

## 修論題名

東京工業大学 理学院物理学系物理学コース 陣内研究室  
奥山広貴 (19M00398)

2020 年 11 月 13 日

# 概要

# 目次

概要	i
第 1 章 LHC-ATLAS 実験と検出器アップグレード計画	1
1.1 LHC について	1
1.2 ATLAS 実験	1
1.3 HL-LHC 実験アップグレード計画	1
第 2 章 新型ピクセルモジュール	2
2.1 シリコン検出器	2
2.2 ピクセルモジュールの構成	2
2.3 モジュールの種類	2
2.4 現行ピクセルモジュールとの違い	2
第 3 章 検出器量産と品質試験	3
3.1 検出器量産	3
第 4 章 モジュール情報及び品質試験結果管理システム	4
4.1 中央データベース	4
4.2 ローカルデータベース	5
4.3 量産時の情報登録・データ同期の流れ	9
4.4 モジュール生産状況の解析	9
第 5 章 品質試験項目:読み出し試験に用いるソフトウェアと学内実験室におけるデモンストレーション	10
5.1 読み出し試験に用いるソフトウェアの概要	10
5.2 読み出し試験結果解析ツールの開発	10
5.3 学内実験室におけるデモンストレーション	10
第 6 章 中央データベースとローカルデータベースのデータ同期ツールに関する研究	14
6.1 サーバーの設置場所による処理時間の違い	14
6.2 モジュール ID のダウンロード機能確認と処理時間測定	15
6.3 読み出し試験結果のアップロード機能確認と処理時間測定	17
第 7 章 まとめ	18

---

7.1	まとめ . . . . .	18
7.2	今後の課題 . . . . .	18
7.3	結論 . . . . .	18
付録 A	ローカルデータベースにおける読み出し試験結果検索システムの性能評価	19
付録 B	読み出し試験に用いたハードウェア詳細	20
参考文献		21
謝辞		22

## 第 1 章

# LHC-ATLAS 実験と検出器アップグレード計画

### 1.1 LHC について

LHC と ATLAS 実験は分けたほうがいい。

### 1.2 ATLAS 実験

#### 1.2.1 内部飛跡検出器

#### 1.2.2 カロリメータ

#### 1.2.3 ミューオン検出器

### 1.3 HL-LHC 実験アップグレード計画

#### 1.3.1 加速器アップグレード

#### 1.3.2 内部飛跡検出器のアップグレード、現行との違い

#### 1.3.3 期待される物理

## 第2章

# 新型ピクセルモジュール

### 2.1 シリコン検出器

#### 2.1.1 半導体

#### 2.1.2 pn 接合

#### 2.1.3 検出原理

#### 2.1.4 放射線損傷

### 2.2 ピクセルモジュールの構成

#### 2.2.1 要求

#### 2.2.2 シリコンセンサー

#### 2.2.3 読み出しフロントエンドチップ

#### 2.2.4 PCB

#### 2.2.5 module carrier

### 2.3 モジュールの種類

### 2.4 現行ピクセルモジュールとの違い

## 第 3 章

# 検出器量産と品質試験

### 3.1 検出器量産

#### 3.1.1 組み立て工程

#### 3.1.2 品質試験

##### 試験項目

電気読み出し試験と pixel analysis それ以外の試験

## 第 4 章

# モジュール情報及び品質試験結果管理システム

前章で述べたように、モジュール生産及び品質試験を世界中で行う。これらの情報はデータベースシステムを用いて管理することが決まっていて、現在この開発を行っている。システムについては、大きく 2 つに分けられる。チェコに設置し、試験運用をしている中央データベースと、各組み立て期間に設置し、運用の際に使用するローカルデータベースである。本章ではこれらのデータベースについて説明する。また、システム開発の中で私が開発を行った仕組みや機能について詳細に説明する。

### 4.1 中央データベース

#### 4.1.1 中央データベースの概要

##### 概要

中央データベースは、新型内部飛跡検出器の製造に関する全ての情報の保存を目的として開発されたデータベースである。ユニコーン大学が開発、運用を行っていて、チェコにデータベースサーバーが設けられている。新型内部飛跡検出器は、前述したようにピクセル検出機とストリップ検出機から構成される。これらを生産するにあたって、シリコンセンサーや電気基板といった小さな部品から製造を行い、それらを用いたモジュールの組み立て、複数モジュールを搭載した stave や ring の組み立てを経て検出器が完成する。また各組み立て段階において、動作確認等を目的とした品質試験を行う。これらの過程における全ての構成部品の情報、及び品質試験結果を中央データベースに保存する。

##### 意義

中央データベースに保存された情報は、検出器運転時の参考値として扱われる。モジュールを例にだすと、品質試験で読み出し試験を行った際の最適な設定値を中央データベースに保存するため、実際の運転時に参照することができる。また運転前の状態における検出器の性能、運転前後での検出機性能比較を行うことができる。HL-LHC では 1 章で述べたように、運転時における放射線量が大きいものとなるため、運転前後での放射線損傷の影響の研究を行うことができ、検出機の寿命の推定や放射線損傷に関しての対策に役に立てることができる。

なんかそれらしいデータかこんな解析に役立てたいみたいなのを考え中です。



FIGURE

図 4.1 中央データベース内でのモジュールの構造

FIGURE

図 4.2 中央データベース内での組み立て工程とテスト項目

FIGURE

図 4.3 ローカルデータベースシステムの概要

### 4.1.2 モジュール情報構造および構成部品との関係の実装

中央データベースにモジュールを登録するためには、1章で述べたように Quad や Triplet といったモジュールの種類、FE chip の種類などのモジュール付属情報、モジュールを構成する部品といったのモジュールの情報構造を決定し、データベース上に定義しておく必要がある。この情報構造をデータベースに実装し、登録できる仕組みを整えた。詳細な構造については図 4.1 に示す。

### 4.1.3 組み立て工程および品質試験の情報形式の実装

モジュールの情報構造の実装に加えて、品質試験の情報を正確に管理するには、モジュール組み立て工程の情報と付随する品質試験の項目をデータベース上に定義する必要がある。これを実装し、テスト結果を適切な組み立て工程へアップロードできる仕組みを整えた。

詳細な構造を図 4.2 に示す。

## 4.2 ローカルデータベース

### 4.2.1 ローカルデータベースの概要と意義

中央データベースでは、前述したようにモジュールの情報のみならず新型内部飛跡検出器に関わるすべての情報を管理する。データベースの機能としては汎用的に使えるようなものになっている。モジュールの組み立て及びその品質試験に関しては3章で述べたように工程が複数に渡り、行う品質試験の数も多い。1つの生産現場で多いところでは数千個のモジュールを作ることになるため、データ管理が簡単にかつ円滑に進むようになっていくのが好ましい。このような理由から、生産現場での生産性、利便性に特化し、円滑な生産をサポートすることを目的としたデータベースシステム(ローカルデータベース)を開発している。システムの概要図を図??に示す。オープンソースのサービスである MongoDB を各生産現場で使い、開発したウェブアプリケーションを併用することでデータ管理や中央データベースとのデータ同期を行うシステムとなっている。

具体的にローカルデータベースは以下のような利点を持つ。

- ローカルにデータベースサーバーを立てるためアクセス速度が早く、円滑にデータ管理を行うことができる。
- モジュールの組み立て工程を管理し、生産者の適切な処理を助ける。

FIGURE

図 4.4 MongoDB の構造

FIGURE

図 4.5 MongoDB の構造

- モジュールに特化したデータ管理、解析を行うことで異常をいち早く検知できる。
- 試験者の情報や試験時間など、テスト結果以外の必要な情報を正確に管理できる。

### 4.2.2 開発状況と今後の方針

現在、以下のような機能を実装している。

今後は他の試験についても機能を拡張していきたい。

### 4.2.3 チュートリアルと普及状況

ローカルデータベースの機能の普及を目的として、2020 年 2 月に CERN 研究所にてシステムのチュートリアルを行った。このチュートリアルは以下のような 2 つのセッションに分けて行った。

- 参加者が実際にサーバーの設定、各ソフトウェアのインストールを行いながら機能を実践するセッション (2 月 3 日から 6 日まで)
- 私が参加者の前で実際に機能を実践し、システムや使い方に対して議論を行うセッション (2 月 7 日)

それぞれのセッションの様子を図 4.2 に示す。数多くの議論を行い、有益なフィードバックを得ることができた。

これを経て現在ローカルデータベースは世界 10 箇所にて導入され、試験運用が開始している。また将来的には全組み立て機関で使うことが決定しており、それに向けたシステム開発、サポートが必要となっている状況である。ローカルデータベースについて、試験運用を行っている機関と導入予定機関の概要を図 4.3 に示す。

### 4.2.4 MongoDB とその内部構造

MongoDB の内部構造について簡単に図 4.4 に示す。

モジュール及び品質試験に関する構造として、図 4.5 にまとめた。

### 4.2.5 中央データベースとの情報同期ツール

生産時にはローカルデータベースと中央データベースにおいて、情報の同期が必要となる。例えば、モジュールの ID や組み立て工程、テスト結果といった情報があげられる。この情報同期のためのインターフェースツールを開発した。主に開発した項目については以下の 2 つである。

- モジュールの ID 及びモジュールを構成する FE chip の ID のダウンロード機能

FIGURE

図 4.6 ダウンロード機能の概要

FIGURE

図 4.7 アップロード機能の概要

FIGURE

図 4.8 モジュールページの例

FIGURE

図 4.9 読み出し試験結果ページの例

- 品質試験中の読み出し試験に関する試験結果アップロード機能

ダウンロード機能についての詳細を図 4.6 にまとめた。中央データベースに登録されているモジュールの ID、FE chip の ID とその数、対応関係をダウンロードしてくるものとなる。

アップロード機能についての詳細を図 4.7 にまとめた。YARR で取得されるデータは FE chip ごとであるため、取得ファイルは FE chip ごとにアップロードすることとしている。

生産の際には、これら機能は後述するウェブアプリケーションを通して使用する。具体的な使用方法については、後の節で述べる。処理速度に関する性能評価を 6 章で述べる。

#### 4.2.6 情報管理アプリケーション

##### 概要、意義

各組み立て機関において、試験者が品質試験結果を閲覧、管理するツールとして、ウェブアプリケーションを開発した。試験結果を迅速に分かりやすく見るシステムを作り、円滑な生産の補助や異常結果の早期発見を目的としている。アプリケーションの機能として、具体的に以下のような項目を実装している。

- 登録モジュール及び品質試験結果の情報閲覧機能
- 登録モジュール及び品質試験結果の検索機能
- 試験者情報の管理及び各種機能
- 中央データベースへアップロードする品質試験結果の登録と組み立て工程の管理機能
- 中央データベースとのデータ同期機能

それぞれの機能の詳細を以下に記す。

##### 登録モジュール及び品質試験結果の情報閲覧

アプリケーションを用いて、登録したモジュール情報やアップロードした品質試験結果を閲覧することができる。モジュールページの例を図 4.8 に、読み出し試験の結果ページの例を図 4.9 に示す。

FIGURE

図 4.10 検索機能の様子

FIGURE

図 4.11 コメント機能

FIGURE

図 4.12 タグ機能

FIGURE

図 4.13 サインオフ機能

### 登録モジュール及び品質試験結果の検索機能

登録モジュールや品質試験結果の一覧ページに検索機能を実装した。確認したいモジュール情報や試験結果を迅速に取得し、閲覧できることを目的としている。検索機能を使用している様子を図 4.10 に示す。一般的なウェブページの検索エンジンのように扱うことができる。

また生産に向けて、検索にかかる処理速度測定を行った。これについては付録 A で詳細を述べる。

### 試験者情報の管理及び各種機能

異常があった際に確認することを目的として、誰が試験を行ったかを記録することが必要である。また、モジュールの登録や中央データベースとのデータ同期など、データベースの機能使用を制限することも必要である。これらを目的として、試験者及びデータベース使用者情報の管理システムを開発、実装した。この機能の概要について図 4.13 に示す。データベースの管理者が試験者を登録することができる。また以下のような機能が登録試験者に制限される。

- 試験結果のアップロード
- 中央データベースへアップロードする品質試験結果の登録と組み立て工程の管理機能
- 中央データベースとのデータ同期機能

上記の他に、登録ユーザは結果に対するコメント機能、任意のタグを貼る機能を使用することができる。各種機能について、使用したときの様子を図 4.11、4.12 に示す。

### 中央データベースへアップロードする品質試験結果の登録と組み立て工程の自動更新

ローカルデータベースへアップロードした品質試験結果の中から、本結果として中央データベースへアップロードする結果を選択する機能を開発した。品質試験は 3 章で述べたように、各モジュール、各組み立て工程に対して行うものであるため、結果選択も同様に工程毎に行うことを想定している。結果選択後、データベースにおける組み立て工程の情報も次のものへ自動的に更新する機能となっている。

あるモジュール、組み立て工程に対して結果を選択する様子を図 4.14 に示す。組み立て工程も自動更新されていることがわかる。

FIGURE

図 4.14 データベースシステム操作の流れ

FIGURE

図 4.15 生産時のモジュール組み立て状況解析の例

#### 中央データベースとのデータ同期機能

モジュールのダウンロード機能、読み出し試験結果のアップロード機能をアプリケーションの中で使うことができる仕組みを実装した。モジュールのトップページからこの操作が可能となっている。

### 4.3 量産時の情報登録・データ同期の流れ

想定している、モジュールに関しての組み立て工程とデータベースシステムでのユーザ操作の流れを図 4.14 に示す。

各ユーザ操作は上述したウェブアプリケーションを用いて行う。最終的には、モジュールの情報及び選択した品質試験の結果が全て中央データベースへ同期されている状態となる。この流れのデモンストレーションを、学内のプロトタイプモジュールを用いて行った。詳細を 5 章で述べる。

### 4.4 モジュール生産状況の解析

上述したデータベースシステムを使って、将来的には世界でモジュール生産がどれだけ進んでいるのかを解析する機能を作ること考えている。全てのモジュールの状況は各生産場所のローカルデータベース上に記録され、組み立て工程ごとに中央データベースへデータ同期する。そのため生産時には、中央データベースで全てのモジュールに関して、現在の組み立て情報を取得でき、世界的な生産状況の解析を行うことができる。想定している解析結果のイメージ図を図 4.15 に示す。

## 第 5 章

# 品質試験項目:読み出し試験に用いるソフトウェアと学内実験室におけるデモンストレーション

### 5.1 読み出し試験に用いるソフトウェアの概要

### 5.2 読み出し試験結果解析ツールの開発

### 5.3 学内実験室におけるデモンストレーション

学内実験室で開発しているソフトウェアを用いて読み出し試験を行い、実際の生産時における流れのデモンストレーションを局所的に行なった。その詳細について以下に示す。

#### 5.3.1 デモンストレーションの流れ

今回のデモンストレーションで確認した機能を以下に示す。

- 中央データベースとローカルデータベースのデータ同期機能 (モジュール ID のダウンロード、試験結果のアップロード)
- 読み出し試験に使う各種機能 (設定ファイル生成、温度、電圧、電流モニタリング、試験結果閲覧)
- 結果選択とピクセル解析機能

またデモンストレーションにおける流れの概要を図 5.1 に示す。

#### 5.3.2 読み出し試験セットアップ

読み出し試験に用いるハードウェアのセットアップを表 5.1、概要を図 5.2、各ハードウェアの写真を 5.3 に示す。各装置の詳細については付録 B に示す。

FIGURE

図 5.1 デモンストレーションの流れ

表 5.1 各ハードウェアの性能

1	2
result 1	result 2

FIGURE

図 5.2 ハードウェアセットアップの概要

FIGURE

図 5.3 各ハードウェアの写真

FIGURE

図 5.4 ダウンロードしたモジュール ID 確認画面

FIGURE

図 5.5 設定ファイル生成のイメージ

FIGURE

図 5.6 生成ファイル確認画面

5.3.3 読み出し試験内容

読み出し試験を通して、モジュールに与える電圧値、電流値、チップ横についている NTC から読み取れる温度を記録した。以下の流れに沿って読み出しを行なった。

5.3.4 機能確認

モジュール ID のダウンロード

登録したモジュールの ID を機能を使ってダウンロードし、ウェブアプリケーションで確認した。確認した画面を図 5.4 に示す。

読み出し試験

- 以下の流れで読み出し試験を行なった。読み出し試験はサーバーのシェルを用いて行う。
- ・ 設定ファイル生成
- ダウンロードしたモジュールの ID を用いて、読み出しに用いる設定ファイルを生成した。イメージを図 5.5、実際に生成したファイルを確認した画面を図 5.6 に示す。
- ・ 試験実施とアップロード
- 上述した流れに沿って読み出し試験を実施した。試験結果は各試験の終わりに自動的にアップロードされるようなシステムとなっている。
- ・ 電圧値、電流値、温度のモニタリング
- 記録した値を Grafana を使ってモニタリングをした。その様子を図 5.7 に示す。

FIGURE

図 5.7 DCS のモニタリング

FIGURE

図 5.8 試験結果の閲覧

FIGURE

図 5.9 各測定値の閲覧

FIGURE

図 5.10 各 scan 結果

FIGURE

図 5.11 scan 結果選択の様子

表 5.2 ピクセル解析結果

1	2
result 1	result 2

表 5.3 scan file の存在確認

1	2
result 1	result 2

・試験結果の閲覧

ウェブアプリケーションを用いて、試験結果を閲覧した。その様子を図 5.8、5.9 に示す。  
各 scan の結果を 5.10 に示す。

結果選択とピクセル解析

読み出し結果を選択し、ピクセル解析を行なった。結果選択の様子を図 5.11、解析結果を表 5.2 に示す。

試験結果アップロード

選択した結果を中央データベースにアップロードし、各ファイルが正しくアップロードされていることを確認した。各ファイルの存在を確認した結果を表 5.3 に示す。

今後の開発方針

生産の本番に向けて、中央データベースにあげるデータを再検討する必要がある。現在のシステムでは 5 つの試験に対して結果ファイルだけをあげるものとなっている。情報としては不十分であり、以下のよ  
うな項目を検討しアップロード情報として付け加えることが必要であり、今後の課題となっている。

- DCS の情報



- 適切な読み出し試験項目
- ピクセル解析の結果

## 第 6 章

# 中央データベースとローカルデータベースのデータ同期ツールに関する研究

開発した機能が生産時に十分であるかどうか見積もりは今後の開発を効率よく進めていく上で重要である。生産時のデータの数や量を想定してその際のシステム性能を見積もることで、今の実装で十分かどうか、改善が必要な場合どのように改善すればいいかを知ることができる。中央データベースとローカルデータベースのデータ同期機能に関して処理時間評価を行った。詳細について以下で説明する。

### 6.1 サーバーの設置場所による処理時間の違い

4 章で述べたように、中央データベースはチェコに設置されている。そのため試験結果のアップロードに関して、各組み立て機関から接続しデータ送信する処理時間は、機関の場所に大きく依存すると考えられる。世界的にデータ同期ツールが不自由なく動くことに向けた開発、改善に役立てることを目的として、データを送信する処理時間を、以下の 3 つの場所に置かれているサーバーを用いて測定した。

- 日本、高エネルギー加速器研究所 (KEK)
- アメリカ、バークレー研究所 (LBL)
- スイス、欧州原子核研究機構 (CERN)

各サーバーの性能を表 6.1 に示す。また各サーバーが置かれている場所の位置関係を図 6.1 に示す。

これらのサーバーは実際に生産の際に使用するものと同程度の性能を持ち、サーバーが置かれている環境も生産時と同じであるとしている。回線の混雑具合などによる処理時間の低下は、本測定では考慮に入

表 6.1 サーバーの性能一覧

設置機関	CPU				Memory [kB]	Disk [GB]
	Type	Core	Thread	Clock speed[GHz]		
KEK	Intel(R) Core(TM) i7-6700	4	8	3.4	15,981,000	197
LBL	Intel(R) Core(TM) i7-8700	6	12	3.7	32,628,000	233
CERN	Intel Core Processor (Broadwell, IBRS)	1	10	2.2	29,978,888	80

FIGURE

図 6.1 各サーバの位置関係

表 6.2 データ同期ツールの中で使用した API

関数名	処理の内容	本ツールでの使用用途
getComponent	登録した装置情報の取得	主にダウンロード時におけるモジュールやチップの情報取得に用いる。
uploadTestRunResults	テスト結果生成	読み出し試験結果生成の際に用いる。
createTestRunAttachment	あるテスト結果に対するバイナリファイルの添付	読み出し試験結果生成後にファイルを添付する際に用いる。

表 6.3 モジュール情報の取得にかかる時間

サーバー	処理時間
KEK	0.49 ± 0.02
LBL	0.37 ± 0.02
CERN	0.30 ± 0.04

表 6.4 1Byte のデータファイル添付にかかる処理時間

サーバー	処理時間
KEK	0.54 ± 0.04
LBL	0.34 ± 0.03
CERN	0.39 ± 0.02

れていない。

### 6.1.1 データ同期ツールに使用する API

中央データベースのデータ取得には、用意されている API のサービスを使用している。ローカルデータベースとのデータ同期ツールの中で主に使用している API を表 6.2 に示す。

### 6.1.2 API 使用にかかる時間

上述した API 使用時の処理時間を各サーバーで測定した。以下の 3 つの測定を行なった。

- getComponent を用いた、登録モジュール情報 1 つの取得時間測定
- createTestRunAttachment を用いて、ある試験結果ページに 1Byte のデータファイルを添付する時間測定
- createTestRunAttachment を用いて、ある試験結果ページに容量の異なるデータファイルを添付、容量に対する時間依存性を測定

最初の 2 項目に関して、各処理時間についてまとめたものを表 6.3、6.4 に示す。またファイル容量と処理時間の関係を図 6.2 に示す。1Byte での測定点も含んでいる。

## 6.2 モジュール ID のダウンロード機能確認と処理時間測定

### 6.2.1 アルゴリズム

開発したモジュール ID をダウンロードするツールのアルゴリズムについて記す。アルゴリズムのイメージを図 6.5 に示す。

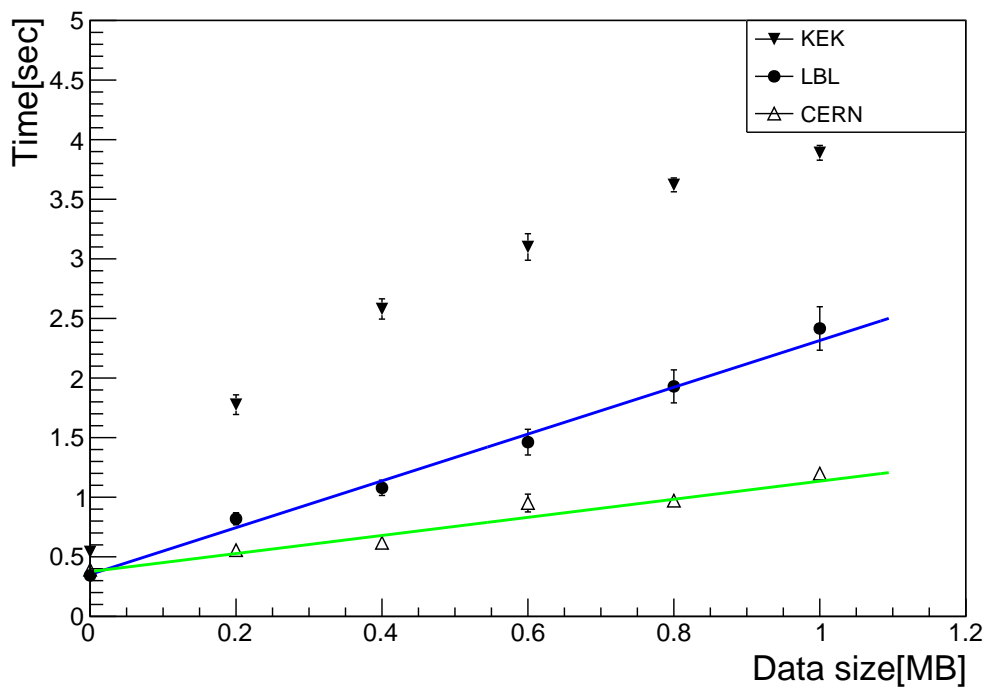


図 6.2 添付するファイルサイズと処理時間の関係

FIGURE

図 6.3 ダウンロードアルゴリズムのイメージ図

表 6.5 登録した KEK モジュール

1	2
result 1	result 2

FIGURE

図 6.4 ダウンロードした KEK モジュールがアプリケーションで確認できている様子

### 6.2.2 機能確認

KEK で組み立てられた 6 台のモジュールを中央データベースに登録し、ダウンロードを行った。登録したモジュールを表 6.5、ダウンロードをしてアプリケーションで確認した様子を図 6.4 に示す。

### 6.2.3 処理時間測定

KEK モジュールをダウンロードした際の処理時間を測定した。これについてまとめたものを表 6.7 に示す。

表 6.6 ダウンロード処理時間測定

1	2
result 1	result 2

FIGURE

図 6.5 アップロードアルゴリズムのイメージ図

表 6.7 アップロード処理時間測定

1	2
result 1	result 2

6.2.4 改善点

6.3 読み出し試験結果のアップロード機能確認と処理時間測定

6.3.1 アルゴリズム

読み出し試験の結果をアップロードするツールのアルゴリズムについて記す。アルゴリズムのイメージを図??に示す。

6.3.2 機能確認

5 章でアップロード機能については確認したため、ここでは割愛する。

6.3.3 処理時間測定

5 章で行った読み出し試験の結果を中央データベースにアップロードした際の処理時間測定を行った。これについてまとめたものを表??に示す。

6.3.4 改善点

構造を変えることかなあ。

## 第 7 章

# まとめ

### 7.1 まとめ

### 7.2 今後の課題

### 7.3 結論

## 付録 A

# ローカルデータベースにおける読み出し 試験結果検索システムの性能評価

## 付録 B

# 読み出し試験に用いたハードウェア詳細

付録がいる場合はどうぞ。



## 参考文献

[テキスト] 東京工業大学理学部物理学科『物理実験学第一』（2009）

## 謝辭