モンテカルロシミュレーションの使い方を教えていただきありがとうございました。

老田将大さん、要旨の添削から ALICE 実験の解析フレームワークについて教えていただきありがとうございます。検出器で取られたデータの流れを教えていただきありがとうございました。他にも多数のアドバイスをしていただきありがとうございました。

山田蓮斗さん、卒業論文の執筆にあたり、章構成から日本語の添削までたくさんアドバイスをいただきありがとうございました。山田さんのアドバイスをもとに図を追加したり、文章の順を変更したりと人に伝わりやすい構成に近づくことができました。また、たくさん雑談をしてくださり適切な息抜きを取ることができました。音楽理論も教えていただきありがとうございました。

和田澪太さん、お菓子の差し入れありがとうございます。卒論のアドバイスもしていただきありがとうございました。ぽん太を紹介していただきありがとうございます。あんなにおいしい居酒屋が近くにあることを知れて大満足です。鶏肉と魚がお気に入りです。卒論シーズンの大根の

おでんは骨の髄まで染み渡りました。また、よく息抜きの雑談をしていただきありがとうございました。

勝野永遠さん、wsl の設定を教えていただきありがとうございました。ノート PC が使いやすくなりました。

中村太陽さん、お菓子の差し入れありがとうございます。ハリボーおいしかったです。

山内航さん、ポケモンカードの新情報を教えていただきありがとうございます。新しい息抜きができました。

西崎君、Typst を教えてくださりありがとうございます。本論文では *LaTeX* を使用しました。

木村君、浜田君、福本君、河本君、寺元君、水野君、愉快な 4 年間をありがとうございました。皆さんと切磋琢磨し単位を取ることができたと思います。たくさんの笑いをありがとうございました。社会人になっても大学院生になっても愉快に過ごしましょう。今後もたくさんの苦難があると思いますが、皆さんの活躍を願っています。

Au personnel de la cafétéria du bâtiment 13, merci de m'avoir servi un délicieux cappuccino et une délicieuse madeleine, ou un délicieux macchiato et une délicieuse madeleine pendant mon séjour au CERN. Grâce à vous, j ' ai pu découvrir à quel point le cappuccino est délicieux. Cela m ' a permis de travailler avec énergie au CERN.

私を産み、育て、支えてくださった家族に、心の底から感謝いたします。楽しく温かい家庭をありがとうございます。実家に帰るたびに元気が戻りました。本当にありがとうございます。最大限の感謝申し上げます。

最後に、
「ありがとうな」
この言葉に尽きる。

参考文献

1. 秋葉康之. クォーク・グルーオン・プラズマの物理. 共立出版, 2014, 184

2. David A. Patterson, Garth Gibson, and Randy H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, vol. 17, no. 3, pp. 109–116, 1988.
https://dl.acm.org/doi/10.1145/50202.50214

3. インテル, "ハイパースレッディングとは？", (Accessed on 02/06/2025)
https://www.intel.co.jp/content/www/jp/ja/gaming/resources/hyper-threading.html

4. ALICE Grid Monitoring with MonALISA, "Active jobs in Hiroshima",(Accessed on 01/27/2025)
http://alimonitor.cern.ch/display

5. ALICE Grid Monitoring with MonALISA, "Grid sites monitoring map",(Accessed on 01/27/2025).
http://alimonitor.cern.ch/map.jsp

6. CERN Accelerating science, "CERN ' s accelerator complex', (Accessed on 01/14/2025)
https://home.web.cern.ch/science/accelerators/accelerator-complex

7. CernVM-FS, "Welcome to CernVM-FS ' s documentation", (Accessed on 02/06/2025)

8. CERN Document Server, "Phase diagram of QCD matter", (Accessed on 02/06/2025)
https://cds.cern.ch/record/2025215

9. HTCondor Manual, "HTCondor Version 24.3.0 Manual"(Accessed on 02/06/2025)
https://htcondor.readthedocs.io/en/latest/

10. LHC ALICE 実験 日本グループ, "LHC 加速器", (Accessed on 01/15/2025)
http://alice-j.org/wp-content/uploads/2018/02/fig6.png

11. LHC ALICE 実験 日本グループ, "ALICE 実験概要", (Accesed on 01/15/2025)
http://alice-j.org/wp-content/uploads/2018/02/ALICE_mini.jpg

12. Squid: Optimising Web Delivery
https://www.squid-cache.org/

13. Supermicro, "Grid Computing", (Accessed on 01/15/2025)
https://www.supermicro.com/en/glossary/grid-computing.

14. TWiki, "WLCG VObox deployment documentation", (Accessed on 02/06/2025)
https://twiki.cern.ch/twiki/bin/view/LCG/WLCGvoboxDeployment

15. Worldwide LHC Computing Grid, "Tier", (Accessed on 02/06/2025)
https://wlcg-public.web.cern.ch/tiers

付 録A　用語集

A.0.1　ノード

ノードはネットワークにつながったコンピュータのことである。

A.0.2　Hyper Threading

1つの物理コアが2つの論理コアとして処理を行い、異なるスレッドを処理する技術である。Hyper Threading はCPUの空き時間を利用し、CPUの処理能力が向上する。広島サイトのWorker Node は Hyper Threading により、スレッド数を2倍にし64 コア分の処理を行う。

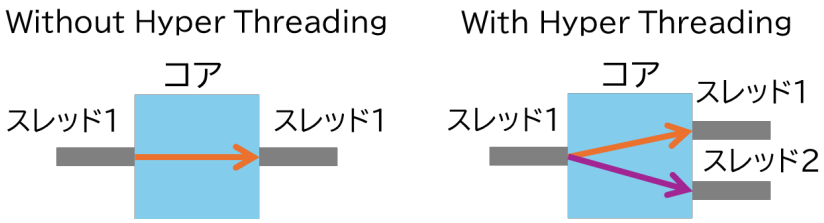


図 A.1: Hyper Threading

A.0.3　RAID

RAID（Redundant Array of Inexpensive Disks）[2] はストレージの性能向上と信頼性するためのディスクアレイ技術である。RAID はパリティと呼ばれる復元データを作り保存する。ディスク損傷時にパリティを用いてデータを復元することができ耐障害性を持つ。nfs11、nfs12、nfs13 ではストレージに RAID 構成を行っている。nfs11 は RAID6 と呼ばれるダブルパリティを用いる RAID 構成をした。RAID6 は2つのディスクが破損した場合までデータの復元が可能である。また、nfs12、nfs13 は RAIDZ3 と呼ばれるトリプルパリティを用いる RAID 構成をした。RAIDZ3 は3つのディスクが破損した場合までデータの復旧が可能である。