

HL-LHC ATLAS ピクセル検出器量産時の品質試験に向けた データベースシステムの構築

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)

2021 年 1 月 24 日

Abstract

The ATLAS Experiment is being conducted with the Large Hadron Colider at CERN. Its purposes are measurement of the Standard Model (SM) and searches for particles beyond the SM.

In order to acquire more statistics and to achieve more advanced measurements and searches, LHC is plannig to increase the luminosity, referred to as HL-LHC. The target luminosity and integrated luminosity is approximately seven times and ten times higher respectively.

Due to the upgrade, it is required for detectors to have more radiation tolerance and high granularity. It is planed to replace the ATLAS inner detector to the new one, referred to as the Inner Tracker(ITk). ITk entirely consists of silicon detectors and covers much more wider solid angle acceptance than the current Inner Detector.

For the production of ITk, we are planning to produce $O(10,000)$ modules and conduct a series of tests for quality control(QC tests) for individual modules. Those are carried out repeatedly in the production flow. All the QC tests should be stored to a central database, which is set up at Unicorn university in Czech Republic, to record the performance of modules itself.

“Local database” system have been developed in a previous study in order to manage data at local production sites and to synchronize informations to the central DB. The system have been under test-use among production sites towards full production. There were items left to be developed for the module production. Particularly the development of the funcions specialized for the module production and QC tests and the synchronizing tools between the central database and local database.

I have developed the database system in this study. I have implemented key functions for the production, for example searching results, synchronizing data between the local and the central database, and validated that we can use the whole functionalities of the system, including my developed tools, using devices at the laboratory.

Additionally I confirmed that we can use the tools with the actual data at the production to measure the processing time of the services. Concerning a function for searching results, the estimated processing time is $2.6 \pm 0.1[\text{sec}]$ for 84,000 results. Concerning a function for synchronizing data between databases, I have implemented two options, downloading module information and uploading electrical readout results. The processing time for the download is $4.0 \pm 0.4[\text{sec}]$ per quad module. The time for the upload is $1.2 \pm 0.1[\text{min}]$ for 5 readout items of quad modules.

概要

欧州原子力研究機構 (CERN) に設置されている大型ハドロン衝突型加速器 (LHC) の 1 つの衝突点にて、ATLAS 実験が行われている。この実験は現在、素粒子物理学の基本理論となっている標準模型の精密測定や標準模型を超えた新粒子の探索を目的とし稼働している。

更なる測定、探索に向けて取得統計数の増加を狙い、LHC では 2025 年より加速器をアップグレードし、ルミノシティをあげる計画を予定しており、これを HL-LHC と呼ぶ。ルミノシティは現在の LHC の約 7 倍、積分ルミノシティは約 10 倍となる予定である。

HL-LHC において、検出器には高い放射線耐性や位置分解能の向上など、現在のものよりも高い水準が要求される。そのため、ATLAS 実験では最内層に設置している内部飛跡検出器の総入れ替えを予定しており、新しく製造する検出器を Inner Tracker (ITk) と呼ぶ。ITk では全ての領域でシリコン検出器が搭載され、また現在の内部飛跡検出器よりも広い立体角をカバーする設計となっている。

ITk の製造に向けてピクセルモジュール約 10,000 台を生産し、各モジュールに対して品質試験を行う予定となっている。品質試験は項目が多く、モジュール組み立て工程の中で何度も行うものである。モジュール情報及び品質試験の結果は固体性能の保持を目的としてチェコに設置されている中央データベースに保存する必要がある。

また各組み立て機関においてモジュール及び品質試験の情報管理に使用することを目的とした「ローカルデータベースシステム」が先行研究で開発されており、いくつかの機関で試験運用が行われている。このシステムには、モジュール生産時における運用に向けて必要不可欠である開発課題が残されていた。特に品質試験に特化した機能の開発や中央データベースとローカルデータベース間の同期機能開発は、生産時には必要となる機能であるが実装されていなかった。

本研究では、モジュール組み立てとその品質試験に向けたデータベースシステム構築を行なった。ローカルデータベースにおける品質試験管理機能の 1 つとして結果検索機能や、データベース間の同期機能を開発し、システムの拡張を行なった。また品質試験におけるデータベース操作の流れを確立し、データベースの機能が一連の流れの中で使用可能であることを確認した。

生産時に想定されるデータを用いて開発機能の処理時間測定を行い、その有用性を評価した。試験結果検索に関する処理時間は、84,000 の結果数に対して $2.6 \pm 0.1[\text{sec}]$ となった。データベース間の同期機能にして、モジュールのダウンロード機能と読み出し試験結果のアップロード機能を実装した。モジュールのダウンロード機能に関して、Quad モジュールのダウンロードに要する処理時間は $4.0 \pm 0.4[\text{sec}]$ であった。読み出し試験結果アップロードに関して、Quad モジュールの読み出し試験 5 項目のアップロードに要する処理時間は $1.2 \pm 0.1[\text{min}]$ であった。この時、結果ページに添付したファイルの容量は 3.9[MB] であった。

目次

概要	i
第 1 章 序論	1
1.1 素粒子標準模型とその問題点	1
1.2 LHC について	5
1.3 ATLAS 実験	5
1.4 HL-LHC 実験アップグレード計画	10
第 2 章 ピクセルモジュール	16
2.1 ピクセルモジュールの構成	16
2.2 ピクセルモジュールの構成部品	16
2.3 新型モジュールの種類	20
第 3 章 モジュール組み立てと品質試験	21
3.1 組み立て工程	21
3.2 品質試験	22
3.3 検出器量産におけるデータ管理	33
第 4 章 モジュール情報及び品質試験結果管理システム	34
4.1 中央データベース	35
4.2 ローカルデータベース	35
4.3 本研究における開発項目	42
第 5 章 品質試験のデモンストレーション	57
5.1 使用したソフトウェア	57
5.2 使用したハードウェア	59
5.3 デモンストレーションの流れ	62
5.4 機能確認	63
5.5 本章のまとめ	67
第 6 章 ローカルデータベースにおける検索機能	72
6.1 実装方法	72
6.2 処理時間測定	78
6.3 改善方法と処理時間測定	81

6.4	本章のまとめ	85
第 7 章	中央データベースとローカルデータベースの同期	86
7.1	サーバーの設置場所による処理時間の違い	86
7.2	モジュール情報のダウンロード	90
7.3	読み出し試験結果のアップロード	98
7.4	本章のまとめ	105
第 8 章	まとめ	106
8.1	本論文のまとめ	106
8.2	現状と今後の課題	107
付録 A	シリコン検出器の原理	109
A.1	半導体 [15]	109
A.2	検出原理	111
付録 B	RD53A とフレキシブル基板	112
B.1	アナログ回路とデジタル回路	112
B.2	試験用電荷入射のイメージ	112
B.3	RD53A のデータフォーマット	112
B.4	フレキシブル基板	115
付録 C	ローカルデータベースのチュートリアルと普及状況	116
付録 D	モジュール生産状況の解析	118
付録 E	ファイル送信時におけるデータ容量と処理時間の考察	119
参考文献		122
謝辞		126

1 第1章

2 序論

3 欧州原子力研究機構 (**CERN**) に設置されている大型ハドロン衝突型加速器 (**LHC**) では、現在、素
 4 粒子物理学の基礎となっている標準模型の精密測定や標準模型を超える物理現象の探索が行われている。
 5 ATLAS 実験は LHC 上にある 4 つの衝突点の 1 つで行われている実験であり、ATLAS 検出器を用いて
 6 崩壊粒子の測定が行われている。LHC では加速器のアップグレード (**HL-LHC**) を予定しており、これ
 7 に向けて ATLAS 検出器のアップグレードを行う。この章では LHC-ATLAS 実験とそのアップグレード
 8 計画について説明する。

9 1.1 素粒子標準模型とその問題点

10 1.1.1 素粒子標準模型

11 現在素粒子物理学で基礎となっており、多くの実験事実を説明している理論を「標準模型」[1] と呼ぶ。
 12 標準模型はスピン 1/2 のフェルミオンである 6 種類の「クォーク」と 6 種類の「レプトン」、スピン 1 の
 13 4 種類の「ゲージボソン」及びスピン 0 の「ヒッグスボソン」から成る。表 1.1、1.2 に一覧を示す。2012
 14 年 7 月に CERN で新粒子の発見が発表され、これがヒッグス粒子であると認定され標準模型における全
 15 ての素粒子が実験的に確認された。

16 自然界には電磁相互作用、強い相互作用、弱い相互作用、重力相互作用の 4 種類の相互作用がある。標
 17 準模型は強い相互作用を記述する量子色力学 (Quantum ChromoDynamics: QCD)、電磁相互作用と弱
 18 い相互作用を統一した電弱統一理論 (Glashow-Weinberg-Salam: GWS) から成る。重力相互作用は標準
 19 理論では扱っていない。強い相互作用は色荷 (Color) を持ち、 $SU(3)_C$ の対称性を持つ。8 種類の g に
 20 よって力が媒介される。電弱相互作用は $SU(2)_L \otimes U(1)_Y$ の対称性を持つ。力を媒介する粒子として W_1
 21 、 W_2 、 W_3 と B が存在する。このうち W_1 、 W_2 が混合して W^\pm 、 W_3 、 B が混合して Z 、 γ となる。

22 1.1.2 ヒッグス機構と結合定数

23 ここではヒッグス機構と結合定数について説明する [13]。標準模型のラグランジアンに質量項を導入す
 24 ると、ゲージ不変性を破ってしまう。これは素粒子が実際に質量を持つ事実と矛盾する。ここで標準模型
 25 がゲージ不変性を保ちながら素粒子が質量を持つために、標準模型のラグランジアンにヒッグスセクター
 26 を導入する。ここで導入するヒッグスセクターは複素スカラー場 (ヒッグス場) ϕ を用いて以下とする。

$$\mathcal{L}_{\text{Higgs}} = (\partial_\mu \phi)^\dagger (\partial^\mu \phi) - V(\phi) \quad (1.1)$$

$$(1.2)$$

表 1.1 標準模型のフェルミオン。

世代	クォーク	電荷 [e]	質量 [MeV/c ²]	レプトン	電荷 [e]	質量 [MeV/c ²]
1	<i>u</i>	2/3	~2.3	<i>e</i> ⁻	-1	0.511
	<i>d</i>	-1/3	~4.8	ν_e	0	< 2.2 × 10 ⁻⁶
2	<i>c</i>	2/3	~1280	μ^-	-1	105.7
	<i>s</i>	-1/3	~95	ν_μ	0	< 0.19
3	<i>t</i>	2/3	~1.73 × 10 ⁵	τ^-	-1	1777
	<i>b</i>	-1/3	~4200	ν_τ	0	< 18

表 1.2 標準模型のボソン。

記号	名称	相互作用	スピン	電荷 [e]	質量 [GeV/c ²]
γ	光子	電磁	1	0	0
W^\pm	荷電弱ボソン	弱	1	± 1	80.4
Z	中性弱ボソン	弱	1	0	91.2
g	グルーオン	強	1	0	0
H	ヒッグスボソン	-	0	0	125

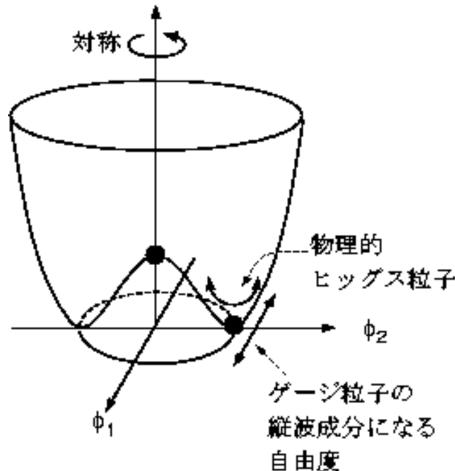


図 1.1 ヒッグスボテンシャル [9]。

27 ここでヒッグスボテンシャル $V(\phi)$ は、ゲージ対称性とくりこみ可能性の要請の下で、以下のように
28 する。

$$V(\phi) = \mu^2 \phi^\dagger \phi + \lambda (\phi^\dagger \phi)^2 \quad (1.3)$$

29 $\mu^2 < 0$ のとき、ボテンシャルが図 1.1 のような形となる。

30 ボテンシャルが最小となる(真空)のは原点ではなく、図におけるリング状の領域である。最小値となる点の中からある1つが選ばれ、その点が現在の真空となっている。これを「自発的対称性の破れ」と呼ぶ。ゲージ粒子の質量は真空点の縦波成分となって現れ、これがヒッグス機構においてゲージ粒子が質量を獲得する原理となる。

34 ここで導入したヒッグスボテンシャル $V(\phi)$ の特徴として、素粒子の質量とヒッグス粒子との結合定数
35 が比例する。真空期待値を $\langle \phi \rangle = v/\sqrt{2}$ とおくと、 W^\pm 、 Z 、フェルミオンとヒッグス粒子の結合の強さ

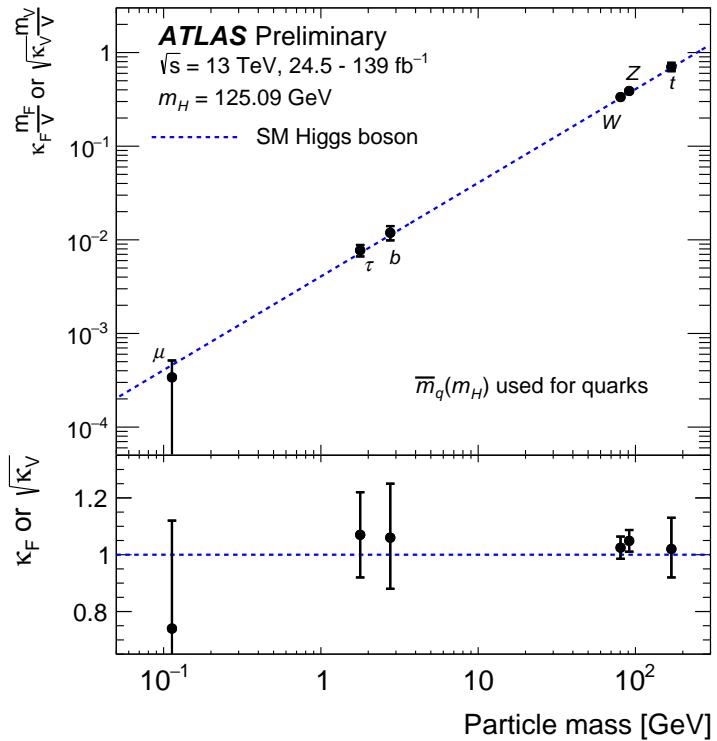


図 1.2 標準模型の素粒子の質量とヒッグス粒子との結合定数の関係 [4]。横軸は粒子の質量、縦軸は各粒子のヒッグス粒子との結合定数を示し、結合の強さを表す。各粒子の測定点が一直線状にのっており、線形性を示していることが分かる。

³⁶ は以下のようになる。

$$W^\pm : \frac{g^2}{2} v = \frac{e}{\sin\theta_W} m_W \quad (1.4)$$

$$Z : \frac{(g')^2}{4} v = \frac{2e}{\sin(2\theta_W)} m_Z \quad (1.5)$$

$$\text{フェルミオン} : Y_f = \frac{\sqrt{2}}{v} m_f \quad (1.6)$$

³⁷ ここで e は素電荷、 θ_W はワインバーグ角と呼ばれる。これらより、素粒子の質量とヒッグス粒子との結合定数が比例することが分かる。

³⁹ ⁴⁰ これらの結合定数は多くの実験で測定されており、線形性が見えている（図 1.2）。このヒッグスセクターの導入及び現在の標準模型は、実験事実と大きく矛盾していないことが分かる。

⁴¹ ⁴² ⁴³ ⁴⁴ 一方、ここで述べた標準模型におけるヒッグスセクターの導入は暫定的なものであり、任意性を持つ。例えば、現在は複素スカラー場 1 つを導入しているが、複数の場を仮定することも可能である。ヒッグス粒子の性質の精密測定によりこのヒッグスセクターの構造にせまることで、標準模型の検証をすることができる。

⁴⁵ ⁴⁶ ⁴⁷ また標準模型を超える新理論ではしばしばこのヒッグスセクターの拡張を行い、標準理論における諸問題の説明を行っている。この時、標準理論とは異なるヒッグス粒子の性質を導く場合がある。ヒッグス粒子の精密測定は、標準模型を超える新理論にも影響を与える。

48 1.1.3 問題点と新理論

49 標準模型は、これまでの実験事実では説明できない以下の問題点を抱えている。

- 50 ● 重力理論との統一.
- 51 ● なぜ素粒子は 3 世代なのか.
- 52 ● 階層性問題.
- 53 ● 強い相互作用における CP 問題.
- 54 ● 暗黒物質, エネルギー.
- 55 ● ニュートリノの質量.
- 56 ● バリオン数の非対称性.

57 現在上にあげた標準模型の諸問題を解くために以下のような理論が考案されている。

- 58 ● 超対称性 [5].
- 59 ● 余剰次元.
- 60 ● テクニカラー.
- 61 ● プレオン模型.
- 62 ● アクション.
- 63 ● シーソー機構.

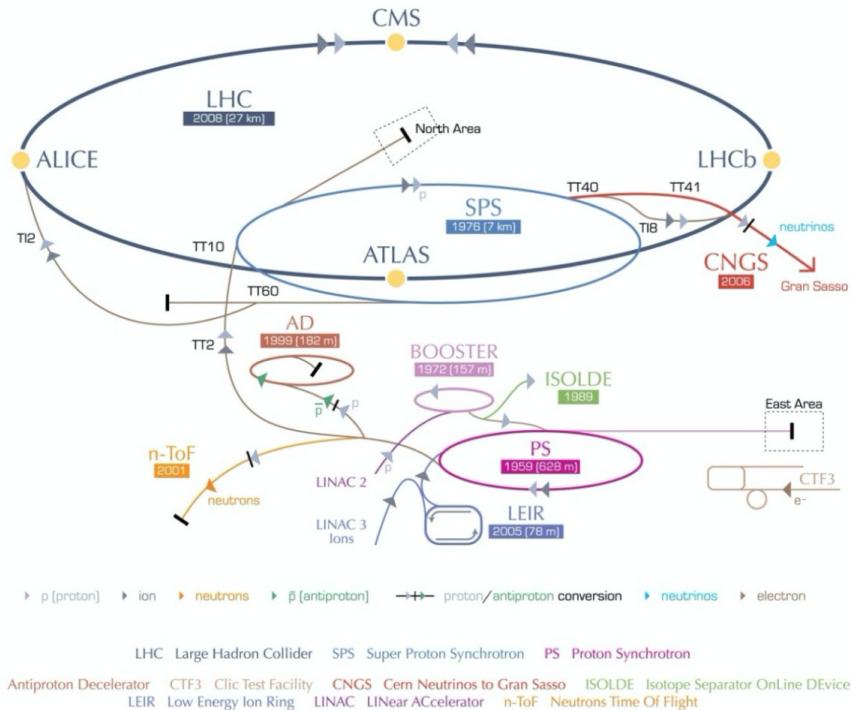


図 1.3 加速器の全体像 [6]。図は CERN に設置されている加速器の全体像を示す。陽子はいくつかの前段加速器で段階的に加速され LHC に入射する。LHC 上には 4 つの衝突点が存在し、それぞれ ALICE、LHCb、CMS、ATLAS 実験が行われている。

1.2 LHC について

LHC は CERN の地下およそ 100 m に設置されている周長 26.7 km の大型ハドロン衝突型加速器である。バンチと呼ばれる陽子のかたまりを 7 TeV まで加速し、衝突させる。世界最大エネルギーの加速器である。

陽子ビームの加速は 4 つの前段加速器を用いて行う。始めに水素ガス中の水素原子から電子を分離することで陽子を生成する。その後最初の線形加速器 (Linear Accelerator: LINAC)、陽子シンクロトロンブースター (Proton Synchrotron Booster: PSB)、陽子シンクロトロン (Proton Synchrotron: PS)、スーパー陽子シンクロトロン (Super Proton Synchrotron) で加速されたのち LHC に入射する。CERN にある加速器の概要を図 1.3 に示す。LHC には 4 つの衝突点があり、それぞれ ALICE(A Large Ion Collider Experiment)、LHCb、CMS(Compact Muon Solenoid)、ATLAS(A Troidal LHC Apparatus) 実験が行われている。それぞれの衝突点には崩壊粒子の飛跡やエネルギーを測定するための検出器が設置されており、取得したデータを元に多様な物理解析が行われている。

1.3 ATLAS 実験

初めに ATLAS 実験に用いる座標系と用語について説明する。まず衝突点を原点として定義しており、ビーム軸を z 軸、これに対して垂直な平面を $x-y$ 平面とする。 z 軸の正方向を Side-A、負方向を Side-C と呼ぶ。 x 軸方向は原点からみて LHC リングの中心に向かう方向であり、 y 軸は上に向かう方向である。方位角 ϕ は z 軸周りの角度であり、極角 θ は z 軸とのなす角である。ATLAS 実験では、極角 θ は以下

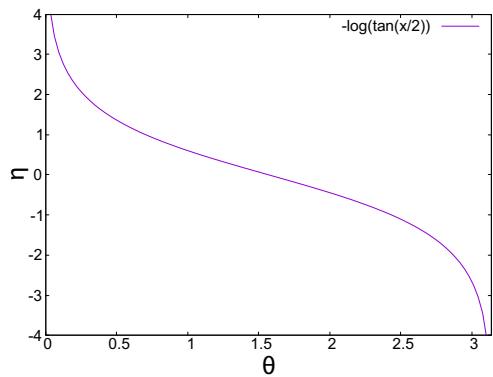


図 1.4 極角 θ と擬ラピディティ η の関係図。横軸は式 1.7 における極角 $\theta(0 \leq \theta \leq \pi)$ 、縦軸は擬ラピディティ η を表している。図のように、 η への変換により定義域が実数全体に広がる。

81 のように擬ラピディティ η で表される。また極角 θ と擬ラピディティ η の関係図を図 1.4 に示す。

$$\eta = -\ln \left(\tan \left(\frac{\theta}{2} \right) \right) \quad (1.7)$$

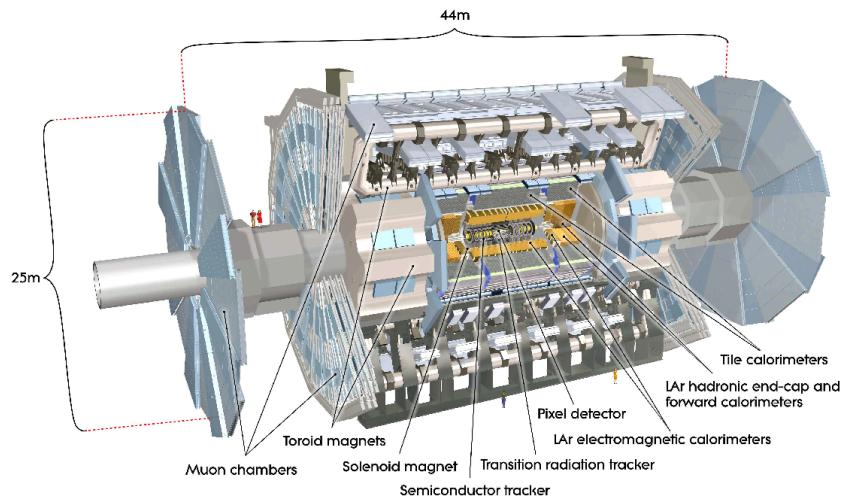


図 1.5 ATLAS 検出器の全体像 [7]。内側から内部飛跡検出器、ソレノイド磁石、カロリメータ、トロイド磁石、ミューオン検出器が設置されている。

1.3.1 ATLAS 検出器

ATLAS 検出器は複数の検出器から構成され、陽子の衝突によって生成された粒子の運動量、エネルギーを測定することができる。最内層に内部飛跡検出器が設置されていて、次に超電導ソレノイド磁石、カロリメータ、トロイド磁石、ミューオン検出器の順に設置されている。衝突点から見た立体角のほとんどを検出器で覆うような設計となっている。ATLAS 検出器の全体図を図 1.5 に示す。

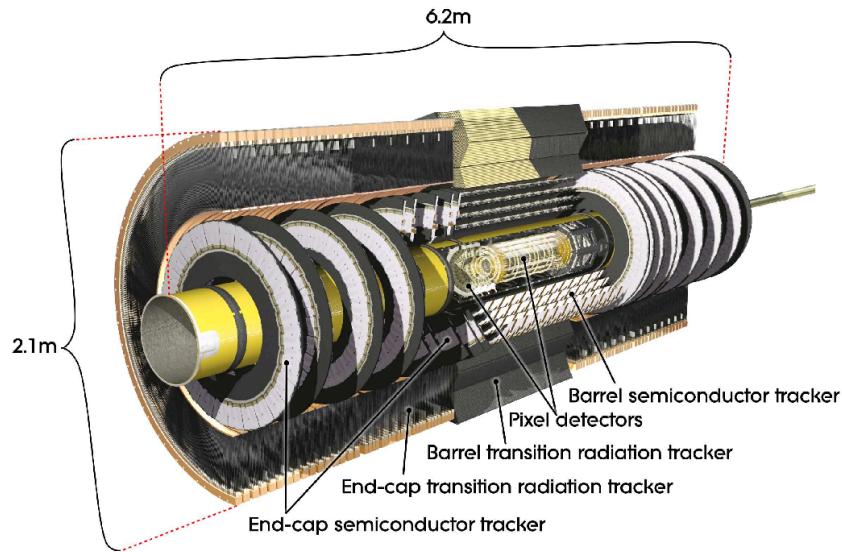


図 1.6 内部飛跡検出器の全体像 [7]。内側からピクセル検出器、ストリップ検出器、遷移放射検出器が設置されている。また円筒状のバレル部とディスク上のエンドキャップ部を構造として持つ。

87 1.3.2 内部飛跡検出器

88 ATLAS 検出器の最内層に位置する検出器である。内部飛跡検出器は粒子の飛跡測定をするための検出
89 器であり、粒子の運動量や衝突点、崩壊点を計算する。検出器の外側には超伝導ソレノイド磁石が設置さ
90 れており、2 T の磁場が z 方向にかけられる。これにより荷電粒子はローレンツ力を受け、軌跡が曲がる。
91 内部飛跡検出器は 3 つの検出器で構成され、内側からピクセル検出器、ストリップ検出器、遷移放射検
92 出器の順に設置されている。ピクセル、ストリップ検出器は階層構造になっており、粒子は複数の検出器
93 を通過する。それぞれの検出器で取得した通過位置をつなぎ合わせることで粒子の飛跡を計算するこ
94 ができる。

95 内部飛跡検出器の全体図を図 1.6 に示す。

96 ピクセル検出器

97 内部飛跡検出器の最内層に位置する検出器である。ピクセル検出器はバレル部が 4 層、エンドキャップ
98 部が 6 層で構成される。バレル部の最内層は IBL(Insertable B-Layer) と呼ばれ、順に B-Layer、Layer-1、
99 Layer-2 となっている。

100 ピクセル検出器の全体図を図 1.7 に示す。

101 ピクセル検出器の各層は、モジュールと呼ばれる最小単位の検出器をいくつも搭載している。ピクセル
102 モジュールを図 1.8 に示す。このピクセルモジュールの詳細については 2 章で述べる。

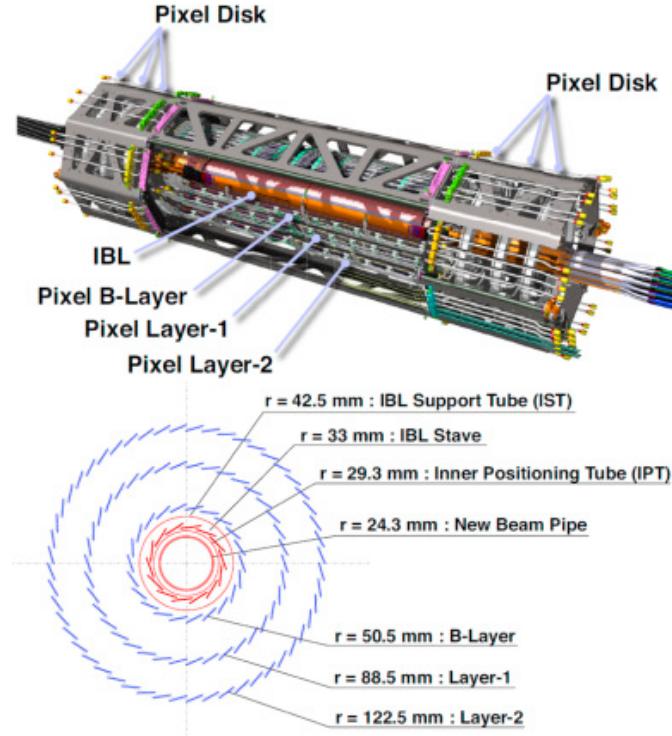


図 1.7 ピクセル検出器の全体像 [8]。上図はピクセル検出器の全体を模式的に表したものであり、下図はビーム軸方向から見たピクセル検出器の断面図である。バレル部の 4 層は内側から IBL、B-Layer、Layer-1、Layer-2 と呼ぶ。

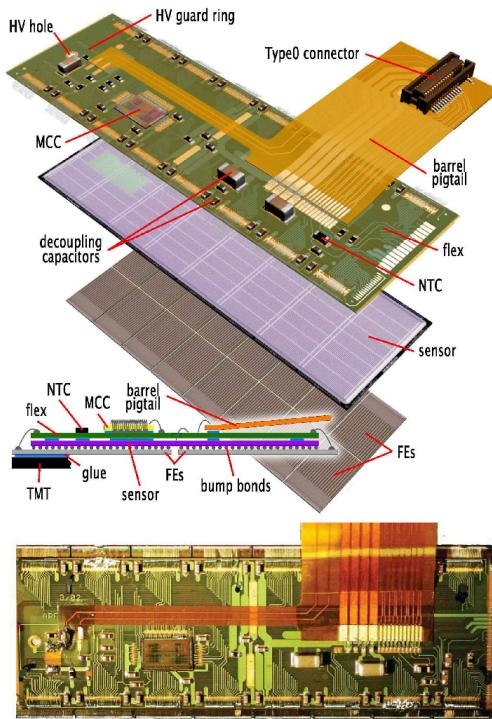


図 1.8 ピクセルモジュールの全体像 [7]。ピクセル検出器は、最小単位としてモジュールと呼ばれる構造を持ち、図はモジュールの全体像を示している。モジュールは、荷電粒子が通過し、信号を生成するセンサー部、AD 変換を行う FE チップ部、データ転送等を行うフレキシブル基板から構成される。

表 1.3 現行 LHC と HL-LHC の比較 [10]。瞬間ルミノシティは 7 倍、積分ルミノシティは 10 倍になることが見積もられており、取得統計数の増加が期待できる。

	LHC	HL-LHC
重心系エネルギー	14	14
瞬間ルミノシティ [$\text{cm}^{-2}\text{s}^{-1}$]	1×10^{34}	7×10^{34}
積分ルミノシティ [fb^{-1}]	300	3,000
1 バンチ交叉あたりの反応数	27	140



図 1.9 HL-LHC 運転計画 [11]。2025 年の初めより HL-LHC の導入、2027 年の途中より本運転が始まる。

1.4 HL-LHC 実験アップグレード計画

LHC では加速器のアップグレードを予定しており、これを HL-LHC アップグレード計画と呼ぶ。詳細を以下に示す。

1.4.1 概要

HL-LHC ではルミノシティ [9] を上げることで、衝突頻度を大きくし、取得統計数を増やす。LHC と HL-LHC の比較を表 1.3 に示す。

LHC の運転計画を表 1.9 に示す。2025 年の初めより HL-LHC の導入が始まり、2027 年の途中から HL-LHC 運転開始の予定となっている。

1.4.2 素粒子物理に対するモチベーション

HL-LHC の素粒子物理に対するモチベーションに 1 つとして、ヒッグス粒子の性質の精密測定があげられる。LHC、HL-LHC における粒子の質量とヒッグス粒子の結合定数の関係と統計誤差の見積もりをまとめたものを表 1.10 に示す。統計誤差が向上する見込みであり、より精密な標準模型の検証を行うこ

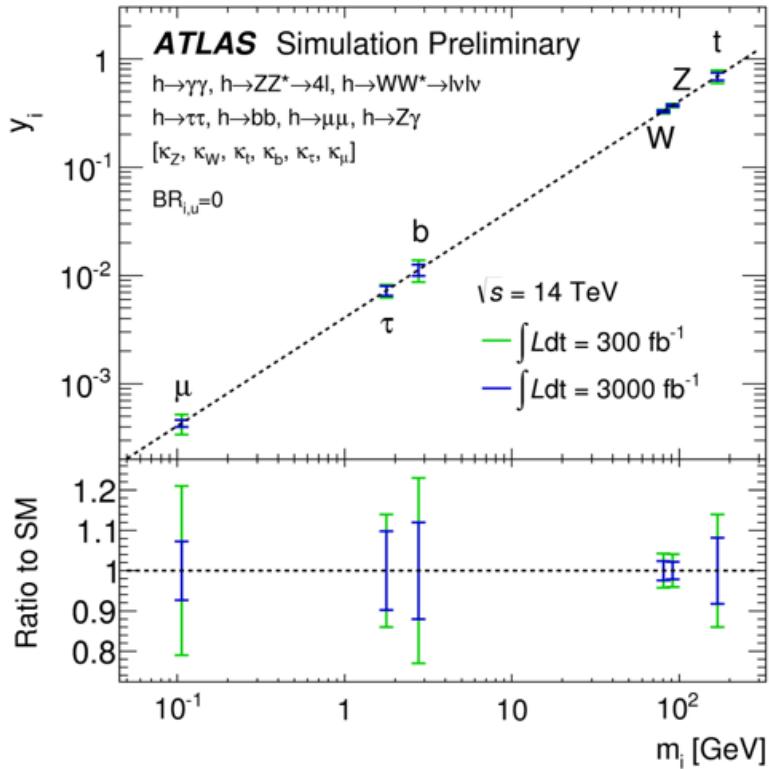


図 1.10 LHC、HL-LHC における粒子の質量とヒッグス粒子の結合定数の関係と統計誤差の見積もり [13]。図において青線、緑線がそれぞれ LHC、HL-LHC の測定における統計誤差の見積もり値を示している。統計誤差は向上する見込みであり、標準模型の精密検証をすることができる。

¹¹⁵ とができる。

¹¹⁶ 1.4.3 内部飛跡検出器のアップグレード

¹¹⁷ HL-LHC のアップグレードによりルミノシティが増加するのに伴い、検出器には以下のような性能が
¹¹⁸ 要求される。

- ¹¹⁹ 放射線耐性の向上。

¹²⁰ 統計量に伴い、放射線量も増加するため高放射線耐性が要求される。

- ¹²¹ 高速読み出し。

¹²² 1 バンチ交叉あたりの反応数 (パイルアップ) の増加に伴いトリガーレートが増加するため、現行
¹²³ よりも高速な読み出しが要求される。トリガーレート 1 MHz に対して、5.12 Gb/s の読み出しス
¹²⁴ ピードが要求される。現行の検出器では 160 Mb/s であり、この要求を満たすことができない。

- ¹²⁵ 検出器の細密化。

¹²⁶ パイルアップが約 5 倍になるため、これらの衝突点を区別するためにはより細密な検出器を使用す
¹²⁷ る必要がある。図 1.11 にイメージを示す。

¹²⁸ HL-LHC に向けて内部飛跡検出器はアップグレードを予定しており、検出器の総入れ替えを行う。アッ
¹²⁹ プグレード後の検出器を **Inner Tracker(ITk)** と呼ぶ。模式図を図 1.12 に示す。

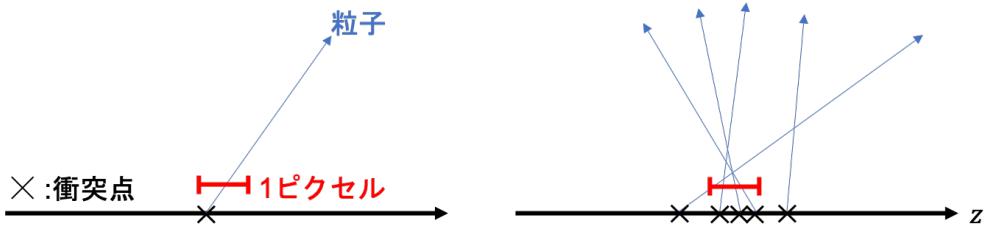


図 1.11 パイルアップ数増加のイメージ。図において黒線がビーム軸と衝突点、青線が生成粒子、赤線が 1 ピクセルを表す。左図が LHC、右図が HL-LHC の様子を表している。HL-LHC ではパイルアップが約 5 倍に増えるため、図のように衝突点の間隔が小さくなる。これらを区別するためには今までよりも細密な検出器を設置する必要がある。

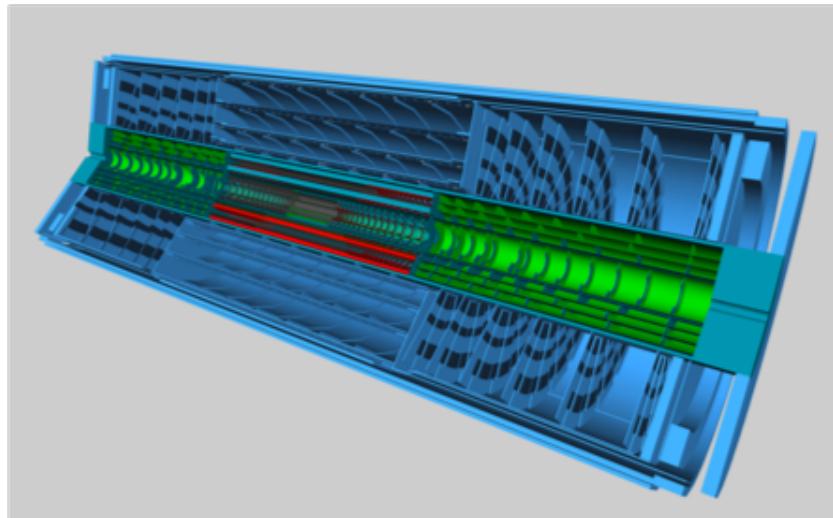


図 1.12 ITk の全体像 [12]。図は ITk 全体の模式図を示している。ITk はピクセル検出器(緑の領域)とストリップ検出器(青の領域)から構成される。

表 1.4 ピクセル検出器設置領域の比較。表は現行と ITk におけるピクセル検出器設置領域の比較を示す。 η の範囲に関して、括弧内は度数法の値を示す。ITk では遷移放射検出器を使用しないため、ピクセル検出器の半径方向のカバー領域が増え、層の数は 4 から 5 となる。また前方方向に生じる粒子測定を行うために、 η の領域も拡大している。

	現行	ITk
$r[\text{mm}]$	33~129	39~279
層の数(バレル部)	4	5
$ \eta $	$< 2.5 (< 19^\circ)$	$< 4 (< 4.2^\circ)$

130 ITk の構成と現行ピクセル検出器との比較

131 図 1.13 に ITk のビーム軸方向の断面図を示す。ITk はピクセル検出器とストリップ検出器で構成される。ピクセル検出器はバレル、傾斜バレル、エンドキャップ部で構成され、バレル部は 5 層となっている。

132 ピクセル検出器の配置に関して、現行と ITk の比較を表 1.4 に示す。またモジュール数の比較を表 1.5 に示す。ITk では現行に比べ、 η が大きい領域まで検出器を設置するため、その分使用するモジュール数も増える。

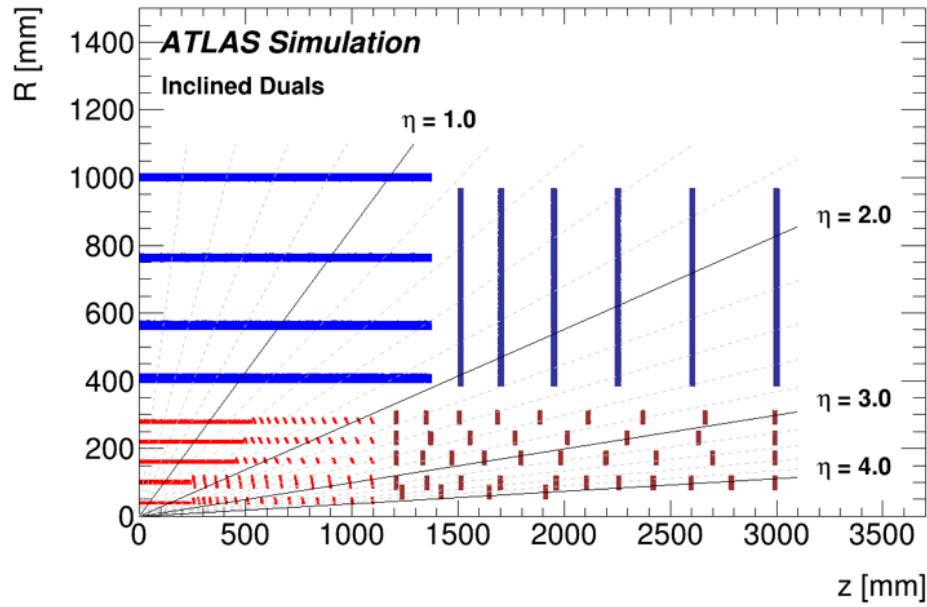


図 1.13 ITk の断面図 [12]。ITk はピクセル検出器とストリップ検出器から構成され、バレル部はそれぞれ 5、4 層の構造を持つ。ピクセル検出器はバレル部、傾斜バレル部、エンドキャップ部に分かれ、 $|\eta| < 4.0$ までの範囲をカバーし、粒子の飛跡測定をすることができる設計となっている。

表 1.5 搭載するピクセルモジュール数の比較。表は現行と ITk のピクセル検出器において、各構造における搭載ピクセルモジュールの数を示している。傾斜バレル部、エンドキャップ部はリングと呼ばれる構造が z 軸方向に層になっており、括弧内の数字は各層におけるリングの数を示す。ITk ではバレル部で現行の約 2 倍になっているのに加え、傾斜バレル部、エンドキャップ部に多くのモジュールを搭載する設計となっていることが分かる。

層	バレル部		傾斜バレル部		エンドキャップ部	
	現行	ITk	現行	ITk	現行	ITk
1	280	192	—	512(16)	—	64(4)
2	286	240	—	520(13)	—	242(11)
3	494	660	—	660(11)	—	320(10)
4	676	960	—	1040(13)	288(3)	352(8)
5	—	1300	—	1300(13)	—	468(9)
合計	1736	3352	0	4032	288	1446

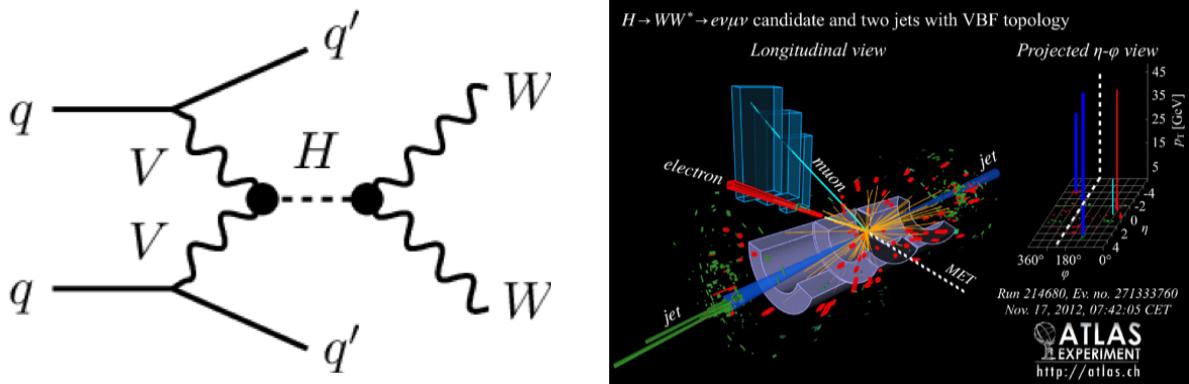


図 1.14 VBF イベントの図 [14]。図は VBF イベントの一つである $H \rightarrow WW^*$ 崩壊チャンネルを示している。左図はファインマン図、右図はそのイベントディスプレイを表す。このイベントにおいて、初めに相互作用したクオーク対は、それぞれハドロン化しじェットと呼ばれる粒子群となる。このときこれらのジェットは前方方向に大きな運動量を持つ。ITk では $|\eta| < 4.0$ の範囲をカバーしており、このようなジェットを捉えることができるため、測定精度が向上する。

表 1.6 VBF $H \rightarrow WW$ の測定における系統誤差の見積もり [12]。表は 3000 fb^{-1} において、 $|\eta| < 2.7$ 、 $|\eta| < 4.0$ の領域を使用した場合の VBF $H \rightarrow WW$ イベント測定における系統誤差の見積もりを示している。 $|\eta| < 4.0$ の領域を使用することで系統誤差が向上する見積もりであることが分かる。ここでは統計誤差は考慮にいれていない。

検出器の使用範囲	$ \eta < 2.7$	$ \eta < 4.0$
系統誤差	22%	12%

1.4.4 物理測定に及ぼす影響

カバーする η の範囲が大きい特徴が生む利点として、前方方向に大きな運動量を持つ粒子を含む物理イベントの測定精度向上があげられる。

この例として、ボゾン粒子結合によるヒッグス粒子生成過程 (Vector boson fusion higgs production, VBF) をあげる。VBF の 1 つのチャンネルに関するファインマン図及びイベントディスプレイを図 1.14 に示す。この衝突により生じる 2 つのクオークはジェットと呼ばれる粒子群となり、多くのジェットは図 1.15 に示すように前方方向に大きな運動量を持つ。

現行 ATLAS 検出器のカロリメータでは $|\eta| < 4.9$ の領域をカバーしているため、エネルギー測定は行っている。ITk では、飛跡検出器でジェットを捉えることによってエネルギーや崩壊点測定精度の向上が期待され、VBF イベントの測定精度を向上することができる。

$VBF H \rightarrow WW$ における系統誤差は表 1.6 のように見積もられる [12]。

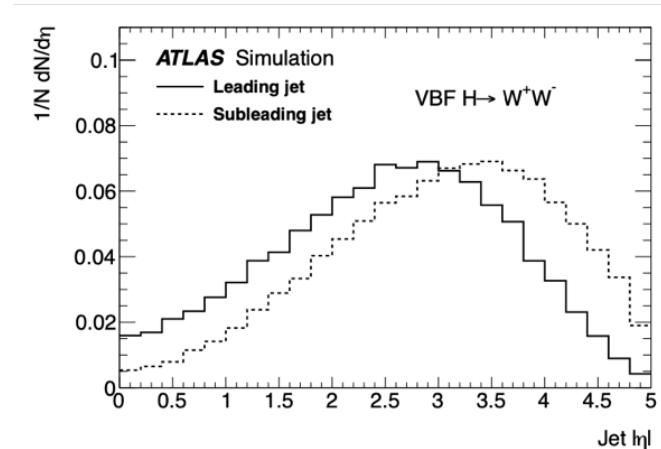


図 1.15 VBF イベントにおけるジェットの運動方向の η 分布 [12]。図はシミュレーションの結果を示している。横軸はジェットの運動方向の η 成分を示しており、縦軸はイベント数に比例する量である。“Leading jet”、“Subleading jet”は各イベントにおいて 1、2 番目に大きな運動量を持つジェットである。 $|\eta| = 3 - 3.5$ 付近にピークを持ち、多くのジェットは前方方向に大きい運動量を持つことが分かる。

第2章

147

ピクセルモジュール

148

149 この章ではピクセルモジュールの構成と各構成部品について説明する。

2.1 ピクセルモジュールの構成

151 ピクセルモジュールはベアモジュールとフレキシブル基板より構成される。ベアモジュールは荷電粒子
152 の通過を検知し、信号を発生するシリコンセンサーと、AD 変換を行う FE チップで構成される。ベアモ
153 ジュールが持つ FE チップの数はモジュールの種類によって異なる。モジュールの構成を図 2.1 に示す。

2.2 ピクセルモジュールの構成部品

2.2.1 シリコンセンサー

154 ピクセルモジュールに搭載するセンサーはシリコン半導体を用いている。センサー内部構造として pn
155 接合を持ち、逆バイアス電圧をかけ空乏層を広げた状態で使用する [15]。この空乏層領域に荷電粒子が通
156 過すると、Bethe-Bloch の式 [16] に従い粒子はエネルギーを損失する。このエネルギー損失に従い、電子
157 ・ホール対が生成、これを収集することで荷電粒子の通過情報をアナログ信号として取得するこ
158 ができる。
159
160

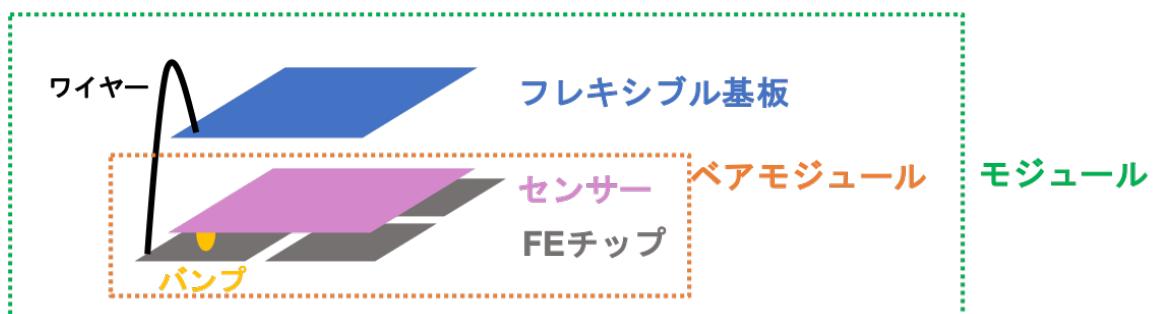


図 2.1 ピクセルモジュールの構成。図は Quad モジュールの構成を模式的に表したものである。モ
ジュールは「ベアモジュール」と呼ばれる心臓部と「フレキシブル基板」を貼り付けることで作られ
る。ベアモジュールはセンサーと FE チップから成る。Quad モジュールの場合、1 枚のセンサーに対
し FE チップは 4 枚である。センサーと FE チップは「パンプ」と呼ばれる構造で電気的に接続して
おり、1 ピクセルに対して 1 つのパンプ接合がなされている。FE チップとフレキシブル基板の接続に
は、ワイヤーが多数 ($O(100)$) 配線される。

表 2.1 センサーのスペック。

項目	プラナー	3D
ピクセルサイズ [μm^2]	50 × 50	50 × 50, 25 × 100
厚さ [μm]	100, 150	150



図 2.2 新型ピクセルモジュールに搭載するシリコンセンサーの断面図。図は新型ピクセルモジュールに搭載するシリコンセンサー断面の模式図を示している。p型半導体にn+型電極を埋め込んだn+ - in - p型と呼ばれる構造を持つ。p+側にフレキシブル基板、n+側にFEチップが付く。逆バイアス電圧を印加するとn+電極側から空乏層が広がる。

161 新型ピクセルモジュールに搭載するセンサーはバルク部にp、電極部にn+を用いたn+ - in - p型である。
 162 模式図を図2.2に示す。n+電極で電子を収集し、信号を取得する。

163 ITkのピクセルモジュールに使用するシリコンセンサーとしてプラナー、3Dの2種類を開発している。
 164 [12]。センサーのスペックを表2.1に記す。

165 2.2.2 読み出しFEチップ

166 読み出しFEチップはシリコン半導体を用いて作られた集積回路である。読み出しFEチップの主な役割は、シリコンセンサーで発生し受け取ったアナログ信号を整形、增幅したのちAD変換し、後段に転送することである。AD変換について、アナログ信号がThresholdを超えた時間幅を測定し、デジタル信号に変換する。この信号の値を**Time over Threshold(ToT)**と呼ぶ。

170 ToTの概念図を図2.3に示す。

171 RD53A

172 RD53A[15]は、新型ピクセルモジュールの研究、開発のために作られたプロトタイプの読み出しFEチップである。RD53Aの性能を表2.2に示す。チップサイズ、ピクセル数は、ITkに搭載予定のモジュールが持つFEチップの半分となっている。

175 FEチップ上の各ピクセルはアナログ回路部とデジタル回路部を持つ。RD53Aでは図2.4に示すように、3つの領域があり、それぞれの領域でピクセルのアナログ回路が異なる。左から順にSynchronous FE、Linear FE、Differential FEと呼ぶ。研究、開発用に3つの領域が設けられているが、性能比較の結果ITkに搭載するモジュールにはDifferential FEを用いることが決定している。なお、デジタル回路部は全てのピクセルにおいて共通である。それぞれの回路図は付録Bに示す。

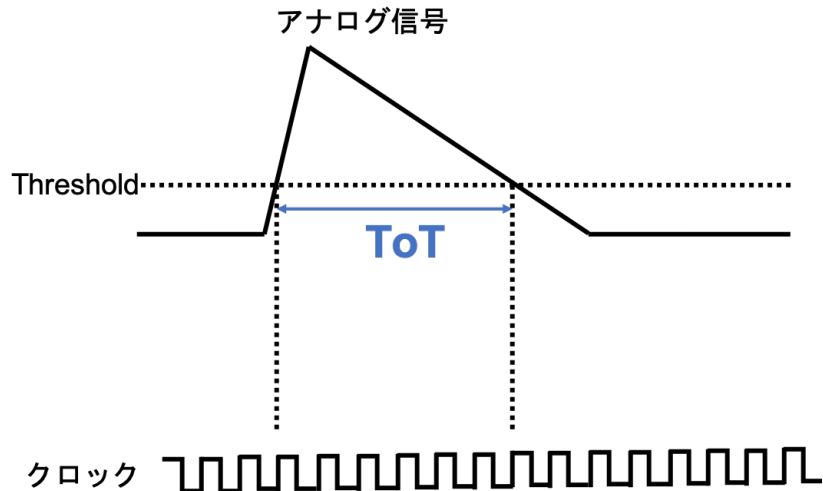


図 2.3 ToT の概念図。FE チップで生成されるデジタル信号は ToT と呼ばれる。図はその概念を模式的に表したものであり、FE チップで行われている AD 変換である。ToT はアナログ信号が Threshold 値を超えた時間幅に相当する。実際には ToT 間のクロック数 [b.c.] として取得される。ここで 1 b.c.(25 ns) は LHC における陽子バンチの交叉間隔に相当する。

表 2.2 RD53A のスペック。

チップサイズ [mm ²]	20.0 × 11.6
ピクセルサイズ [μm ²]	50.0 × 50.0
ピクセル数 [行 × 列]	400 × 192
トリガーレート [kHz]	1000
データレート [Mbps]	1280 × 4
ToT の範囲	0~15(4 ビット)
放射線耐性 [Mrad(トータルドーズ効果 [17])]	500

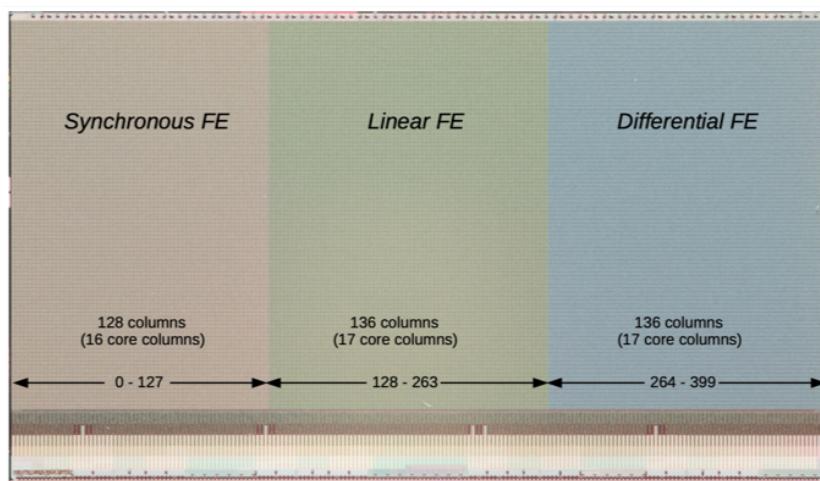


図 2.4 RD53A[15]。ピクセル数は 400 列 × 192 行となっている。図のように RD53A は 3 つの領域が設けられており、それぞれ Synchronous FE、Linear FE、Differential FE と呼ぶ。各領域のピクセルが持つアナログ回路が異なる。

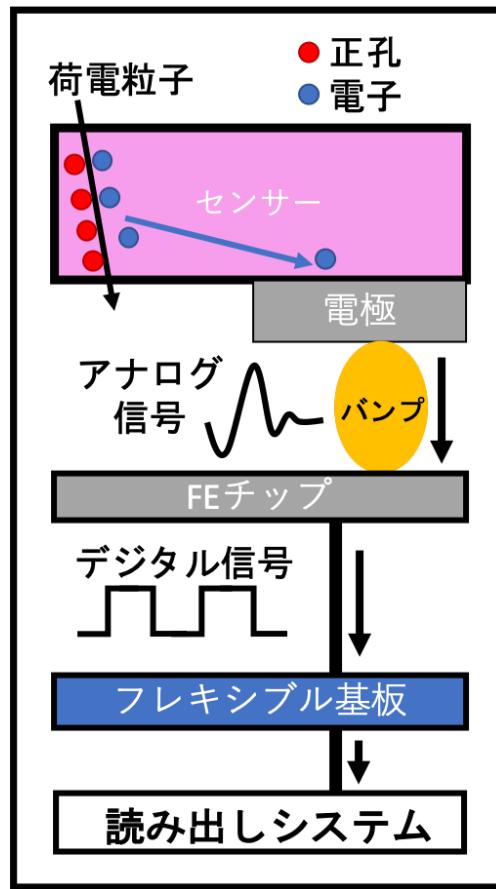


図 2.5 ピクセルモジュールにおける信号伝達の様子。図はピクセルモジュールにおける信号伝達の様子を模式的に表したものである。初めに、荷電粒子がセンサー部を通してエネルギー損失に応じた電子・ホール対を生成する。電子は電極により収集され、バンプを通してアナログ信号として FE チップに送られる。FE チップでは信号を整形、増幅後、デジタル信号に変換する。その後ワイヤーを通してフレキシブル基板、後段の読み出しシステムへ転送される。

¹⁸⁰ 2.2.3 フレキシブル基板

¹⁸¹ フレキシブル基板は、基板上に電子部品が搭載されたものである。FE チップからのデジタル信号を後¹⁸² 段の回路へ転送する他、FE チップ、センサーへの電圧印加制御の役割も担う。

¹⁸³ フレキシブル基板は材料としてポリイミドを使用している。一般的にリジット基板に使われるガラスや¹⁸⁴ エポキシに比べ物質量が小さいため、粒子との相互作用確率を小さくし、物理測定に対する影響を小さく¹⁸⁵ する狙いがある。

¹⁸⁶ 2.2.4 信号伝達

¹⁸⁷ モジュールの信号伝達の様子を模式的に表したものを見ると図 2.5 に示す。センサーで生成した信号は FE¹⁸⁸ チップ、フレキシブル基板の順に送られ、後段の読み出しシステムへ転送される。

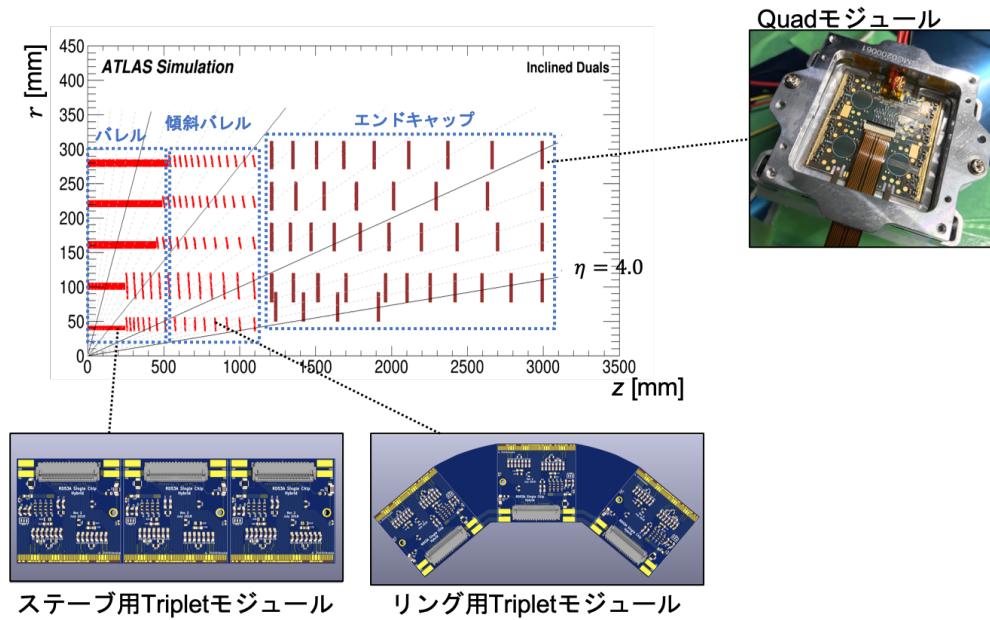


図 2.6 搭載するモジュールのプロトタイプと ITk における配置。2 種類の Triplet モジュールと Quad モジュールを搭載する予定となっている。Triplet モジュールは最内層のステープ構造、リング構造の領域に使われる。それ以外の領域には Quad モジュールが使われる。

189 2.3 新型モジュールの種類

190 現在、ITk に搭載するモジュールとして以下の 3 つのモジュールを予定している。

- 191 • ステープ用 Triplet モジュール
- 192 • リング用 Triplet モジュール
- 193 • Quad モジュール

194 Quad モジュールはシリコンセンサー 1 つ、FE チップを 4 枚搭載する。一方、Triplet モジュールはシリコンセンサー 1 つ、FE チップを 1 枚搭載するシングルベアモジュールを 3 つ搭載する。ITk 最内層は Triplet モジュールで構成される。バレル部を構成する Triplet モジュールをステープ用、傾斜バレル、エンドキャップ部を構成するものをリング用と呼ぶ。それ以外の全ての領域は Quad モジュールで構成する。それぞれのモジュールの図及び検出器上における位置の一例を図 2.6 に示す。

199 第3章

200 モジュール組み立てと品質試験

201 ITk に搭載するピクセルモジュールは、世界中に存在する複数の組み立て機関で製造される。またモ
202 ジュールの組み立てとともに性能確認のための品質試験を行う。

203 本研究の主題はこのモジュール品質試験のデータ管理を行うデータベースシステムの開発である。本章
204 ではデータ管理の対象となるモジュールの組み立て工程と品質試験の概要を説明する。開発したデータ
205 ベースシステムについては 4 章で説明する。

206 3.1 組み立て工程

207 モジュール組み立て機関は、初めにベアモジュールとフレキシブル基板をそれぞれの製造機関から受け
208 取る。組み立て工程として以下が設定されている。

209 1. ベアモジュール・フレキシブル基板貼り付け.

- 210 • 受け取ったベアモジュールとフレキシブル基板を接着剤を用いて貼り付ける。

211 2. ワイヤー配線.

- 212 • ワイヤー配線を行い、FE チップとフレキシブル基板を電気的に接続する。

213 3. ワイヤー保護.

- 214 • ワイヤーが損傷があり断線が起きると、そのワイヤーに接続されているピクセルの読み出しが
215 できなくなる。これを防ぐため、モジュールに屋根型の構造を取り付け、ワイヤーを物理的に
216 保護する。

217 4. パリレンコーティング.

- 218 • モジュール読み出し部以外での電通や放電を防ぐため、パリレン高分子を用いてモジュールを
219 保護する。

220 5. 温度サイクル試験.

- 221 • ITk 運転時の環境温度は -45°C から 40°C まで変化しうる [18]。この温度変化に耐えうるか
222 を確認するため、温度サイクルを行う。

223 6. 低温耐久試験.

- 224 • ITk 運転時の典型的な環境温度は $-15\sim0^{\circ}\text{C}$ 付近である。これに耐えうる性能を持つかを確
225 認するために、低温下にモジュールを長時間設置する耐久試験を行う。

226 流れと各組み立て工程のイメージを図 3.1 に示す。

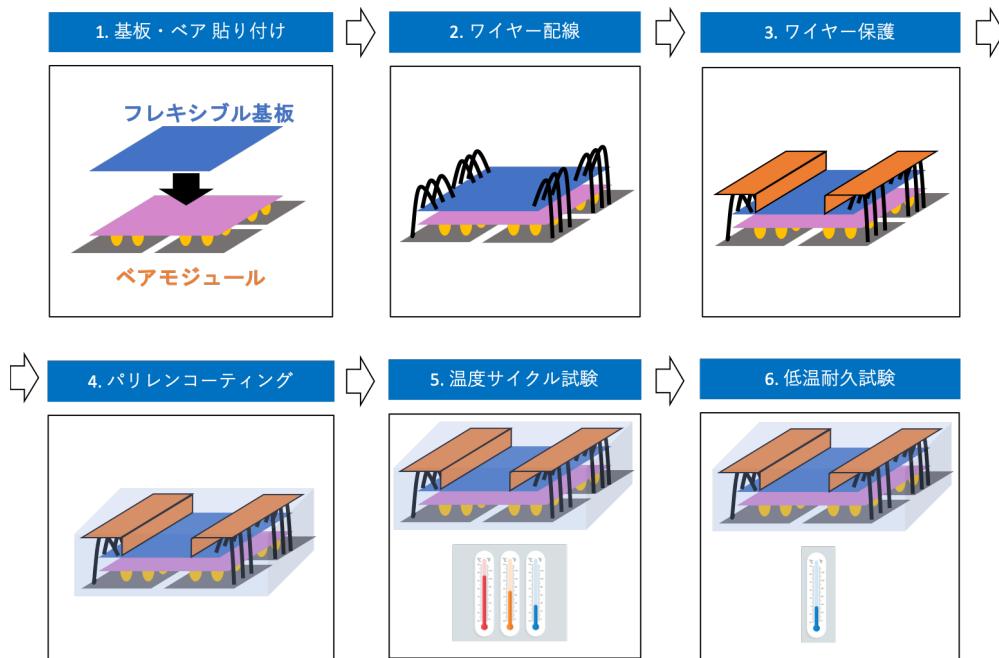


図 3.1 組み立て工程のイメージ図。モジュール組み立て機関はペアモジュール、フレキシブル基板を受け取り、それらの貼り付け、ワイヤー配線、ワイヤー保護、パリレンコーティング、温度サイクル試験、低温耐久試験の順に組み立てを行う。

3.2 品質試験

各組み立て工程に対して、いくつかの品質試験を行う。行う品質試験の代表的なものを以下に示す。

3.2.1 外観検査

モジュールの外観写真を撮り、モジュールに以下のような欠陥がないかを確認する。また外観検査の様子を図 3.2 に示す。

- 抵抗等取り付け部品の損傷.
- ワイヤーの接着位置確認.
- 回路やワイヤーの断線.
- 付着汚れ.

3.2.2 質量測定

モジュールの質量を測定する試験である。

3.2.3 平坦性測定

モジュール上の位置座標を何点か測定し、モジュールの平坦度、厚さ、歪み具合等を測定する。測定の様子と解析の例を図 3.3 に示す。

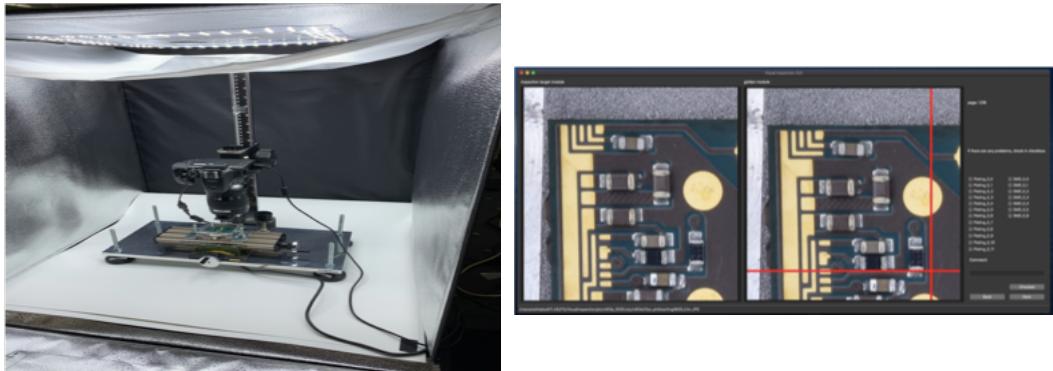


図 3.2 外観検査の様子。図は日本で実際に行われた外観検査における写真撮影の様子(左図)と撮影写真の確認画面(右図)である。左図のように、写真撮影は光量を制御するため暗箱の中で行っている。写真撮影後、右図のように撮影した写真と良好なモジュールの写真を見比べ、電気部品の損傷、回路の断線などの何らかの所見があった場合には記録をする。

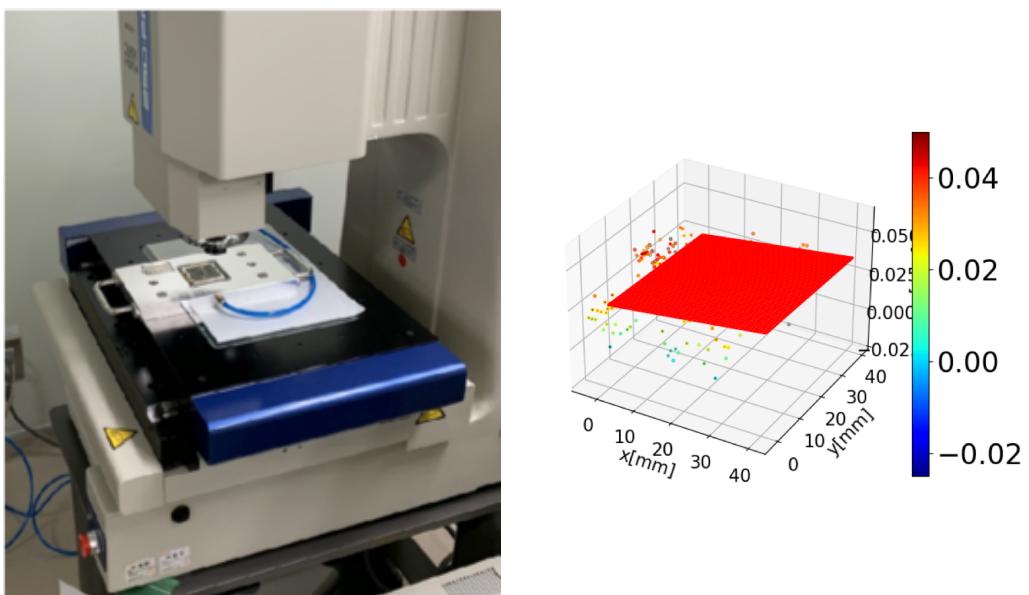


図 3.3 平坦性測定の様子。図は日本で実際に行われた平坦性測定における測定の様子(左図)と解析結果(右図)である。左図のように専用の装置を用いてモジュールの位置座標を何点か測定する。得られた測定点は右図のように図示し、平面でフィッティングを行う。フィット平面からのズレや、モジュールの厚みを計算する。

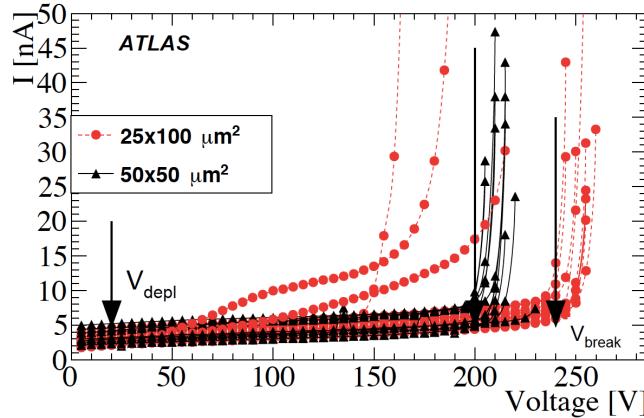


図 3.4 センサー電流-電圧特性結果の例 [6]。図は複数のセンサーに対する測定結果の例を表している。横軸にセンサーに対する印加電圧 [V]、縦軸に電流 [nA] を示す。逆バイアス電圧を印加しているため、電流がほとんど流れていないと、ある地点から電流が急激に増加していることが分かる。この降伏電圧はセンサーにより個体差があるが、どれも 150 – 250 V 付近となっていることが分かる。

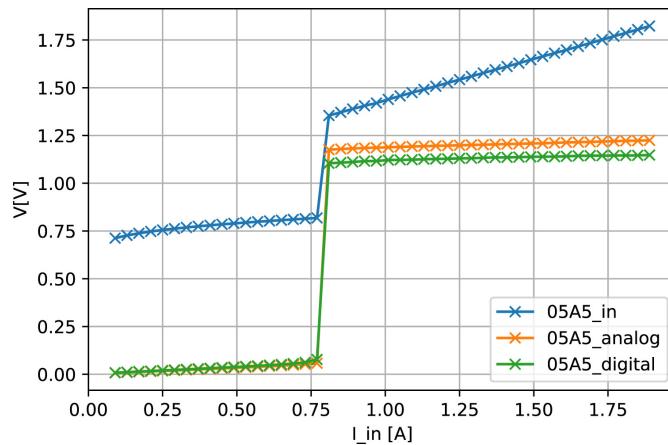


図 3.5 FE チップ電流-電圧特性試験結果の例 [20]。横軸に FE チップに対する設定電流、縦軸は測定した実際に FE チップにかかる電圧値を示している。青、橙、緑はそれぞれ回路全体、アナログ回路部、デジタル回路部に加わる電圧値である。0.75 V 付近で急激に電圧が増大しており、この段階で FE チップが機能するようになる。それ以降は線形になっているのが分かる。

241 3.2.4 センサー電流-電圧特性確認

モジュールのシリコンセンサーに逆バイアス電圧をかけ、電流-電圧特性をみる。印加電圧を段階的に変化させながら電流値を記録し、電圧-電流間の関係を確認する。この試験の結果の例を図 3.4 に示す。逆方向電圧では電流はほとんど流れないと、降伏電圧に達すると急激に増大する。これは pn 接合の特性 [15] であり、正常であればこの振る舞いを確認することができる。

246 3.2.5 FE チップ電流-電圧特性確認

FE チップに対して電圧をかけ、電流-電圧特性をみる。設定電流を段階的に変化させながら電圧値を記録し、電流-電圧間の関係を確認する。この試験の結果の例を図 3.5 に示す。FE チップは抵抗として振る舞い、電流、電圧間の関係は線形性を持つ。

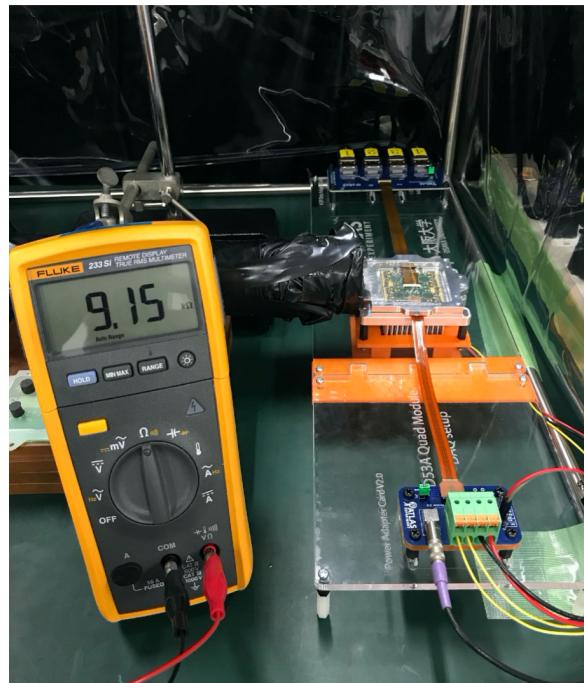


図 3.6 読み出し試験の様子。図は日本で実際に行われた読み出し試験の様子である。図の右側に見えるように、モジュールにケーブルを配線し、PCとの通信を行うことでデータ取得をする。また図の左側では抵抗値を測定している。これはモジュールに付属しているサーミスタの抵抗値を確認することで、温度情報を取得するためのものである。

250 3.2.6 読み出し試験

251 読み出し試験では、読み出し回路が正常に機能するのかを確認する。読み出し試験の様子を図 3.6 に示
252 す。モジュールに読み出しケーブルを配線し、PC と通信を行い、データを取得する。また試験において
253 温度、FE チップやセンサーに与える電圧、電流といった検出器の操作、環境関わる情報取得を行
254 い、異常がないかを確認する。これらの情報は「Detector Control System, DCS」と呼ぶ。

255 汎用読み出しシステム YARR

256 YARR(Yet Another Rapid Readout) システム [21] は、ピクセルモジュール用に開発された、
257 PCI Express(PCIe) 接続を用いた読み出しシステムである。ファームウェア、ソフトウェアから構成さ
258 れる。YARR ではファームウェア上で行う処理はデータ通信やトリガー処理等の最低限に抑え、その他
259 多くの処理をソフトウェアで担うという特徴がある。

260 読み出し試験に使用するファイルと変数

261 YARR で扱う全てのファイルは **JSON**(**JavaScript Object Notation**) と呼ばれる形式で記述さ
262 れる。

263 YARR を用いた読み出し試験では以下の設定ファイルが要求される。

264 ● 試験設定ファイル.

265 – 読み出し試験の初期設定や解析手法を記述する。

266 ● ハードウェア設定ファイル.

267 – 試験に用いるハードウェアの指定や設定を記述する。

268 ● 接続設定ファイル.

269 – 読み出しを行う FE チップの種類やチャンネルを記述する。

270 ● FE チップ設定ファイル.

271 – 各 FE チップ毎に出力され、全ピクセルに共通な試験の設定値、各ピクセル固有の設定値を記
272 述する。

273 また試験 1 項目ごとに以下のファイルが、1 つのディレクトリに生成される。

274 ● 試験結果ファイル.

275 – 読み出し試験の結果値を記述する。

276 ● 試験設定ファイル.

277 ● FE チップ設定ファイル.

278 – 試験の中で変更が加えられるため、各 FE チップにつき試験前後の 2 つのファイルが出力さ
279 れる。

280 ● 試験ログ.

281 – 試験情報を記録する。

282 後述するローカルデータベースシステムにおいてはさらに以下の設定ファイルを用いる。

283 ● データベース設定ファイル.

284 ● 試験者設定ファイル.

285 ● 試験場所設定ファイル.

286 読み出し試験項目において以下の *Occupancy* という量が定義される。試験用電荷の入射数や発行トリ
287 ガーの数等、発行した信号数を n_i 回、取得信号数を n_0 としたとき、

$$\text{Occupancy} = \frac{n_0}{n_i} \times 100 [\%] \quad (3.1)$$

288 と定義する。

289 読み出し試験では各 FE チップに対して「レジスタ」と呼ばれる設定値が存在し、FE チップ上の全ピ
290 クセルに対して共通なものを「グローバルレジスタ」、各ピクセル固有のものを「ピクセルレジスタ」と
291 呼ぶ。レジスタの 1 つに Threshold 用レジスタがある。読み出し試験ではこのレジスタの調整を行う。

292 **読み出し試験項目**

293 以下に読み出し試験項目の一覧を示す。

294 **レジスタの読み書き.**

295 グローバル及びピクセルレジスタの読み書きが正常にできるのかを確認する試験.

296 **デジタル回路読み出し.**

297 各ピクセルのデジタル回路部に試験用電荷を入射し、信号の応答数を確認する試験. デジタル回路
298 部の性能確認に用いる.

299 **アナログ回路読み出し.**

300 各ピクセルのアナログ回路部に試験用電荷を入射し、信号の応答数を確認する試験. アナログ回路
301 部の性能確認に用いる.

302 **Threshold 測定.**

303 各ピクセルの Threshold 値を測定する試験.

304 **Threshold グローバルレジスタ調整.**

305 Threshold グローバルレジスタの変更、基準となる Threshold に近づけるための調整.

306 **Threshold ピクセルレジスタ調整、再調整、精密調整.**

307 Threshold ピクセルレジスタの変更、基準となる Threshold に近づけるための調整.

308 **ToT グローバルレジスタ調整.**

309 ToT グローバルレジスタの変更、基準となる ToT に近づけるための調整.

310 **ノイズ占有率測定.**

311 各ピクセルのノイズの頻度を確認する試験.

312 **スタッキピクセル測定.**

313 入力電荷の有無にかかわらず、常に信号を出力するピクセルを確認する試験.

314 **クロストーク測定.**

315 各ピクセルのクロストークの有無を確認する試験. 隣接する Analog FE に試験用電荷を入射し、
316 応答を確認する.

317 **バンプ接続確認測定.**

318 各ピクセルのバンプ接合が正常かを確認する試験. 隣接する Analog FE に大きめの試験用電荷を
319 入射する. バンプが正常であればクロストークにより応答を確認できる.

320 各試験における試験用電荷入射の方法について、付録 B に示す。

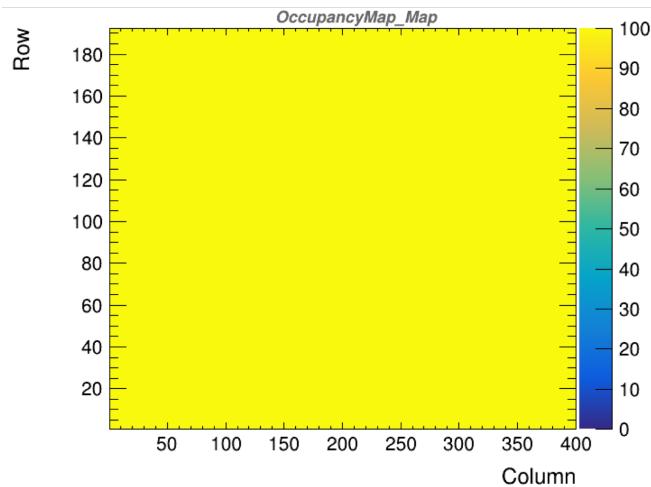


図 3.7 デジタル回路読み出しにおける *Occupancy* 分布の例。図は RD53A を用いたデジタル回路読み出しの結果であり、横軸、縦軸はそれぞれピクセルの列 (400)、行 (192) に対応する。*z* 軸は各ピクセルにおける *Occupancy* の値を示している。正常なピクセルは 100 付近になることが期待され、図では全てのピクセルで 100 であると分かる。

表 3.1 デジタル回路読み出しにおける結果ファイル一覧。

ファイル名	データ	ファイルサイズ
OccupancyMap	各ピクセルの <i>Occupancy</i> を記す。	1.5 MB
Enmask	<i>Occupancy</i> = 100 のピクセルを 1、それ以外を 0 とした値を持つ。	1.3 MB
L1Dist	信号取得タイミングの分布を持つ。	0.53 kB

321 主な試験項目の詳細について以下で説明する。

322 デジタル回路読み出し

323 各ピクセルのデジタル回路部に試験用電荷を入射し、信号を取得する。*Occupancy*(式 3.1) は、入射電
324 荷数 n_i と取得信号数 n_0 で定義される。YARR のデフォルトでは $n_i = 100$ となっている。

325 デジタル回路読み出しにおける *Occupancy* の二次元分布の例を図 3.7 に示す。一般にこの試験におい
326 て、*Occupancy* は 100 であることが望ましい。

327 この試験の結果として出力されるファイルを表 3.1 に示す。

328 アナログ回路読み出し

329 各ピクセルのアナログ回路部に試験用電荷を入射し、信号を取得する。デジタル回路読み出しと同様、
330 *Occupancy*(式 3.1) は、入射電荷数 n_i と取得信号数 n_0 で定義され、*Occupancy* は 100 であることが望
331 ましい。試験結果として出力されるファイルはデジタル回路読み出しと同じ形式である。

332 Threshold 測定

333 各ピクセルのアナログ回路部に試験用電荷を入射し、入射電荷数 n_i と取得信号数 n_0 より *Occupancy*
334 を測定する。これを入射電荷量を増加させて繰り返し行う。あるピクセルにおけるこの処理結果の例を図
335 3.8 に示す。

336 この分布を以下の式でフィッティングする。フィッティングの形に由来し、これを「S カーブフィッティ

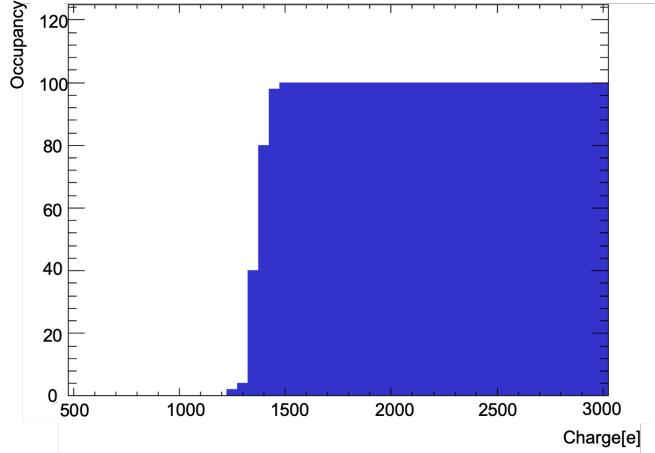


図 3.8 入射電荷量と *Occupancy* の関係。横軸はピクセルに対する入射電荷量 [e]、縦軸は *Occupancy* を示す。入射電荷量の単位 e は、電荷量 C を素電荷 $e = 1.6 \times 10^{-19}[\text{q}]$ で割ったものである。Threshold 測定の際にはこのように入射電荷量を増加させながら *Occupancy* の値を測定する。測定結果を後述する式 (3.2) でフィットし、Threshold とノイズの値を得る。

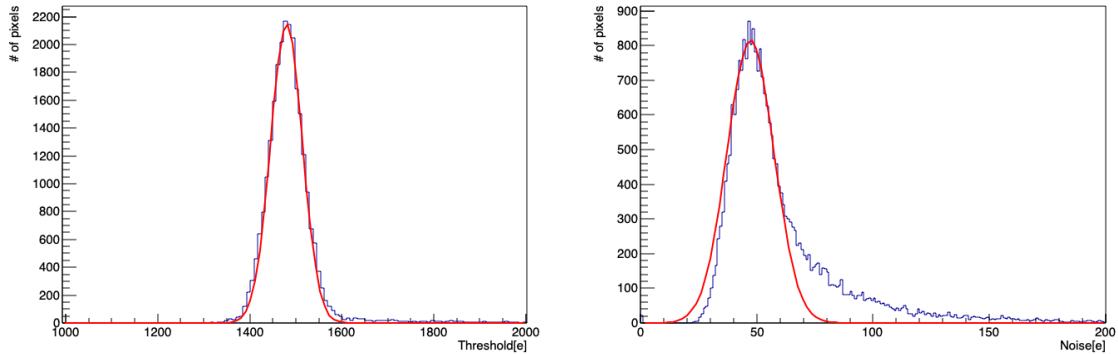


図 3.9 Threshold 値とノイズの分布。図はある FE チップ上のピクセルにおける Threshold 値（左図）とノイズ（右図）の分布を示している。また赤線はガウス関数によるフィット関数を示しており、これにより平均値と幅を取得する。

337 「ング」と呼ぶ。

$$f(x) = 0.5 \times \left[2 - \operatorname{erfc} \left(\left\{ \frac{x - Q}{\sigma \times \sqrt{2}} \right\} \right) \right] \times p \quad (3.2)$$

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) \quad (3.3)$$

338 ここで $\operatorname{erf}(x)$ はガウスの誤差関数である [22]。

339 得られた Q 、 σ がそれぞれピクセルの Threshold 値、ノイズに相当する。ある FE チップにおける
340 Threshold 値、ノイズの分布の例を図 3.9 に示す。この分布をガウス関数でフィットすることで、全ピク
341 セルに対する Threshold、ノイズの平均値と幅が得られる。

342 この試験において、出力される結果ファイルを表 3.2 に示す。

343 ToT 測定

344 各ピクセルのアナログ回路部に試験用電荷を入射し、ToT を測定する。この操作を複数回（デフォルト
345 では 100 回）行い、平均値と幅を求める。出力される結果ファイルを表 3.3 に示す。

表 3.2 Threshold 測定における結果ファイル一覧。

ファイル名	データ	ファイルサイズ
ThresholdMap	各ピクセルの Threshold 値を記す。	2.2 MB
ThresholdDist	Threshold の分布を記す。	4.6 kB
NoiseMap	各ピクセルのノイズを記す。	2.2 MB
NoiseDist	ノイズの分布を記す。	2.3 kB
Chi2Map	各ピクセルの S カーブフィッティングにおける χ^2 の値を記す。	2.3 MB
Chi2Dist	χ^2 の分布を記す。	1.1 kB
StatusMap	各ピクセルのフィットの良さを 0 から 5 の 6 段階で記す。	1.3 MB
StatusDist	フィットの良さの分布を記す。	0.49 kB
TimePerFitDist	S カーブフィットに要した時間を記す。	3.1 kB
Scurve	あるピクセルの入射電荷量と Occupancy の関係を記す。 複数のピクセルについて出力する。	1.0 kB
sCurve	全てのピクセルの Scurve を重ね合わせた値を記す。	1.3 MB

表 3.3 ToT 測定における結果ファイル一覧。

ファイル名	データ	ファイルサイズ
MeanToTMap	各ピクセルの ToT の平均値を記す。	1.9 MB
MeanToTDist	ToT 平均値の分布を記す。	0.59 kB
SigmaToTMap	各ピクセルの ToT の標準偏差を記す。	2.2 MB
SigmaToTDist	ToT 標準偏差の分布を記す。	1.8 kB
L1Dist	信号取得タイミングの分布を記す。	0.59 kB

表 3.4 ノイズ占有率測定における結果ファイル一覧。

ファイル名	データ	ファイルサイズ
OccupancyMap	各ピクセルの Occupancy を記す。	1.3 MB
NoiseOccupancyMap	各ピクセルの NoiseOccupancy を記す。	1.3 MB
NoiseMask	$NoiseOccupancy < 10^{-6}$ のピクセルを 1、それ以外を 0 とした値を記す。	1.3 MB

346 ノイズ占有率測定

347 試験用電荷を入射せずに、周波数 f [Hz]、時間 t [sec] のトリガーをかけ、取得信号数 n_0 を測定する。
 348 YARR のデフォルトでは $f = 5000$ 、 $t = 300$ である。Occupancy(式 3.1) は、発行したトリガー数
 349 $n_i = f \times t$ と取得信号数 n_0 で定義される。

350 また NoiseOccupancy を以下で定義する。NoiseOccupancy の分布の例を図 3.10 に示す。

$$351 \quad NoiseOccupancy = \frac{n_0}{t} \times (25 \times 10^{-9}) \quad (3.4)$$

352 あるピクセルにおいてノイズの量が大きいと Occupancy、NoiseOccupancy が値を持つ。読み出しに
 353 おいてはノイズは小さいことが望まれ、これらの値は 0 に近いと正常なピクセルであると言える。

354 出力される結果ファイルを表 3.4 に示す。

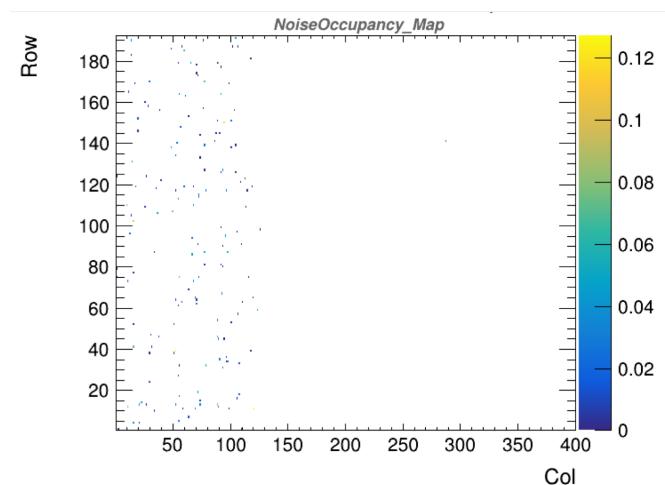


図 3.10 ノイズ占有率測定における *NoiseOccupancy* 分布の例。図は RD53A を用いたデジタル回路読み出しの結果であり、横軸、縦軸はそれぞれピクセルの列(400)、行(192)に対応する。z 軸は各ピクセルにおける *NoiseOccupancy* の値を示している。正常なピクセルでは限りなく 0 に近い値になることが期待され、図ではほとんどのピクセルが 0 となっているが、列番号が 0 から 120 付近の領域で値を大きく持つピクセルが存在する。

表 3.5 ピクセル解析の評価基準一覧 [23]。読み出し試験において各ピクセルが正常に機能しているかを判断するためにピクセル解析を行う必要があり、その判断基準が表のように定義されている。この基準一覧は不良評価であり、全ての基準に当てはまらないピクセルが良好と判断される。不良ピクセルの数や分布は、モジュールの品質を決定する 1 つの要素となり、ITk に搭載するモジュールの選択や配置の決定に用いられる。

評価名	読み出し項目	評価基準
Digital Dead	Digital scan	$Occupancy < 1$
Digital Bad	Digital scan	$Occupancy < 98 \text{ or } Occupancy > 102$
Merged Bump	Analog scan Crosstalk scan	$Occupancy < 98 \text{ or } Occupancy > 102$ High Crosstalk
Analog Dead	Analog scan	$Occupancy < 1$
Analog Bad	Analog scan	$Occupancy < 98 \text{ or } Occupancy > 102$
Tuning Failed	Threshold scan	S カーブフィット失敗 (YARR では $\chi^2 = 0$ となる)
Tuning Bad	Threshold scan ToT scan	$ Threshold - Threshold_{\text{平均}} > 5 \times Threshold_{\text{幅}}$ $ToT = 0 \text{ or } 15$
High ENC	Threshold scan	$ \text{ノイズ} - \text{ノイズ}_{\text{平均}} > 3 \times \text{ノイズ}_{\text{幅}}$
Noisy	Noise scan	$NoiseOccupancy > 10^{-6}$
Disconnected Bump	Disconnected bump scan Source scan	現段階では未決定 $Occupancy$ が FE チップ全体平均の 1%
High Crosstalk	Crosstalk scan	$Occupancy > 0 \text{ with } 25\text{ke} \text{ (sync FE)}$ $Occupancy > 0 \text{ with } 40\text{ke} \text{ (lin and diff FE)}$

355 Threshold 調整とピクセル解析

356 今後この論文では、この試験を読み出し試験と呼ぶ。以下の流れで読み出しを行う。

- 357 • デジタル回路読み出し.
- 358 • アナログ回路読み出し.
- 359 • Threshold 測定.
- 360 • Threshold グローバルレジスタ調整.
- 361 • Threshold ピクセルレジスタ調整.
- 362 • ToT グローバルレジスタ調整.
- 363 • Threshold グローバルレジスタ再調整、精密調整.
- 364 • Threshold 測定.
- 365 • ノイズ占有率測定.
- 366 • スタックピクセル測定.
- 367 • クロストーク測定.

368 この試験の中では読み出し回路の性能確認に加えて、各ピクセルが持つ Threshold、ToT をある共通値に
369 近づける調整を行う。

370 測定後、試験結果の解析を行い、モジュール上の各ピクセルが正常かどうかを判断する。設定されてい
371 る評価基準を表 3.5 に示す。不良ピクセルには評価基準に応じた評価名が付けられる。

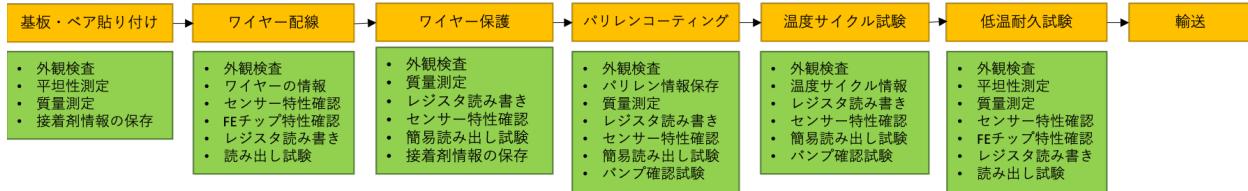


図 3.11 組み立て工程と対応する品質試験一覧。節 3.1 で述べた各組み立て工程において、複数の品質試験を行う必要がある。図は各組み立て工程と対応する品質試験一覧を示している。試験数は 30 程度存在する。

372 簡易読み出し試験

373 品質試験項目の 1 つに、以下の項目を扱う簡易読み出し試験がある。

- 374 • レジスタの読み書き.
- 375 • デジタル回路読み出し.
- 376 • アナログ回路読み出し.
- 377 • Threshold 測定.
- 378 • ToT 測定.
- 379 • バンプ接続確認測定.

380 3.2.7 各組み立て工程における品質試験

381 各組み立て工程と品質試験項目を図 3.11 に示す。図より、1 モジュールに対して行われる品質試験は 382 30 程度存在することが分かる。

383 3.3 検出器量産におけるデータ管理

384 各組み立て機関で $O(100) \sim O(1,000)$ のモジュールを作り、上述したように 1 モジュールに対して 30 385 程度の品質試験を行う。1,000 モジュール作る機関であれば 3,000 の試験結果が得られる。この数の品質 386 試験結果を正確に管理することが必要となる。特に読み出し試験については、1 試験につき更に細かい項目に細分化され、項目、結果ファイルも多様である。これらの情報は最終的に共通のデータベースに保存 387 する必要があり、各組み立て機関で適切に管理する必要がある。

388 本研究では各組み立て機関におけるモジュール情報及び品質試験のデータ管理を簡易化することを目的 389 として、データベースシステムの構築を行った。このシステムについて 4 章で説明する。

第4章

モジュール情報及び品質試験結果管理システム

3章で述べたように、モジュール組み立て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定していて、現在この開発を行っている。システムは、ITkの製造に関わる全情報を保存する中央データベースと、各組み立て機関に設置し、データ管理を行うローカルデータベースから成る。データベースシステムの模式図を図4.1に示す。

本章ではこれらのデータベースの概要と、本研究における開発項目を説明する。

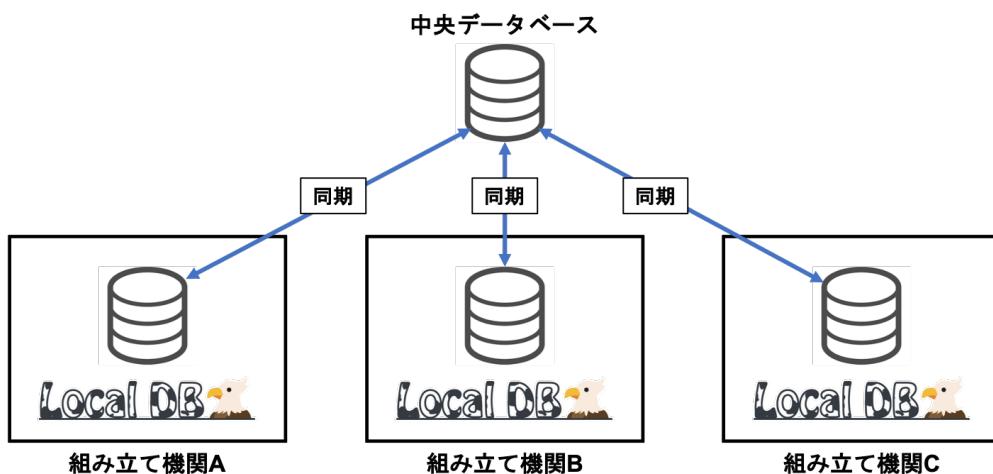


図4.1 データベースシステムの模式図。モジュール情報と品質試験を管理するデータベースはITkの製造に関わる全情報を保存する中央データベースと、各組み立て機関に設置し、データ管理を行うローカルデータベースからなる。全モジュール組み立て機関はローカルデータベースを設置しデータを管理する。中央データベースに全ての情報を保存するために、データベース間は同期が行われる必要がある。

399 4.1 中央データベース

400 4.1.1 中央データベースの概要

401 概要

402 中央データベースは、ITk 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。
403 ユニコーン大学 [24] 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。
404 ITk の製造にあたって、シリコンセンサやフレキシブル基板といった小さな部品から製造を行い、それ
405 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ステープやリングの組み立てを経て ITk
406 が完成する。また各組み立て段階において、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
407 報、及び品質試験結果を中央データベースに保存する。

408 意義

409 中央データベースを扱う重要な目的の 1 つに、ITk に用いるモジュールの選別とその配置の決定があげ
410 られる。中央データベースに保存された品質試験結果を解析し、ITk に搭載するモジュールを選別を行
411 う。また、モジュールの配置も品質を考慮して決定する。例として以下の 2 つをあげる。

- 412 1. $|\eta|$ が小さい領域には、不良ピクセルが少なく品質の良いモジュールを搭載する。
413 2. 不良ピクセルの領域や不良モジュールが固まらないようにする。

414 項目 1 に関して、ヒッグスなど物理解析で興味の対象となる粒子の多くは相対的に大きい質量をもち、 $|\eta|$
415 の小さい領域に生成される。図 1.14 のイベントディスプレイでも、ヒッグス粒子から生成される電子や
416 ミューオンが持つ $|\eta|$ は小さいことが分かる。そのため、 $|\eta|$ が小さい領域には品質の良いモジュールを配
417 置する。項目 2 に関して、不良ピクセルやモジュールの領域がばらけるようにモジュールの配置を決定
418 する。

419 次に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
420 だすと、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
421 照することができる。また運転前後での検出器性能比較を行うことができる。

422 4.2 ローカルデータベース

423 4.2.1 ローカルデータベースの意義と概要

424 中央データベースでは、前述したようにモジュール情報のみならず ITk の製造に関わるすべての情報
425 を管理する。データベースの機能としては、製造を通して汎用的に使えるものになっている。モジュール
426 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
427 い。1 つの組み立て機関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
428 にかつ円滑に進むようになっているのが好ましい。このような理由から、組み立て機関での生産性、利便
429 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を開発しており、これを
430 「ローカルデータベース」と呼ぶ。システムの概要図を図 4.2 に示す。オープンソースのサービスである
431 MongoDB[25] と、ローカルデータベース用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
432 央データベースとの同期を行うシステムとなっている。ここで開発しているウェブアプリケーションは

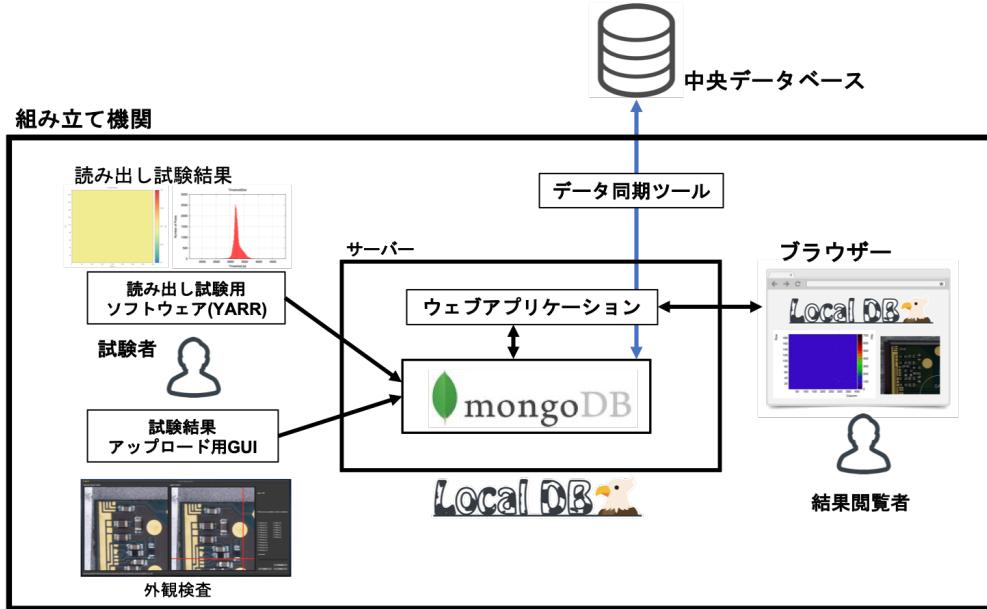


図 4.2 ローカルデータベースシステムの概要。各組み立て機関で MongoDB とローカルデータベース用ウェブアプリケーションの立ち上げを行い、独自にデータ管理をするシステムとなっている。ウェブアプリケーションには Python パッケージである Flask を使用している。品質試験者は図のようにいくつかのソフトウェアを用いて試験結果を MongoDB に保存する。保存された結果はウェブアプリケーションによって閲覧することができる。また結果は中央データベースに集める必要があるため、同期ツールを用いて試験結果の共有を行う。

433 Python のウェブフレームワークである Flask[26] を使用している。

434 ローカルデータベースシステムは以下のような利点を持ち得る。これらの利点を持つシステムの開発、
435 実装を行うことが開発課題となっている。

- 436 • ローカルにデータベースサーバーを立てるため通信が速く、円滑にデータ管理を行うことができる。
- 437
- 438 • モジュールの組み立て工程を管理し、組み立て工程に応じた試験項目の表示や解析を行うことで生産者の適切な処理を助ける。
- 439
- 440 • モジュールに特化したデータ管理、結果解析を行うことで異常をいち早く検知できる。
- 441 • 試験者情報や試験時間など、品質試験結果以外の必要な情報を正確に管理できる。

442 一方で欠点として、ローカルデータベースと中央データベース間で同期を正確に実行することが技術的に困難であることがあげられる。この問題は本研究で取り組んだ内容であり、以下により解決する。それ
443 ぞれの詳細については後述する。

- 444 • 同期ツールの開発。
- 445 • データベース操作の流れの確立。
- 446 • 操作方法の普及に向けたチュートリアル、ドキュメント作成。

447 4.2.2 先行研究と開発課題

448 先行研究で開発された項目と、本研究で取り組んだ開発課題を以下に示す。

450 先行研究

451 先行研究 [27]においてデータベースシステムの設計と開発がなされ、図4.2におけるいくつかの項目が
452 開発された。以下に実装項目を記す。

- 453 • 読み出し試験結果保存のためのMongoDB内部構造の設計.
454 • YARRからMongoDBへの読み出し試験結果アップロード機能.
455 • 試験結果のソート及び閲覧が可能な、Flaskを用いたウェブアプリケーションの開発と実装.

456 モジュールの組み立て、品質試験に向けたデータ管理、そしてローカルデータベースの利点を活かすに
457 は以下のような開発課題が残されていた。

- 458 1. 中央データベースの内部構造の実装(節4.3.1).
459 2. 中央データベースとローカルデータベース間の同期ツールの開発(節4.3.2).
460 3. ローカルデータベースにおける品質試験に特化したデータ管理と機能提供.
461 4. 量産時におけるデータベース操作の流れの確立(節4.3.7).

462 本研究ではこれらの開発課題に対して取り組みを行なった。特に開発項目3に関しては、以下の機能の
463 開発、実装を行なった。

- 464 (i) 品質試験者及びシステムユーザ管理機能及びコメント付加等の各種ユーザ機能(節4.3.3).
465 (ii) 組み立て工程の管理と品質試験結果の選択機能(節4.3.4).
466 (iii) 読み出し試験におけるピクセル解析ツールの開発(節4.3.5).
467 (iv) 読み出し試験結果の検索機能(節4.3.6).

468 なお先行研究で開発したデータ構造やウェブアプリケーションは既に複数の機関で試験運用されていた
469 ため、本研究ではその構造や使用しているパッケージを大きく変更せずに、拡張する形で開発を行なった。

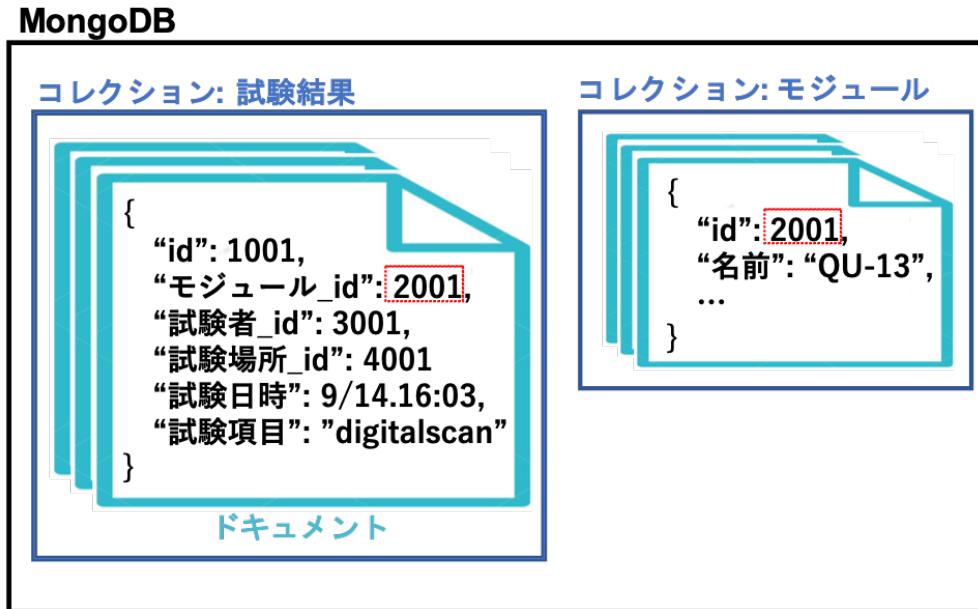


図 4.3 MongoDB の構造の例。図のように MongoDB では JSON 形式でデータを格納する。1 枚の JSON インスタンスをドキュメントと呼び、複数のドキュメントが格納されている枠組みをコレクションと呼ぶ。ドキュメントの構造及びコレクション間の関係等を決めることでデータベースの構造を定義する。ドキュメント間の紐付けは、各ドキュメント内部に ID を持つことで可能となる。図の例では “試験結果” コレクションのドキュメントが {“モジュール_id”:2001} の情報を持つこと、これが “モジュール” コレクションに保存されているドキュメントの ID に対応する。

4.2.3 MongoDB と内部構造

MongoDB とは NoSQL に分類されるデータベースである。MongoDB の内部構造 [28] について簡単に表したものを見ると、一般的な SQLDB のようにテーブル形式ではなく、JSON 形式で情報を格納する。情報を保持している一枚の JSON インスタンスを「ドキュメント」と呼び、「コレクション」と呼ばれる枠に複数のドキュメントが格納されている。各ドキュメントは「ID」と呼ばれるハッシュ値を持っていて、異なるコレクションにおけるドキュメント間の紐付けはこの ID を用いて行う。

ローカルデータベースシステムにおいて、MongoDB を使用する主な利点を以下に示す。

- 各コレクションに格納するドキュメント構造が動的であるため、開発を柔軟に行うことができる。
- JSON 形式でデータを保持するため情報取得の際の整形処理が容易であり、ウェブアプリケーションとの親和性が高い。
- データのキャッシュをメモリ上に置き処理を実行するため、高速な読み書きが可能 [29]。

モジュール及び品質試験に用いる主なコレクションを表 4.1 に示す。また先行研究で設計された読み出し試験結果に関する内部構造を図 4.4 に示す。

表4.1 品質試験に用いる主なコレクション。ローカルデータベースシステムにおいて、MongoDB 内に 2 つのデータベースを設置し、使用する。

データベース名	コレクション名	情報
localdb	component	モジュール情報、FE チップ情報
	childParentRelation	FE チップとモジュールの関係性
	testRun	読み出し試験結果
	componentTestRun	component と testRun の関係性
	user	読み出し試験実施者
	institute	読み出し試験実施場所
	comments	コメント情報
	QC.module.status	各モジュールに対する組み立て工程及び選択された試験結果
	QC.result	品質試験結果
localdbtools	QC.status	組み立て工程及び試験項目
	viewer.user	登録ユーザの情報
	viewer.query	読み出し結果キーワード、検索機能実行時に使用
	viewer.tag.docs	モジュールや試験結果に付けるタグの情報

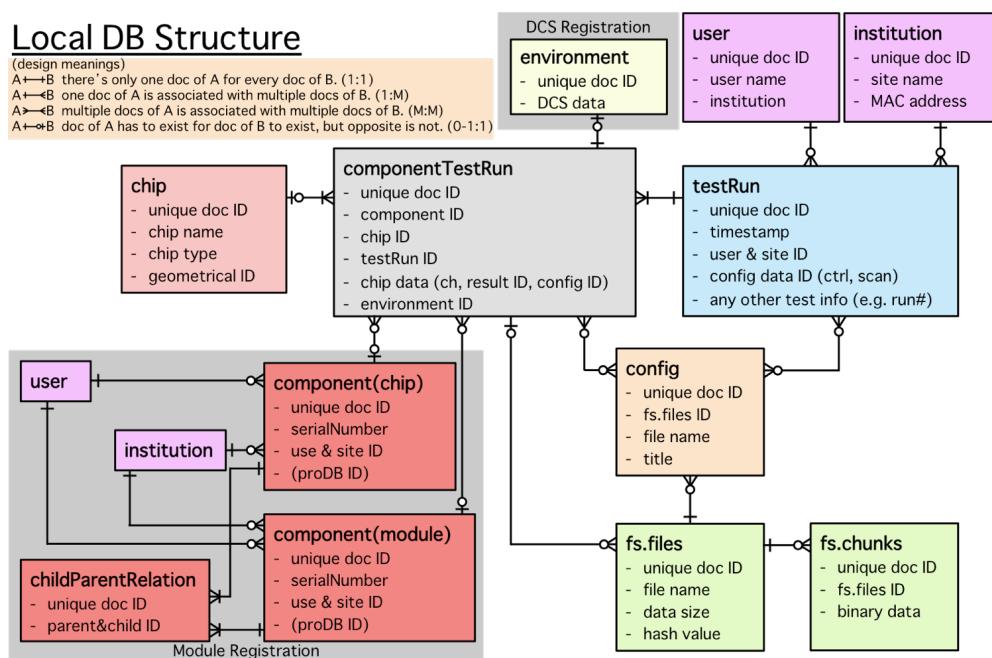


図 4.4 先行研究により設計された読み出し試験結果の MongoDB 内部構造 [27]。それぞれの四角はコレクション、直線は ID によるドキュメント間のリンクを示している。直線が十字になっている場合は対応するドキュメントが 1 つであることを示し、分岐しているものは複数であることを示す。また直線状の白丸は対応するドキュメントが存在しない場合があることを示している。

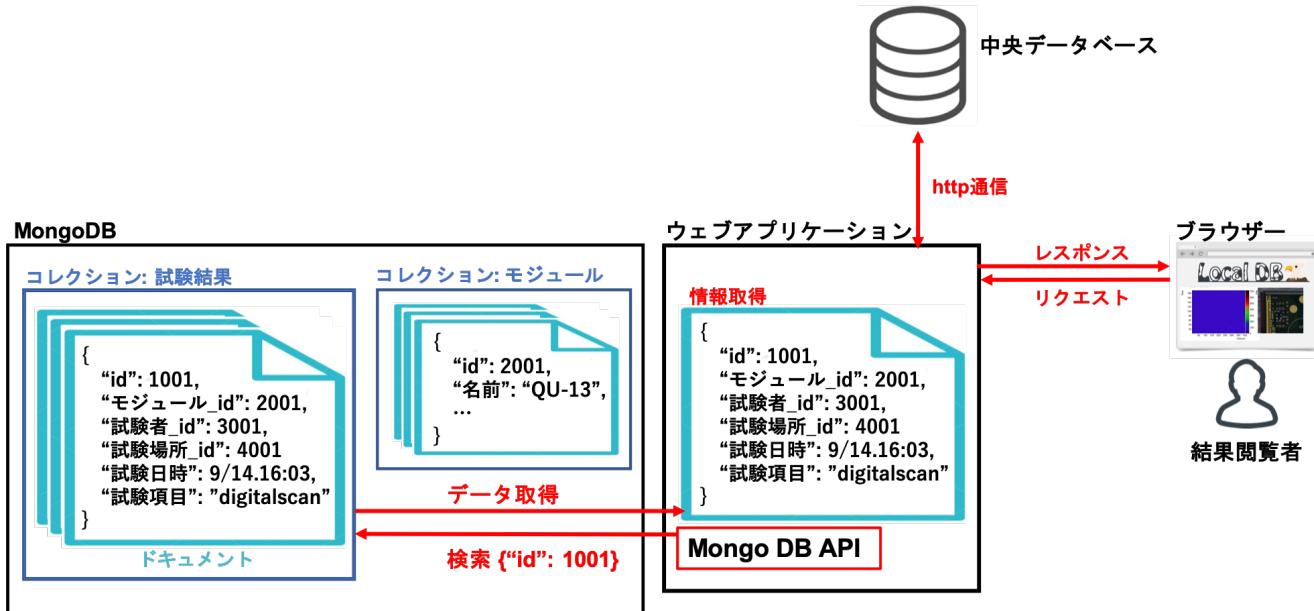


図 4.5 ウェブアプリケーション処理のイメージ。ウェブアプリケーションでは MongoDB と通信を行う API(PyMongo) を用いて、データベースのコレクションに検索をかけることで情報を取得する。取得した情報は整形されたのちブラウザに送信、中央データベースとの同期等の処理に用いられる。

4.2.4 ウェブアプリケーション

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提供している。アプリケーション開発には、Python のウェブフレームワークである Flask を使用している。またアプリケーションにおいて MongoDB との通信に用いる API として、Python ライブライブラリである PyMongo[30] を用いている。アプリケーション処理に特化したイメージを図 4.5 に示す。このようにアプリケーションはデータベースとブラウザ、データベース間のインターフェースとなる。試験結果を迅速に分かりやすく見るシステムを作り、円滑な生産補助や異常結果の早期発見を目的としている。またデータベースの情報管理のみならず、同期ツールや、後述する試験結果解析ツールなどの外部スクリプトの実行、結果取得等、生産時における多くのデータベース操作はこのアプリケーションを用いて行う。

ウェブアプリケーションでは、現在以下の機能を使用することができる。またある品質試験の結果ページを図 4.6 に示す。

- 登録モジュール情報及び品質試験結果の閲覧、解析.
- ローカルデータベースにおけるユーザ管理機能.
- データベース間同期実行機能.

LocalDB TOP / COMPONENTS / TEST

Result: 1348

Information

Component

Result

Scan

Key	Data
runNumber	1348
testType	std_digitalscan
stage	MODULEWIREBONDING
component	20UPGR00000001 20UPGFC9999999
startTime	2020/11/13 19:49:40
finishTime	2020/11/13 19:49:45
user	hokuyama
site	lazulite
targetCharge	-1
targetTot	-1
exec	-r configs/controller/specCfg.json -c db-data/connectivity.json -s configs/scans/rd53a/std_digitalscan.Json -W
stopwatch	analysis: 626 config: 37 processing: 1 scan: 2821
QC	False
environment	True
plots	L1Dist EnMask OccupancyMap
passed	True
qcTest	False
qaTest	False
summary	False

Output Data

Type	Format	Chip	Display	Download
ctrlCfg	json			
dbCfg	json			
siteCfg	json			
userCfg	json			
scanCfg	json			
beforeCfg	json	20UPGFC9999999		
afterCfg	json	20UPGFC9999999		
EnMask	json	20UPGFC9999999		
OccupancyMap	json	20UPGFC9999999		
L1Dist	json	20UPGFC9999999		

PLOT JSROOT

L1Dist plotly

20UPGFC9999999

EnMask plotly

20UPGFC9999999

OccupancyMap plotly

20UPGFC9999999

DCS plot

© 2019 ATLAS Japan ITk and [Tokyo Tech](#) with the great help of LBNL.

Released under the GNU license, please read [LICENSE.TXT](#)

Please refer to the LocalDB official document. [Link](#)

If found any problem, please contact to [Hide Oide](#), [Eunchong Kim](#), [Arisa Kubota](#), [Hiroki Okuyama](#), [Satoshi Kinoshita](#)

図 4.6 品質試験結果ページの例。図は品質試験項目であるデジタル回路読み出しの結果を表している。図の上部に試験情報や設定値、下部に結果のグラフが表示されているのを確認できる。

4.3 本研究における開発項目

498 以下は本研究で開発した項目である。

500 4.3.1 開発項目 1: 中央データベースの内部データ構造の実装

501 中央データベースの枠組みは開発、運用が行われていたが、モジュール情報やその品質試験結果に関する構造は未実装であった。これらの情報を保存するために、構造の定義、実装を行う必要があった。以下の項目の情報構造について、モジュール開発グループの中で国際的に議論を行なながら、構造実装を行った。実装の際には中央データベースが提供しているデータベース操作用の API を用いた。

- 505 1. モジュールの種類とその構成部品 (図 4.7).
 506 2. モジュール組み立て工程と付随する品質試験 (表 4.2).

507 また項目 1 に関して、Quad モジュールに関する例を図 4.8 に示す。

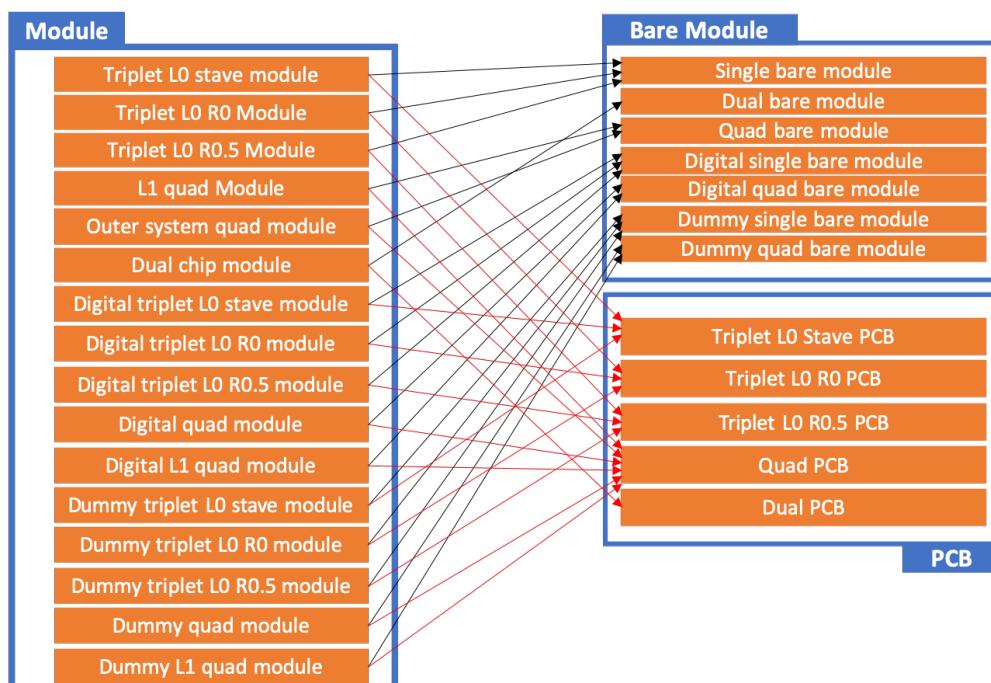


図 4.7 中央データベースにおけるモジュールの種類と構造。中央データベースにモジュールを登録するときの情報として、モジュールの種類、構成部品を図のように実装した。図の左側に実装したモジュールの種類を示しており、Triplet、Quad というように、モジュールの種類ごとに登録できるシステムとなっている。また矢印は構成部品を指しており、各モジュールは対応する Bare Module(ベアモジュール)と PCB(フレキシブル基板)を持つ。DB の中でモジュールと構成部品の紐付けも同時にを行うことができる。

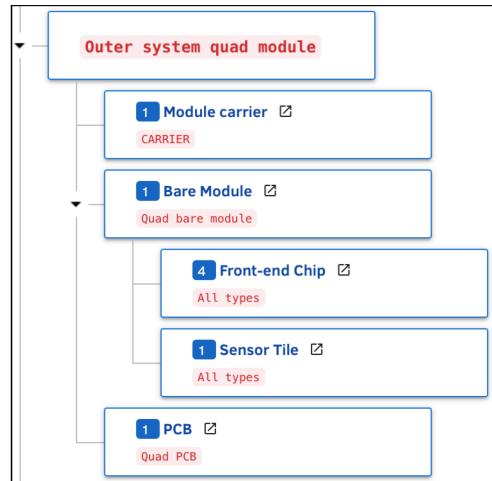


図 4.8 中央データベース内におけるモジュール構造の一例 (Quad モジュール)。例として Outer system quad module の中央データベース内の構造を示している。この種類では構成要素としてそれぞれ対応する種類の Module carrier、Bare Module、PCB を持つことがわかる。さらに Bare Module は FE chip を 4、Sensor を 1 持つことが分かり、Quad モジュールの構造が正しく実装されていることが分かる。

表 4.2 中央データベースにおける組み立て工程と付随するテスト項目。モジュールの組み立て工程及び品質試験をアップロードするため、表のような構造を実装した。データベース内でこの表に沿った組み立て工程の登録、更新、試験結果のアップロードができるようになった。

組み立て項目	付随する組み立て情報及び品質試験項目
1. Bare to PCB assembly	Visual Inspection Metrology Mass measurement Glue information
2. Wirebonding	Visual Inspection Wirebond information (Wirebond pull test) First power up Sensor IV SLDO VI Chip configuration Pixel failure test
3. Wirebond Protection	Visual Inspection Potting information Sensor IV Register test Readout for basic electrical
4. Parylene Coating	Visual Inspection Parylene information Mass measurement Sensor IV Register test Readout for basic electrical Bump bond quality
5. Thermal Cycling	Visual Inspection Thermal cycling info Sensor IV Register test Readout for basic electrical Bump bond quality
6. Burn-in	Visual Inspection Metrology Mass Measurement First power up Sensor IV SLDO VI Chip configuration Pixel failure test
7. Reception	

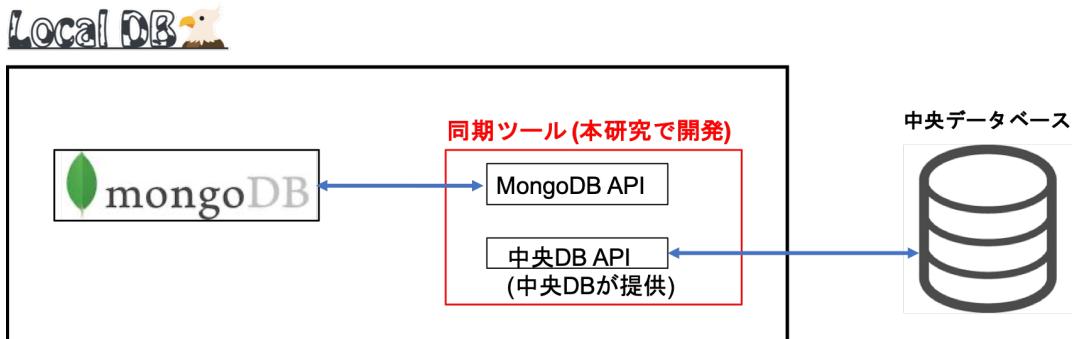


図 4.9 同期ツール処理のイメージ。本研究で開発を行っているのは図の赤線の領域に対応する同期ツールである。このツールは Python を用いて開発しており、処理の中でローカルの MongoDB と通信する API と、中央データベースが開発、提供をしている API を用いることで、2 つのデータベース間の同期を行っている。このとき、中央データベースとの通信は http 通信で行われる。

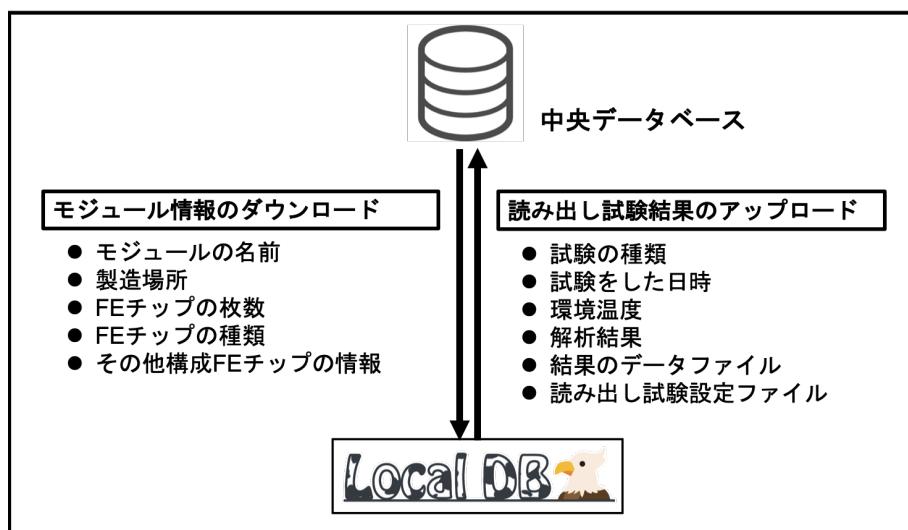


図 4.10 同期機能の概要。本研究ではモジュール情報のダウンロードと読み出し試験結果のアップロード機能を実装した。主に図に示している情報を同期する機能となっている。

508 4.3.2 開発項目 2: データベース同期ツールの開発

509 モジュール情報や品質試験結果の共有のために、中央データベースとローカルデータベースの間で同期
510 が行われる必要がある。これを行うツールを設計、開発を行った。

511 ツールの中では中央データベースが開発、提供している中央データベース通信用 API と、節 4.2.4 で述
512 べたローカルの MongoDB と通信する API の 2 つを用いることで同期を行っている。同期ツール処理の
513 イメージを図 4.9 に示す。

514 特に本研究では以下の機能を実装した。

- 515 ● モジュール及び構成する FE チップ情報のダウンロード機能.
- 516 ● 読み出し試験結果のアップロード機能.

517 これらの機能のイメージを図 4.10 に示す。実装の詳細及び処理時間測定について 7 章で述べる。

518 4.3.3 開発項目 3-(i): ユーザ管理機能及び各種機能

519 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、データベースの機能使用を制限することも必要である。例えば試験結果の閲覧等、データベース自体に変更を加えない機能は権限がなくても使用可能であることが望ましいが、モジュール登録やコメント付加機能等、データベースに書き込みを行う操作はユーザ制限をする必要があると考えられる。

523 これらの理由により、試験者及びデータベース使用者情報の管理機能を開発、実装した。この詳細を以下で述べる。

525 機能概要

526 データベース権限の段階として、管理者、権限付きユーザ、一般ユーザの3段階を設けた。各ユーザが使うことのできる機能を表4.3に示す。ここで一般ユーザはウェブページの閲覧のみができる、特に登録を必要としない。

529 権限付きユーザの機能としてモジュール及び試験結果にコメント、タグをつける機能を実装した。使用したときの様子を図4.11、4.12に示す。

531 ユーザ登録操作

532 表4.3において管理者と権限付ユーザの登録について説明する。

533 データベースシステム導入時に管理者のアカウントを作成する。コマンドプロンプト上で開発したスク534 リプトを実行することで管理者登録が行われる。この際ユーザ名とパスワードを入力する。

535 権限付ユーザについて、全ての品質試験者及びデータベースユーザ機能使用者は管理者によってユーザ536 登録される必要がある。登録はウェブアプリケーションを用いて行い、以下の情報を入力する。

- 537 ● ユーザ名.
- 538 ● 氏名.
- 539 ● 所属機関.
- 540 ● メールアドレス.

541 管理者が登録を完了すると、登録されたメールアドレスに登録完了メールと仮パスワードが届く。この542 メールに従い、ウェブアプリケーション上でユーザがパスワード登録を完了する。

543 このようにメール機能を用いることでパスワード漏洩の防止、管理者操作の削減を目的としている。

表 4.3 ローカルデータベースユーザ権限及び使用機能一覧。ローカルデータベースシステムにおけるユーザとして、管理者、権限付きユーザ、一般ユーザの3つを設けた。全てのユーザがウェブアプリケーションの閲覧をすることができる。管理者、権限付きユーザにはデータベース読み書き権限とウェブアプリケーションログイン権限が与えられ、試験結果のアップロード、アプリケーション上のユーザ機能の実行ができる。また管理者は権限付きユーザを登録することができる。

ユーザ	付加される権限	使用できる機能
管理者	ユーザ管理権限	権限付きユーザ登録機能
	データベース読み書き権限	権限付きユーザが使用可能な機能全般
	ウェブアプリケーションログイン権限	
権限付ユーザ	データベース読み書き権限	試験結果のアップロード
	ウェブアプリケーションログイン権限	中央データベースとの同期機能 その他ウェブアプリケーションの機能（コメント、タグ）
一般ユーザ		モジュール情報及び試験結果の閲覧

The screenshot shows a web application interface for 'LocalDB' under 'TOP / COMPONENTS / TEST'. At the top right, it says 'Asia/Tokyo' and 'okuyama'. Below the header, it displays 'Result: 26'. On the left, there are two sections: 'Information' and 'Component'. On the right, there is a 'Comments' section. It contains a table with columns: Comment, componentType, Name, Institution, and Date. One row in the table is shown with the comment 'Good results!!', component type 'front-end_chip', name 'okuyama', institution 'TokyoTech', and date '2020-11-29 09:24:03.498000'. Below the table is a form with fields for 'comment', 'okuyama', 'institution', and a 'submit' button.

図 4.11 ウェブアプリケーションにおけるコメント機能。権限付きユーザ及び管理者はモジュールや試験結果に対してコメントをすることができる。図のようにページの右側にコメント欄があり、コメントをテキスト形式で記述することができる。

The screenshot shows a 'Scan List' page for 'LocalDB' under 'TOP / COMPONENTS / TEST'. At the top right, it says 'Asia/Tokyo' and 'Sign in'. Below the header, it says 'ITk database for Yarr'. It features a search bar with 'Input keywords', 'Partial match' (radio button), 'Perfect match' (radio button), and a 'Search' button. The main area is titled 'Test Data' and contains a table with columns: Module Name, Chip Name, Test Type, User, Site, Date, Link, and Tag. There are three rows in the table, each corresponding to a test result for 'JohnDoe_0' with 'std_digitalscan' as the test type, 'okuyama' as the user, and 'default_host' as the site. The dates are '2020/10/27 15:40:34', '2020/10/27 15:38:14', and '2020/10/27 15:37:41'. The 'Link' column for all rows contains a blue link labeled 'result page'. The 'Tag' column for the first and third rows contains the word 'anomaly', while the second row contains the word 'good'.

図 4.12 ウェブアプリケーションにおけるタグ機能。権限付きユーザ及び管理者はモジュールや試験結果に対してタグをつけることができる。図は試験結果の一覧ページであり、図の表において一番右の列がつけられたタグを示している。図では“anomaly”や“good”といったタグが付けられていることが分かる。

544 機能の仕組み

545 ユーザ登録の際には内部で以下の2つの処理が行われるように実装した。

- 546 1. MongoDBアカウントの作成、読み書き権限の付与.
 547 2. ウェブアプリケーションで用いるユーザ情報ドキュメントの作成.

548 1の処理を行う理由は、登録ユーザが試験結果をMongoDBにアップロードできるようにするためである。2の情報は、ウェブアプリケーション内でのログイン判断、ユーザの情報保持に使う。この情報を保存するコレクションは表4.1においてviewer.userに保存される。2つの処理について、実際に保存されるドキュメントの例をコード4.1、4.2に示す。

ソースコード4.1 MongoDBアカウント情報を持つドキュメントの例。リスト中の“roles”より、localdbとlocaldbtoolsの読み書き権限が付加されていることが分かる。

```
552 1 {
553 2   "_id" : "localdb.hokuyama",
554 3   "userId" : UUID("fee321eb-83b8-434a-a4a0-fff638b5db36"),
555 4   "user" : "hokuyama",
556 5   "db" : "localdb",
557 6   "credentials" : {
558 7     ...
559 8   },
560 9   "roles" : [
561 10    {
562 11      "role" : "readWrite",
563 12      "db" : "localdb"
564 13    },
565 14    {
566 15      "role" : "readWrite",
567 16      "db" : "localdbtools"
568 17    }
569 18  ]
570 19 }
```

ソースコード4.2 ウェブアプリケーションで扱うユーザ情報を持つドキュメントの例。リスト4.1で示したものとは別に、ウェブアプリケーション内でユーザ情報を扱うためにこのドキュメントを保持する必要がある。ウェブにおいてログインはこのドキュメントの存在確認をもってなされる。パスワードはハッシュ化して保存している。

```
573 1 {
574 2   "_id" : ObjectId("5f0bbe84ef87af2628865de7"),
575 3   "sys" : {
576 4     "rev" : 0,
577 5     "cts" : ISODate("2020-07-13T10:53:07.943Z"),
578 6     "mts" : ISODate("2020-07-13T10:53:07.943Z")
579 7   },
580 8   "username" : "hokuyama",
581 9   "name" : "Hiroki Okuyama",
582 10  "auth" : "readWrite",
583 11  "institution" : "Tokyo Institute of Technology",
584 12  "Email" : "okuyama@hep.phys.titech.ac.jp",
585 13  "password" : "5f4dcc3b5aa765d61d8327deb882cf99"
586 14 }
```

589 4.3.4 開発項目 3-(ii): 組み立て工程の管理と品質試験結果の選択機能

590 ローカルデータベースの中で組み立て工程の管理及び各工程に対応する品質試験の選択機能を実装し
 591 た。組み立て工程の管理に関して、中央データベースに実装した項目をローカルデータベースでも管理
 592 し、モジュールの現工程を保存する。品質試験の選択に関して、ローカルデータベースへアップロードし
 593 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同期する結果を選択する必要がある。これは
 594 不必要な結果を同期せず、中央データベースを簡潔に保つ目的がある。

595 品質試験は各モジュール、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
 596 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は次のものへ自動的に更新する
 597 機能となっている。

598 **概要**

599 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 4.13 に示す。組み立て工程も自動更
 600 新されていることがわかる。

601 **仕組み**

602 コード 4.3、4.4 のようなドキュメントを作成、保存する。コード 4.3 は全てのモジュールに対して共通
 603 のドキュメントであり、組み立て工程と各工程における品質試験項目を記録する。これらの情報は中央
 604 データベースから取得する。この情報を参照することでローカルデータベース内部で組み立て工程の管理
 605 が可能となる。

606 コード 4.4 は各モジュールに対して 1 つ存在し、以下のような情報を保持する。

- 607 ● モジュールの現組み立て工程.
- 608 ● 各工程における選択された品質試験結果の ID.

ソースコード 4.3 組み立て工程及び品質試験一覧情報のドキュメント。ローカルデータベース内に 1 つこのドキュメントを保持し、品質試験結果選択、組み立て工程の更新時にこのドキュメントを参照する。このドキュメントは中央データベースより情報取得して作成する。“stage_flow”の中に組み立て工程のリストを順番通り保持している。“stage_test”の中に各組み立て工程に対応する試験一覧を保持している。

```

609 1 {
610 2   "_id" : ObjectId("5fc89aa232d56b29091fd64d"),
611 3   "sys" : {
612 4     "mts" : ISODate("2020-12-03T07:58:26.310Z"),
613 5     "cts" : ISODate("2020-12-03T07:58:26.310Z"),
614 6     "rev" : 0
615 7   },
616 8   "dbVersion" : 1.01,
617 9   "proddbVersion" : 1.01,
618 10  "stage_flow" : [
619 11    "MODULETOPCB",
620 12    "MODULEWIREBONDING",
621 13    "MODULEWIREBONDPROTECTION",
622 14    "MODULEPARYLENECOATING",
623 15    "MODULETHERMALCYCLING",
624 16    "MODULEBURNIN",
625 17    "MODULERECEPTION"
626 18  ],
627 19  "stage_test" : {
628 20    "MODULETOPCB" : [
629 21      "OPTICAL",
630 22      "GLUE_MODULE_FLEX_ATTACH",
631 23      "MASS",
632 24    ]
633 25  }
634 26}
635 27
636 28
637 29
638 30
639 31
640 32
641 33
642 34
643 35
644 36
645 37
646 38
647 39
648 40
649 41
650 42
651 43
652 44
653 45
654 46
655 47
656 48
657 49
658 50
659 51
660 52
661 53
662 54
663 55
664 56
665 57
666 58
667 59
668 60
669 61
670 62
671 63
672 64
673 65
674 66
675 67
676 68
677 69
678 70
679 71
680 72
681 73
682 74
683 75
684 76
685 77
686 78
687 79
688 80
689 81
690 82
691 83
692 84
693 85
694 86
695 87
696 88
697 89
698 90
699 91
700 92
701 93
702 94
703 95
704 96
705 97
706 98
707 99
708 100
709 101
710 102
711 103
712 104
713 105
714 106
715 107
716 108
717 109
718 110
719 111
720 112
721 113
722 114
723 115
724 116
725 117
726 118
727 119
728 120
729 121
730 122
731 123
732 124
733 125
734 126
735 127
736 128
737 129
738 130
739 131
740 132
741 133
742 134
743 135
744 136
745 137
746 138
747 139
748 140
749 141
750 142
751 143
752 144
753 145
754 146
755 147
756 148
757 149
758 150
759 151
760 152
761 153
762 154
763 155
764 156
765 157
766 158
767 159
768 160
769 161
770 162
771 163
772 164
773 165
774 166
775 167
776 168
777 169
778 170
779 171
780 172
781 173
782 174
783 175
784 176
785 177
786 178
787 179
788 180
789 181
790 182
791 183
792 184
793 185
794 186
795 187
796 188
797 189
798 190
799 191
800 192
801 193
802 194
803 195
804 196
805 197
806 198
807 199
808 200
809 201
810 202
811 203
812 204
813 205
814 206
815 207
816 208
817 209
818 210
819 211
820 212
821 213
822 214
823 215
824 216
825 217
826 218
827 219
828 220
829 221
830 222
831 223
832 224
833 225
834 226
835 227
836 228
837 229
838 230
839 231
840 232
841 233
842 234
843 235
844 236
845 237
846 238
847 239
848 240
849 241
850 242
851 243
852 244
853 245
854 246
855 247
856 248
857 249
858 250
859 251
860 252
861 253
862 254
863 255
864 256
865 257
866 258
867 259
868 260
869 261
870 262
871 263
872 264
873 265
874 266
875 267
876 268
877 269
878 270
879 271
880 272
881 273
882 274
883 275
884 276
885 277
886 278
887 279
888 280
889 281
890 282
891 283
892 284
893 285
894 286
895 287
896 288
897 289
898 290
899 291
900 292
901 293
902 294
903 295
904 296
905 297
906 298
907 299
908 300
909 301
910 302
911 303
912 304
913 305
914 306
915 307
916 308
917 309
918 310
919 311
920 312
921 313
922 314
923 315
924 316
925 317
926 318
927 319
928 320
929 321
930 322
931 323
932 324
933 325
934 326
935 327
936 328
937 329
938 330
939 331
940 332
941 333
942 334
943 335
944 336
945 337
946 338
947 339
948 340
949 341
950 342
951 343
952 344
953 345
954 346
955 347
956 348
957 349
958 350
959 351
960 352
961 353
962 354
963 355
964 356
965 357
966 358
967 359
968 360
969 361
970 362
971 363
972 364
973 365
974 366
975 367
976 368
977 369
978 370
979 371
980 372
981 373
982 374
983 375
984 376
985 377
986 378
987 379
988 380
989 381
990 382
991 383
992 384
993 385
994 386
995 387
996 388
997 389
998 390
999 391
1000 392
1001 393
1002 394
1003 395
1004 396
1005 397
1006 398
1007 399
1008 400
1009 401
1010 402
1011 403
1012 404
1013 405
1014 406
1015 407
1016 408
1017 409
1018 410
1019 411
1020 412
1021 413
1022 414
1023 415
1024 416
1025 417
1026 418
1027 419
1028 420
1029 421
1030 422
1031 423
1032 424
1033 425
1034 426
1035 427
1036 428
1037 429
1038 430
1039 431
1040 432
1041 433
1042 434
1043 435
1044 436
1045 437
1046 438
1047 439
1048 440
1049 441
1050 442
1051 443
1052 444
1053 445
1054 446
1055 447
1056 448
1057 449
1058 450
1059 451
1060 452
1061 453
1062 454
1063 455
1064 456
1065 457
1066 458
1067 459
1068 460
1069 461
1070 462
1071 463
1072 464
1073 465
1074 466
1075 467
1076 468
1077 469
1078 470
1079 471
1080 472
1081 473
1082 474
1083 475
1084 476
1085 477
1086 478
1087 479
1088 480
1089 481
1090 482
1091 483
1092 484
1093 485
1094 486
1095 487
1096 488
1097 489
1098 490
1099 491
1100 492
1101 493
1102 494
1103 495
1104 496
1105 497
1106 498
1107 499
1108 500
1109 501
1110 502
1111 503
1112 504
1113 505
1114 506
1115 507
1116 508
1117 509
1118 510
1119 511
1120 512
1121 513
1122 514
1123 515
1124 516
1125 517
1126 518
1127 519
1128 520
1129 521
1130 522
1131 523
1132 524
1133 525
1134 526
1135 527
1136 528
1137 529
1138 530
1139 531
1140 532
1141 533
1142 534
1143 535
1144 536
1145 537
1146 538
1147 539
1148 540
1149 541
1150 542
1151 543
1152 544
1153 545
1154 546
1155 547
1156 548
1157 549
1158 550
1159 551
1160 552
1161 553
1162 554
1163 555
1164 556
1165 557
1166 558
1167 559
1168 560
1169 561
1170 562
1171 563
1172 564
1173 565
1174 566
1175 567
1176 568
1177 569
1178 570
1179 571
1180 572
1181 573
1182 574
1183 575
1184 576
1185 577
1186 578
1187 579
1188 580
1189 581
1190 582
1191 583
1192 584
1193 585
1194 586
1195 587
1196 588
1197 589
1198 590
1199 591
1200 592
1201 593
1202 594
1203 595
1204 596
1205 597
1206 598
1207 599
1208 600
1209 601
1210 602
1211 603
1212 604
1213 605
1214 606
1215 607
1216 608
1217 609
1218 610
1219 611
1220 612
1221 613
1222 614
1223 615
1224 616
1225 617
1226 618
1227 619
1228 620
1229 621
1230 622
1231 623
1232 624
1233 625
1234 626
1235 627
1236 628
1237 629
1238 630
1239 631
1240 632
1241 633
1242 634
1243 635
1244 636
1245 637
1246 638
1247 639
1248 640
1249 641
1250 642
1251 643
1252 644
1253 645
1254 646
1255 647
1256 648
1257 649
1258 650
1259 651
1260 652
1261 653
1262 654
1263 655
1264 656
1265 657
1266 658
1267 659
1268 660
1269 661
1270 662
1271 663
1272 664
1273 665
1274 666
1275 667
1276 668
1277 669
1278 670
1279 671
1280 672
1281 673
1282 674
1283 675
1284 676
1285 677
1286 678
1287 679
1288 680
1289 681
1290 682
1291 683
1292 684
1293 685
1294 686
1295 687
1296 688
1297 689
1298 690
1299 691
1300 692
1301 693
1302 694
1303 695
1304 696
1305 697
1306 698
1307 699
1308 700
1309 701
1310 702
1311 703
1312 704
1313 705
1314 706
1315 707
1316 708
1317 709
1318 710
1319 711
1320 712
1321 713
1322 714
1323 715
1324 716
1325 717
1326 718
1327 719
1328 720
1329 721
1330 722
1331 723
1332 724
1333 725
1334 726
1335 727
1336 728
1337 729
1338 730
1339 731
1340 732
1341 733
1342 734
1343 735
1344 736
1345 737
1346 738
1347 739
1348 740
1349 741
1350 742
1351 743
1352 744
1353 745
1354 746
1355 747
1356 748
1357 749
1358 750
1359 751
1360 752
1361 753
1362 754
1363 755
1364 756
1365 757
1366 758
1367 759
1368 760
1369 761
1370 762
1371 763
1372 764
1373 765
1374 766
1375 767
1376 768
1377 769
1378 770
1379 771
1380 772
1381 773
1382 774
1383 775
1384 776
1385 777
1386 778
1387 779
1388 780
1389 781
1390 782
1391 783
1392 784
1393 785
1394 786
1395 787
1396 788
1397 789
1398 790
1399 791
1400 792
1401 793
1402 794
1403 795
1404 796
1405 797
1406 798
1407 799
1408 800
1409 801
1410 802
1411 803
1412 804
1413 805
1414 806
1415 807
1416 808
1417 809
1418 810
1419 811
1420 812
1421 813
1422 814
1423 815
1424 816
1425 817
1426 818
1427 819
1428 820
1429 821
1430 822
1431 823
1432 824
1433 825
1434 826
1435 827
1436 828
1437 829
1438 830
1439 831
1440 832
1441 833
1442 834
1443 835
1444 836
1445 837
1446 838
1447 839
1448 840
1449 841
1450 842
1451 843
1452 844
1453 845
1454 846
1455 847
1456 848
1457 849
1458 850
1459 851
1460 852
1461 853
1462 854
1463 855
1464 856
1465 857
1466 858
1467 859
1468 860
1469 861
1470 862
1471 863
1472 864
1473 865
1474 866
1475 867
1476 868
1477 869
1478 870
1479 871
1480 872
1481 873
1482 874
1483 875
1484 876
1485 877
1486 878
1487 879
1488 880
1489 881
1490 882
1491 883
1492 884
1493 885
1494 886
1495 887
1496 888
1497 889
1498 890
1499 891
1500 892
1501 893
1502 894
1503 895
1504 896
1505 897
1506 898
1507 899
1508 900
1509 901
1510 902
1511 903
1512 904
1513 905
1514 906
1515 907
1516 908
1517 909
1518 910
1519 911
1520 912
1521 913
1522 914
1523 915
1524 916
1525 917
1526 918
1527 919
1528 920
1529 921
1530 922
1531 923
1532 924
1533 925
1534 926
1535 927
1536 928
1537 929
1538 930
1539 931
1540 932
1541 933
1542 934
1543 935
1544 936
1545 937
1546 938
1547 939
1548 940
1549 941
1550 942
1551 943
1552 944
1553 945
1554 946
1555 947
1556 948
1557 949
1558 950
1559 951
1560 952
1561 953
1562 954
1563 955
1564 956
1565 957
1566 958
1567 959
1568 960
1569 961
1570 962
1571 963
1572 964
1573 965
1574 966
1575 967
1576 968
1577 969
1578 970
1579 971
1580 972
1581 973
1582 974
1583 975
1584 976
1585 977
1586 978
1587 979
1588 980
1589 981
1590 982
1591 983
1592 984
1593 985
1594 986
1595 987
1596 988
1597 989
1598 990
1599 991
1600 992
1601 993
1602 994
1603 995
1604 996
1605 997
1606 998
1607 999
1608 1000
1609 1001
1610 1002
1611 1003
1612 1004
1613 1005
1614 1006
1615 1007
1616 1008
1617 1009
1618 1010
1619 1011
1620 1012
1621 1013
1622 1014
1623 1015
1624 1016
1625 1017
1626 1018
1627 1019
1628 1020
1629 1021
1630 1022
1631 1023
1632 1024
1633 1025
1634 1026
1635 1027
1636 1028
1637 1029
1638 1030
1639 1031
1640 1032
1641 1033
1642 1034
1643 1035
1644 1036
1645 1037
1646 1038
1647 1039
1648 1040
1649 1041
1650 1042
1651 1043
1652 1044
1653 1045
1654 1046
1655 1047
1656 1048
1657 1049
1658 1050
1659 1051
1660 1052
1661 1053
1662 1054
1663 1055
1664 1056
1665 1057
1666 1058
1667 1059
1668 1060
1669 1061
1670 1062
1671 1063
1672 1064
1673 1065
1674 1066
1675 1067
1676 1068
1677 1069
1678 1070
1679 1071
1680 1072
1681 1073
1682 1074
1683 1075
1684 1076
1685 1077
1686 1078
1687 1079
1688 1080
1689 1081
1690 1082
1691 1083
1692 1084
1693 1085
1694 1086
1695 1087
1696 1088
1697 1089
1698 1090
1699 1091
1700 1092
1701 1093
1702 1094
1703 1095
1704 1096
1705 1097
1706 1098
1707 1099
1708 1100
1709 1101
1710 1102
1711 1103
1712 1104
1713 1105
1714 1106
1715 1107
1716 1108
1717 1109
1718 1110
1719 1111
1720 1112
1721 1113
1722 1114
1723 1115
1724 1116
1725 1117
1726 1118
1727 1119
1728 1120
1729 1121
1730 1122
1731 1123
1732 1124
1733 1125
1734 1126
1735 1127
1736 1128
1737 1129
1738 1130
1739 1131
1740 1132
174
```

```

633 24     "METROLOGY"
634 25   ],
635 26   "MODULEWIREBONDING" : [
636 27     "WIREBONDING",
637 28     "OPTICAL",
638 29     "SENSOR_IV",
639 30     "PIXEL_FAILURE_TEST",
640 31     "SLDO_VI",
641 32     "WIREBOND",
642 33     "CHIP_CONFIGURATION"
643 34   ],
644 35   "MODULEWIREBONDPROTECTION" : [
645 36     "OPTICAL",
646 37     "POTTING",
647 38     "MASS",
648 39     "READOUT_IN_BASIC_ELECTRICAL_TEST",
649 40     "SENSOR_IV",
650 41     "REGISTER_TEST"
651 42   ],
652 43   ...
653 44 },
654 45 ...
655 46 }

```

ソースコード 4.4 モジュールの組み立て工程及び品質試験結果管理のためのドキュメント例。各モジュールにおいて現在の組み立て工程及び選択された品質試験結果がこのドキュメントに保存される。ドキュメント内の“currentStage”に現工程を保持する。また選択した試験結果の ID を“QC_results”に、組み立て工程ごとに保持する。

```

657 {
658   "_id" : ObjectId("5fc4be4c12a45922a91b0e75"),
659   "sys" : {
660     "mts" : ISODate("2020-11-30T09:41:32.411Z"),
661     "cts" : ISODate("2020-11-30T09:41:32.411Z"),
662     "rev" : 0
663   },
664   "dbVersion" : 1.01,
665   "proddbVersion" : 1.01,
666   "component" : "5fa79114e615fa000a1a5976",
667   "currentStage" : "MODULEWIREBONDPROTECTION",
668   "latestSyncedStage" : "MODULEWIREBONDING",
669   "status" : "created",
670   "rework_stage" : [],
671   "QC_results" : {
672     "MODULETOPCB" : {
673       "OPTICAL" : "5fc4c2cfb6c93d451e2c9ac1",
674       "GLUE_MODULE_FLEX_ATTACH" : "-1",
675       "MASS" : "5fc4c2da27766dc6e89c024f",
676       "METROLOGY" : "5fc4c2eaf1f19d9cb5859f00"
677     },
678     "MODULEWIREBONDING" : {
679       "WIREBONDING" : "-1",
680       "OPTICAL" : "5fc4c4c8b7d0c86912b4958f",
681       "SENSOR_IV" : "5fc4c59e9e283a57ccaa1088",
682       "PIXEL_FAILURE_TEST" : "5fc342f6e9f1f5eafedfb92",
683       "SLDO_VI" : "-1",
684       "WIREBOND" : "-1",
685       "CHIP_CONFIGURATION" : "-1"
686     },
687     "MODULEWIREBONDPROTECTION" : {
688       "OPTICAL" : "-1",
689       "POTTING" : "-1",
690       "MASS" : "-1",
691       "READOUT_IN_BASIC_ELECTRICAL_TEST" : "-1",
692       "SENSOR_IV" : "-1",
693       "REGISTER_TEST" : "-1"
694     },
695     ...
696   },
697   ...
698 }

```

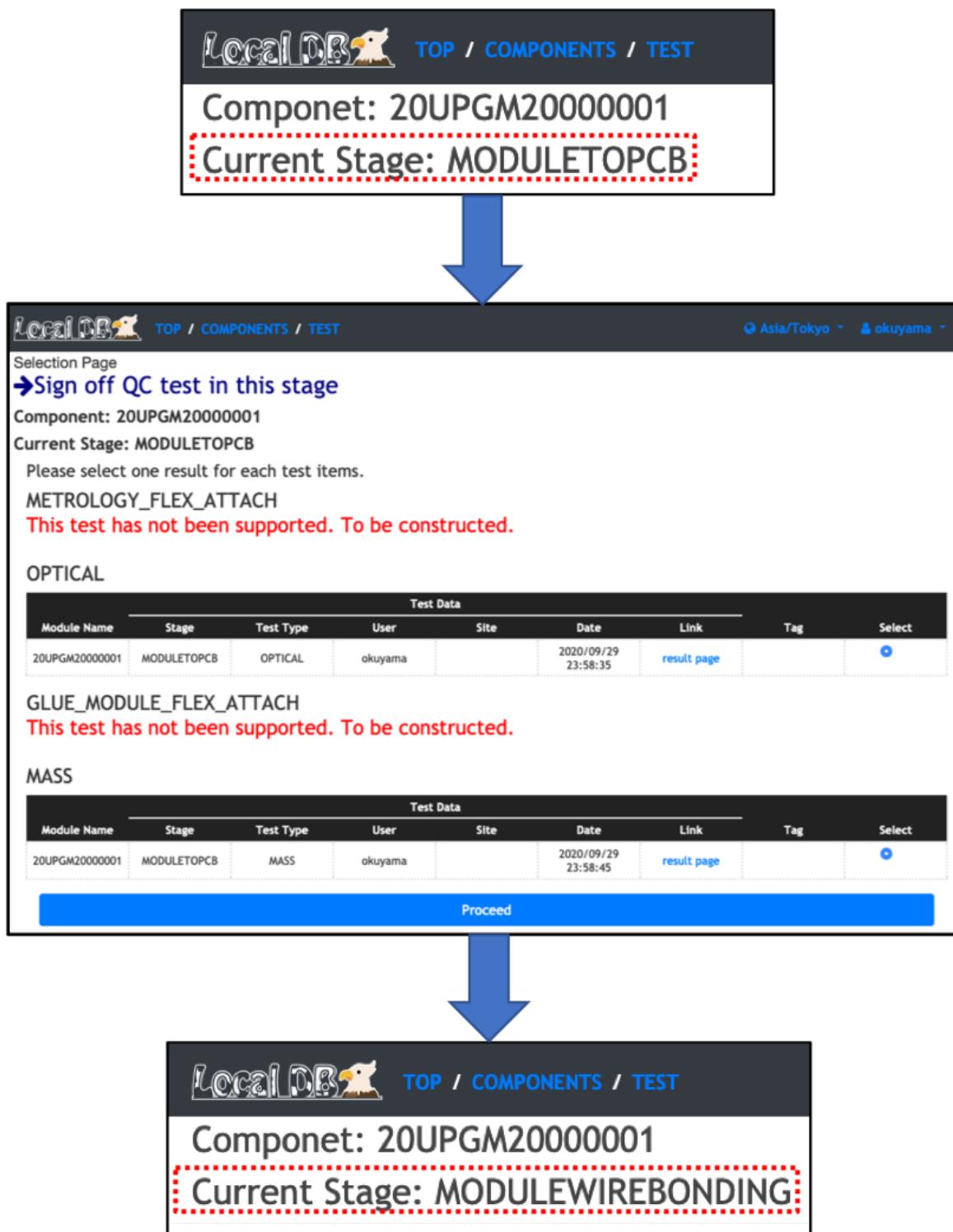


図4.13 結果選択画面及び組み立て工程表示の例。図の上部で組み立て工程が“MODULETOPCB”である。図の中部において品質試験結果選択処理を行なっており、この図では“OPTICAL”と“MASS”的結果を選択している。この時、ローカルデータベース内部では選択された結果にタグ付けがなされる。これらの結果は中央データベースと同期される。また結果選択後は組み立て工程が自動的に更新される。図の下部では”MODULEWIREBONDING”になっていることが分かる。

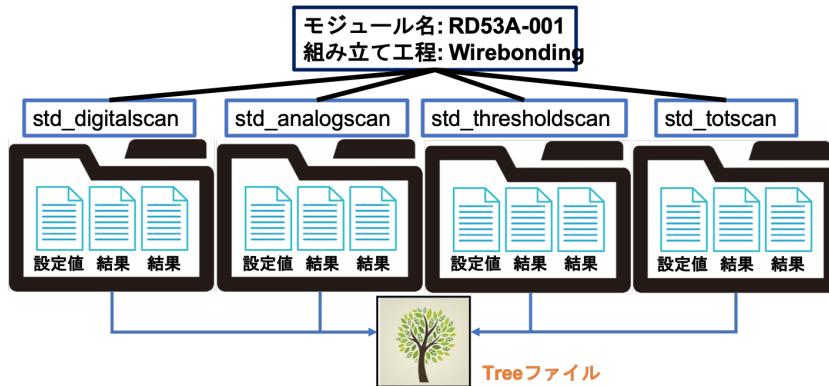


図 4.14 ピクセル解析ツールにおけるファイル統合処理のイメージ。YARR の出力ファイル及びディレクトリは std_digitalscan(デジタル回路読み出し) や std_thresholdscan(Threshold 測定) というように読み出し項目ごとである。ピクセル解析ツールでは、図のようにあるモジュールに関連する結果ファイルを統合し、ピクセルごとに行う解析処理を簡易化する狙いがある。

4.3.5 開発項目 3-(iii): 読み出し試験結果におけるピクセル解析ツール

節 3.2.6 で述べたように、読み出し試験では各ピクセルが正常に機能しているのかを調べるためにピクセル解析を行う。これを円滑に行うために、ピクセル解析ツールを開発した。また開発した解析ツールをローカルデータベースシステムに組み込んだ。このツールについての詳細を以下に示す。

概要

YARR で読み出し試験を行った場合、結果ファイル及びディレクトリは試験項目ごとにわかれ生成される。また各結果ファイルにはモジュール上の全ピクセル結果が JSON の形で保存されている。

一方、ピクセル解析ではあるピクセルに着目し、複数の試験結果を同時に参照する必要がある。この解析を行う場合に、ピクセルは 192×400 個あるため、ピクセルごとに複数の試験結果ファイルの読み込み、値取得の処理を行うのは計算コストがかかり、非効率である。例えば参照したい試験結果ファイルが 5 個あるとすると $(192 \times 400) \times 5$ 回のファイル読み込み、値取得の処理を行うことになる。

そこで、開発した解析ツールでは複数の結果ファイルを 1 つに統合し、ピクセルごとの解析処理を単純化する役割を担っている。開発には Python と C++ を用いた。また CERN が提供している解析フレームワークである ROOT[31] を使用し、いくつかの試験結果統一ファイルの作成に、ROOT 内部機能である Tree を使用した。このファイル統合処理のイメージを図 4.14 に示す。また実際に作った Tree ファイルと、データ構造のイメージを図 4.15 に示す。

ROOT 内部には、条件を定義し選別を行う機能が充実しているため、不良ピクセルを選別する解析に適しているパッケージであると考えた。

ツールの内部構造と処理の流れ

開発したツールは、主に以下で説明する 3 つの実行ファイルで構成される。それぞれの役割について説明する。

getData.py (Python)

データベースから対象となるデータファイルを取得、キャッシュファイルとしてサーバー上の一時

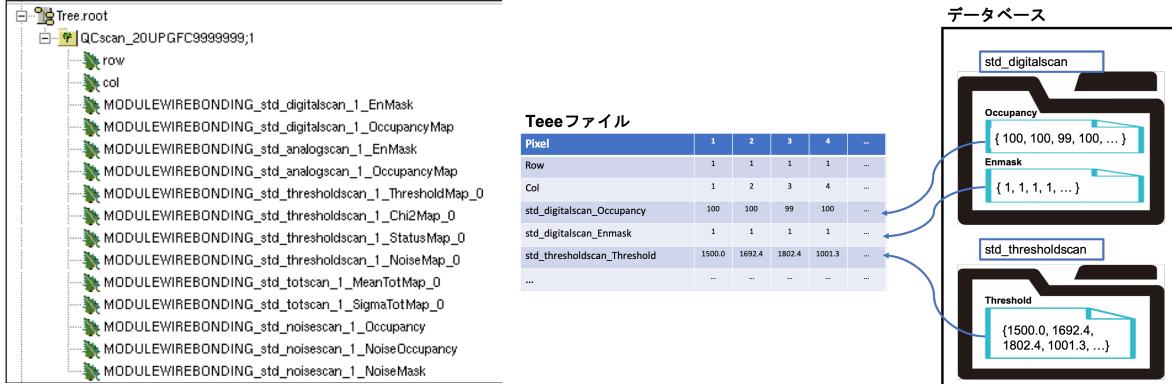


図 4.15 Tree ファイルとそのデータ保持のイメージ。実際にこのツールを用いて作った Tree ファイルの内部構造の様子(左)とそのデータ保持のイメージ(右図)を示す。Tree ファイルでは、右図のように 1 つの表に試験結果をまとめている。各行が std_digitalscan といった各読み出し試験項目における 1 つの測定値に対応し、各列が 1 ピクセルに対応する。モジュール上の行列 (Row, Col) の番号を表の上部に持つておくことで、モジュール上におけるピクセルの位置情報を保持する。ピクセル解析ではあるピクセルに着目し複数の試験結果を同時に参照する必要がある。右図のように複数結果を 1 つのテーブル上に保持し、各列を参照することでピクセルごとに解析を行う。

```

723     ディレクトリに保存。
724     makeTree (C++)
725         getData.py を用いて生成されたキャッシュファイルを読み込み、Tree ファイルを作成。
726     analysis (C++)
727         作成した Tree ファイルを読み込みピクセル解析を実行、結果値やプロットを出力。
728     処理の流れのイメージを図 4.16 に示す。データベースとの通信に関しては MongoDB や現システムと
729     の親和性を考慮し、Python を使用した。Tree ファイル作成やその後の解析処理のスクリプトは、ROOT
730     を使用する観点から C++ を使用した。またピクセル解析以外の解析処理の作成を見越して、Tree 作成
731     部と解析処理部のファイルは分割した。

```

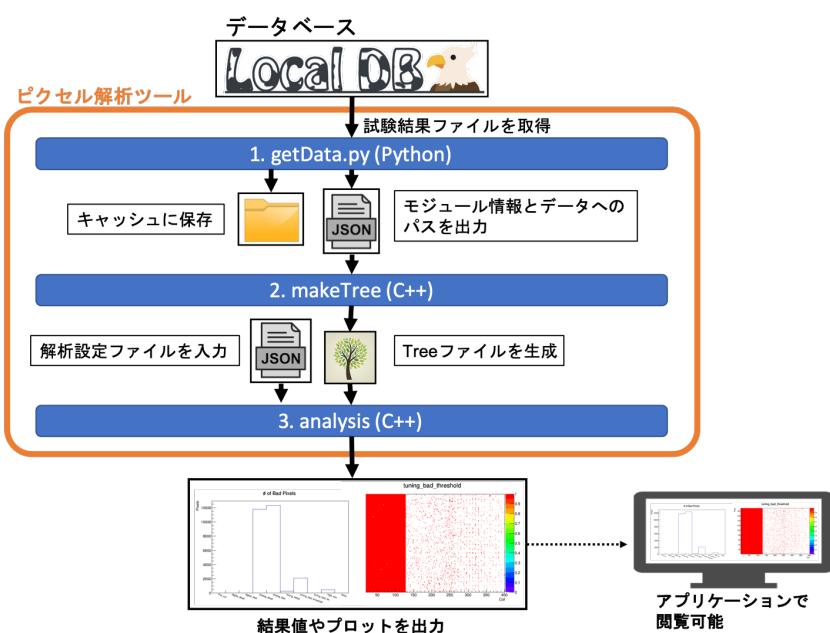
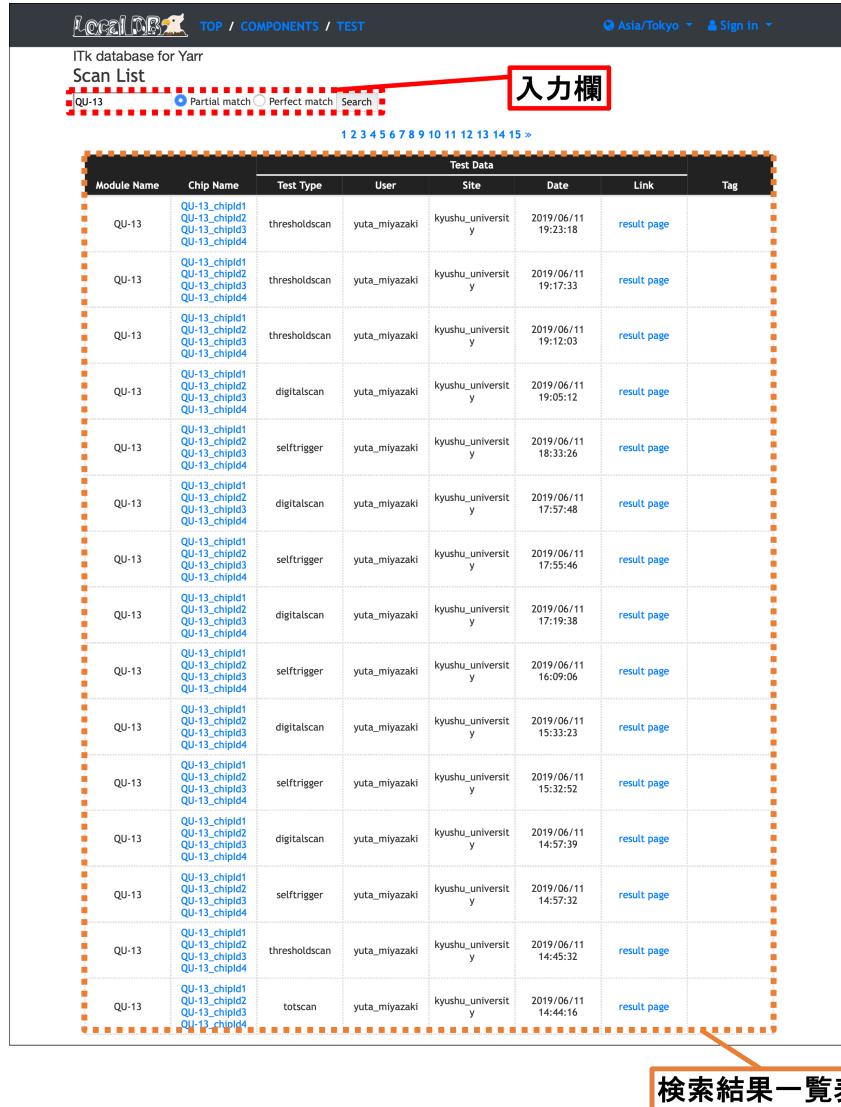


図 4.16 ピクセル解析ツールの処理の流れ。ピクセル解析ツールは、図のように 3 つの実行ファイルにより構成される。getData.py(Python) にてデータベースから結果の取得を行い、makeTree(C++) にて Tree ファイルの作成、analysis(C++) にてピクセル解析及び結果のプロットが output される。



LocalDB TOP / COMPONENTS / TEST

ITk database for Yarr

Scan List

QU-13 Partial match Perfect match Search

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 >

Test Data

Module Name	Chip Name	Test Type	User	Site	Date	Link	Tag
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	thresholdscan	yuta_miyazaki	kyushu_universit y	2019/06/11 19:23:18	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	thresholdscan	yuta_miyazaki	kyushu_universit y	2019/06/11 19:17:33	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	thresholdscan	yuta_miyazaki	kyushu_universit y	2019/06/11 19:12:03	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	digitalscan	yuta_miyazaki	kyushu_universit y	2019/06/11 19:05:12	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	selftrigger	yuta_miyazaki	kyushu_universit y	2019/06/11 18:33:26	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	digitalscan	yuta_miyazaki	kyushu_universit y	2019/06/11 17:57:48	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	selftrigger	yuta_miyazaki	kyushu_universit y	2019/06/11 17:55:46	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	digitalscan	yuta_miyazaki	kyushu_universit y	2019/06/11 17:19:38	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	selftrigger	yuta_miyazaki	kyushu_universit y	2019/06/11 16:09:06	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	digitalscan	yuta_miyazaki	kyushu_universit y	2019/06/11 15:33:23	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	selftrigger	yuta_miyazaki	kyushu_universit y	2019/06/11 15:32:52	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	digitalscan	yuta_miyazaki	kyushu_universit y	2019/06/11 14:57:39	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	selftrigger	yuta_miyazaki	kyushu_universit y	2019/06/11 14:57:32	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	thresholdscan	yuta_miyazaki	kyushu_universit y	2019/06/11 14:45:32	result page	
QU-13	QU-13_chipd1 QU-13_chipd2 QU-13_chipd3 QU-13_chipd4	totscan	yuta_miyazaki	kyushu_universit y	2019/06/11 14:44:16	result page	

図 4.17 ウェブアプリケーションにおける検索機能の様子。図は検索結果一覧表示のページである。図の上部に入力欄があり（赤破線）、ここにキーワードを入力し検索を実行する。図の例では“QU-13”と入力し検索しており、モジュール名「QU-13」の試験結果が一覧表示されていることが分かる。

検索結果一覧表

4.3.6 開発項目 3-(iv): 読み出し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、閲覧す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 4.17 に示す。キーワードを入力し、検索することができる仕組みとなっていて、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。生産に向けて、検索にかかる処理時間測定を行った。検索機能実装方法の詳細と処理時間についての詳細は 6 章で述べる。現在は单一キーワード検索の他に、以下の機能を実装している。

- 完全一致、部分一致検索。
- AND、OR 検索。

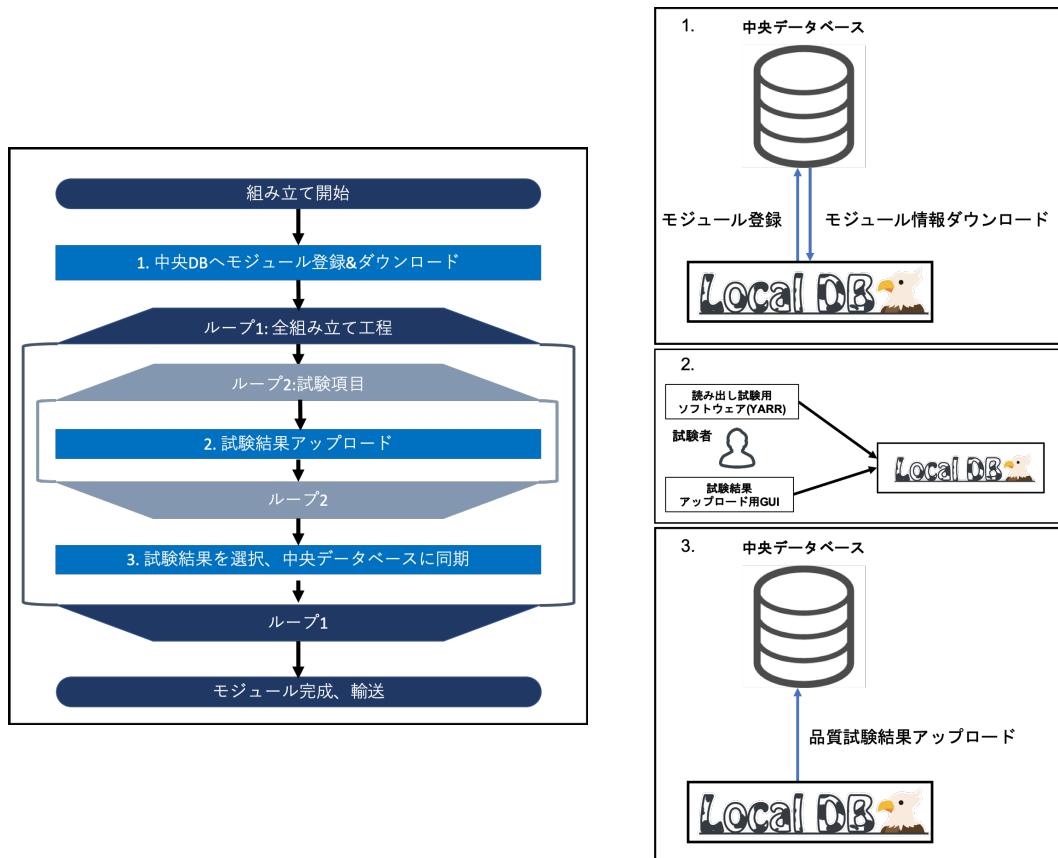


図 4.18 各モジュールにおけるデータベース操作の流れ。初めに中央データベースにモジュール登録及びローカルデータベースへモジュール情報のダウンロードを行う(処理 1)。その後、ダウンロードしたモジュールに対して組み立て工程に応じた試験結果を取得、ローカルデータベースへアップロードする(処理 2)。各組み立て工程の終わりに試験結果の選択を行い、中央データベースに試験結果を同期する(処理 3)。

741 4.3.7 開発項目 4: 量産時におけるデータベース操作の流れの確立

742 量産時におけるデータベース操作の流れを確立した。以下に従い、モジュール組み立て時におけるデータ
743 管理がなされる。

- 744 中央データベースへモジュール登録及び登録情報のダウンロード。
- 745 2で登録したモジュールに対して紐つける形で、品質試験結果をローカルデータベースへアップ
746 ロード。
- 747 3.組み立て工程毎に品質試験結果の選択を行い、中央データベースへアップロード。

748 流れのイメージを図 4.18 に示す。品質試験結果の中央データベースへのアップロードは各組み立て工
749 程毎に行う。ローカルデータベース内で品質試験結果を組み立て工程毎にまとめて扱い、各モジュールの
750 現組み立て工程を正確に管理する目的がある。

751 全組み立て工程が終了すると、モジュールの情報及び品質試験結果が全て中央データベースへ同期され
752 ている状態となる。

753 このデータベース操作の流れの確立に向けて、2020年2月にCERNに行なったシステムチュートリアル
754 でこの一連の流れを扱った。チュートリアルは私が主体となって主催し、使用デバイス、ソフトウェ

755 アの製作、セットアップ、ドキュメント製作、チュートリアル本番時のチーチャー等を担当した。この
756 チュートリアルを通してユーザに対する操作方法の周知と機能確認を行うことができた。チュートリアル
757 の概要を付録 C に記す。またシステムのドキュメント [32] を作成し、その中に具体的なデータベース操
758 作の流れを記述している。このように量産時に各機関が適切な流れでデータ管理ができるようなサポート
759 を行っている。

760 データベース操作の流れの試験として、実際に品質試験のデモンストレーションを行い、各機能が正常
761 に動作するのかの確認を行なった。詳細を 5 章で述べる。

762 第5章

763 品質試験のデモンストレーション

764 品質試験のデモンストレーションを行った。今回のデモンストレーションでは、品質試験項目として部分的に読み出し試験を行い、データベース機能や試験の流れを確認した。この章の前半ではデモンストレーションで使用したソフトウェア、ハードウェアについて説明し、後にデモンストレーションの内容、機能確認について述べる。

768 5.1 使用したソフトウェア

769 使用したソフトウェアを以下に示す。また、これらソフトウェアの概要を図 5.1 に示す。

- 770 • YARR (commit:6b3ffe92).
- 771 • MongoDB (version: v4.2.6).
- 772 • ローカルデータベース用ウェブアプリケーション (tag: ldbtoolv1.4).
- 773 • 中央データベースとの同期ツール (tag: ldbtoolv1.4).
- 774 • ピクセル解析ツール (tag: v1.0.2).
- 775 • 時系列データ用データベース (InfluxDB[33](version: 1.8.0)).
 - 776 – 時系列情報に特化したデータベース。このシステムにおいては温度、電圧など DCS 情報を時
 - 777 間情報と共に保存、管理するために用いる。
- 778 • InfluxDB 解析ソフト (Grafana[34](version: 5.1.0)).
 - 779 – InfluxDB に保存された情報の解析、閲覧に用いる。ウェブブラウザー上で DCS 情報を閲覧
 - 780 することができる。
- 781 • 電源操作用ソフト。
 - 782 – モジュールに電圧を供給するために、電源を遠隔で操作する。また電圧、電流値を取得し、
 - 783 InfluxDB にアップロードする。CERN で開発されている PySerial[35](commit:0d14fcdb) を
 - 784 改良し作成した。
- 785 • 温度読み出し用ソフト。
 - 786 – GPIO 通信により取得できる ADC 値を取得、温度に変換し InfluxDB へアップロードする処
 - 787 理を実行する。RaspberryPi 上で処理を行う。

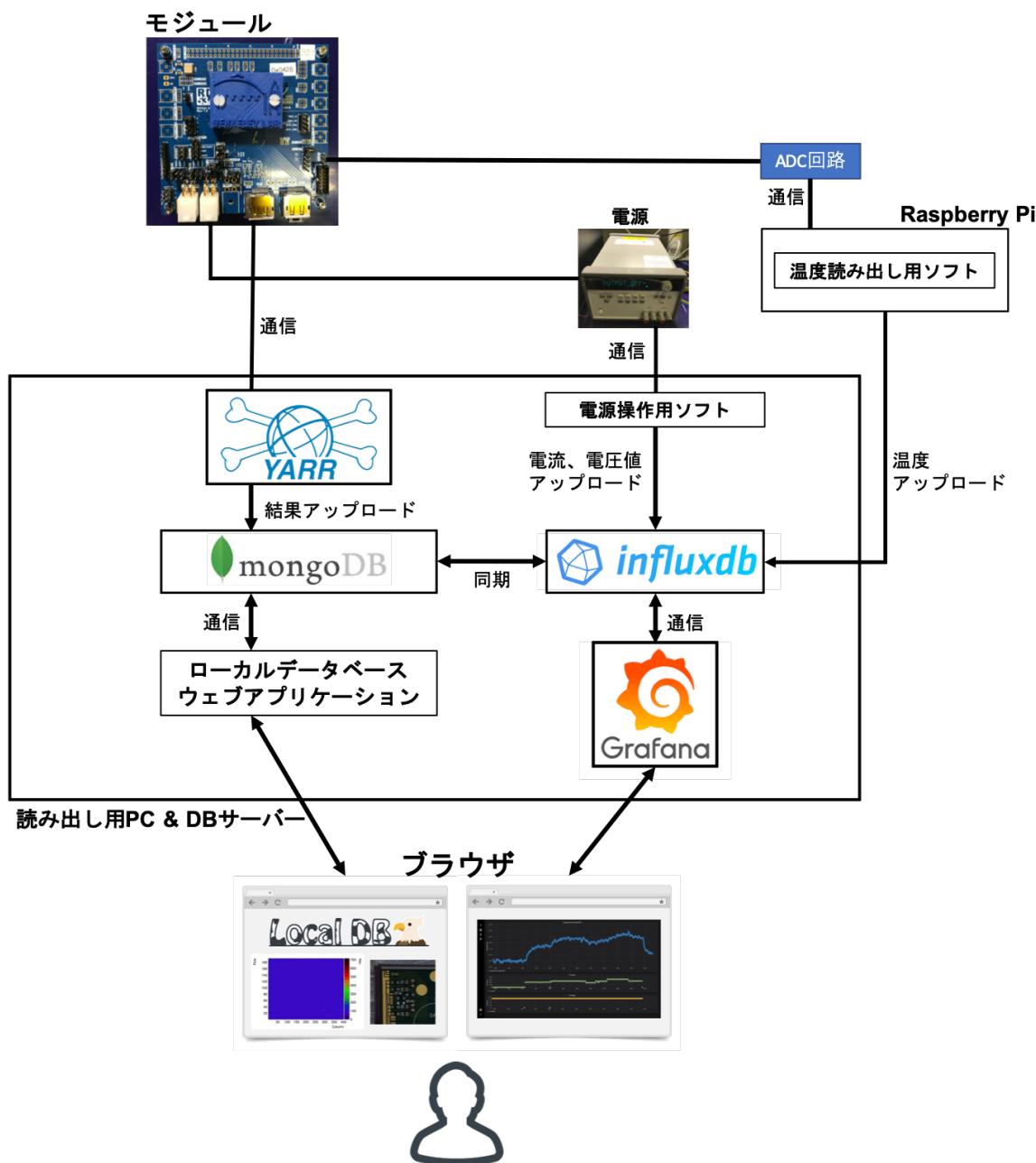


図 5.1 読み出し試験に用いるソフトウェアの概要。FE チップの読み出しとそのデータ通信は YARR を用いて行われる。試験結果は MongoDB にアップロードされ、試験者はウェブアプリケーションを通じて結果を確認することができる。電源操作用ソフトを用いて電源のスイッチ、電圧、電流値の取得がなされ、取得値は InfluxDB に保存される。モジュール付属のサーミスタ読み出しシステムを用いて FE チップ付近の温度を読み出し、InfluxDB に保存する。InfluxDB に保存された DCS データは Grafana を用いてブラウザ上で確認することができる。また読み出し試験に関わる DCS 情報は MongoDB に同期されるため、ローカルデータベースアプリケーションを通して確認ができる。

788 5.2 使用したハードウェア

789 使用したハードウェアについて、以下に詳細を記す。

790 5.2.1 各装置

791 RD53A シングルモジュール (RD53A Single Chip Card, SCC)[36]

792 今回読み出しに使うモジュールとして、研究室で所有している RD53A シングルモジュール (**RD53A**
793 **Single Chip Card, SCC**) を使用した。SCC は試験用に作られた FE チップを一枚搭載するモジュー
794 ルである。また今回使用したものはシリコンセンサーを持たない。SCC は FE チップ電源端子、データ
795 転送端子をもち、読み出しを行う際はそれぞれ配線をする。FE チップ付近には NTC サーミスタを搭載
796 していて、ボード上の端子からその抵抗値を取得することで温度を測定することができる。図 5.2 に写真
797 を示す。

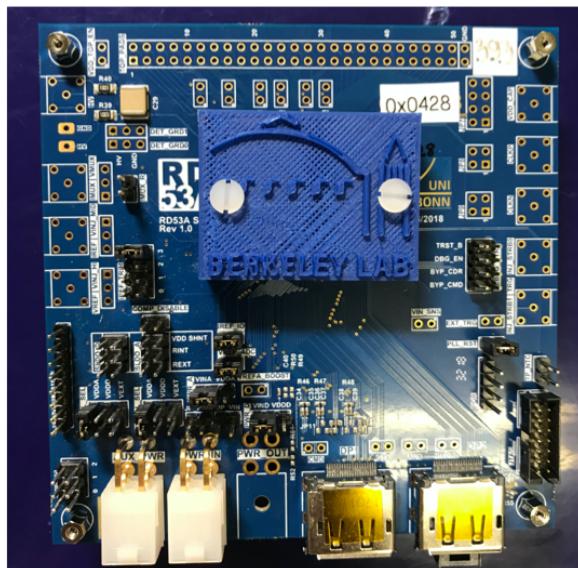


図 5.2 RD53A シングルモジュール (SCC)[36]。RD53A の FE チップを一枚搭載したモジュールである。図の中心部の青いカバーで囲われた中に FE チップが設置されている。このボード上にはデータ転送端子、電源端子、サーミスタ抵抗取得端子が設置されており、これらの端子に対応する配線をすることで読み出し試験のセットアップを組む。

798 モジュールサーミスタ温度読み出しシステム

799 モジュール付属サーミスタの抵抗値を取得し温度を測定するために、ADC、Raspberry Pi を用いた温
800 度読み出しシステムを作成した。このシステムの中で扱った装置を表 5.1 に、回路図、実際に配線した様
801 子を図 5.3 に示す。サーミスタの抵抗値に対応する ADC 値を温度に変換することで読み出しを行ってい
802 る。ADC 値は GPIO 通信 [37] により取得している。

803 電源

804 モジュールの FE チップに対する電圧供給に KEYSIGHT の E3646A 60W デュアル出力電源 [40](図
805 5.4) を用いた。

表 5.1 溫度読み出しシステムに使用した装置一覧。

装置	機種
ADC	MCP3002[38]
RaspberryPi	Raspberry Pi 3 Model B Plus Rev 1.3[39]

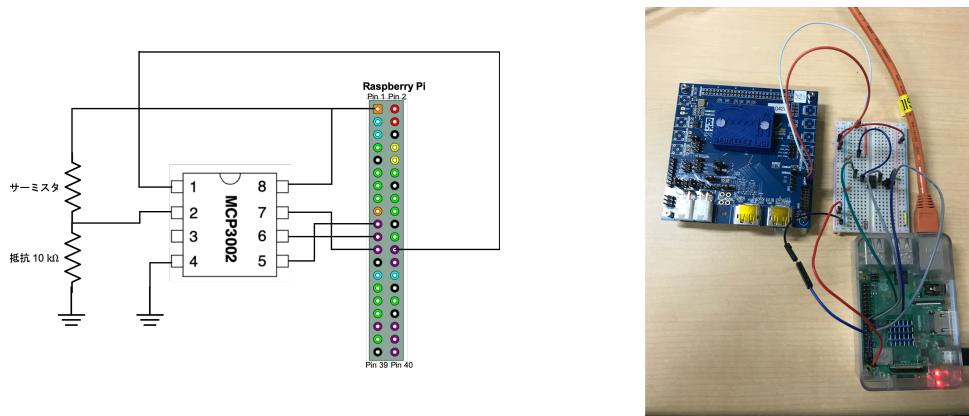


図 5.3 モジュール付属サーミスタを用いた温度読み出し回路。図は回路図（左図）と実際に配線して読み出しを行っている様子（右図）を示している。10 kΩ 抵抗、MCP3002、Raspberry Pi を用いて読み出し回路を作成した。抵抗と MCP3002 は右図のようにブレッドボード上に設置した。ADC と RaspberryPi は GPIO 通信 [37] を行うことで、ADC 値を取得している。



図 5.4 用いた電源。FE チップ電圧供給のための電源として KEYSIGHT の E3646A 60W デュアル出力電源 [40] を用いた。

806 FPGA ボード

807 FPGA ボードに XpressK7[41] を用いた。YARR ファームウェアを FPGA 上にプログラムし通信を行なった。XpressK7 を図 5.5 に示す。

809 FMC-DisplayPort 変換カード

810 FPGA ボードには FMC 端子がついていて、モジュールと通信を行うためには FMC-DisplayPort 変
811 換カードが必要となる。使用した変換カード（通称:Ohio カード）を図 5.6 に示す。

812 PC

813 YARR ソフトウェアをインストールし、読み出しを行なった PC の性能を表 5.2 に示す。



図 5.5 使用した FPGA ボード (XpressK7[41])。中央に FMC 端子、右側に FPGA チップ、下側に PCI Express が配置されている。FPGA チップの上にはファンをついているため、この図では確認できない。



図 5.6 使用した FMC-DisplayPort 変換カード。図の右側に FMC 端子、左側に miniDisplayPort が 4 つ配置されており、FMC 端子を 4 つの mini-DisplayPort に変換する。今回読み出す FE チップは 1 枚であるため、1 つの端子だけを用いる。

表 5.2 読み出しに使用した PC の性能。研究室で所有する PC を使用した。OS は centOS7 である。

CPU	Type	Core	Thread	Clock speed[GHz]	Memory [GB]	Disk [GB]
Intel(R) Core(TM) i7-8700K CPU		6	12	3.7	16.18	214

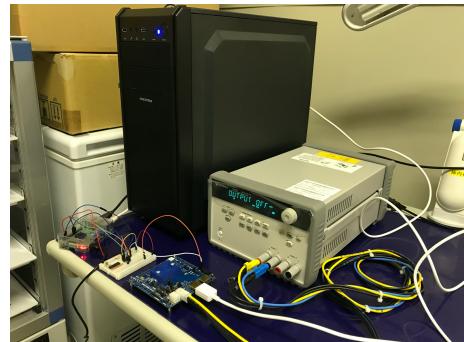
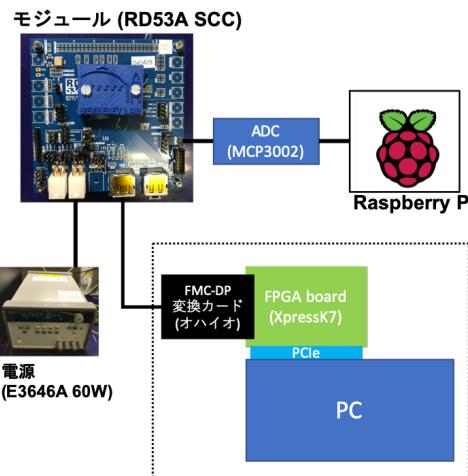


図 5.7 ハードウェアセットアップ。図はセットアップの概要 (左図) と実際に設置した様子 (右図) を示している。FE チップ読み出し装置、電源、サーミスタ読み出し装置をそれぞれ設置、配線し、読み出し操作、データ取得を行った。

5.2.2 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概要を図 5.7 に示す。

816 5.3 デモンストレーションの流れ

817 デモンストレーションで用いるため、中央データベースにおける登録 ID の定義方法 [42] に従い以下の
818 モジュール、FE チップを登録した。登録したモジュール、FE チップの ID を以下に示す。

819 モジュール ID 20UPGR00000001.

820 FE チップ ID 20UPGFC9999999.

821 今回のデモンストレーションではこれらを用いてデータベース操作の確認を行う。

822 確認した機能を行った流れの順に以下に示す。

- 823 1. 中央データベースからモジュール情報のダウンロード.
- 824 2. 読み出し試験実施、結果をローカルデータベースにアップロード.
- 825 3. DCS 情報の取得、監視.
- 826 4. 試験結果検索.
- 827 5. 試験結果閲覧.
- 828 6. 結果選択とピクセル解析.
- 829 7. 中央データベースへ試験結果のアップロード

Latest Result								
Module Name	Chip Name	Current Stage	Test Type	User	Site	Date	Link	Tag
20UPGR00000001	20UPGFC999999	MODULEWIREBONDING	PIXEL_FAILURE_TEST	hokuyama		2020/12/05 07:05:51	result page	Tag List

図 5.8 ダウンロードしたモジュール情報確認画面。図の表において、今回登録した 20UPGR00000001 の ID を持つモジュールがローカルデータベースのアプリケーション上で確認できていることが分かる。また対応する FE チップの ID も確認することができる。

830 5.4 機能確認

831 読み出し試験を通して、各ソフトウェア機能が正しく動くことを確認した。詳細を以下に記す。

832 5.4.1 中央データベースからモジュール情報のダウンロード

833 中央データベースから登録したモジュール情報をダウンロードし、ウェブアプリケーションで確認した。確認した画面を図 5.8 に示す。今回行った試験結果はこのモジュールに紐つける形でローカルデータベースに保存した。

836 5.4.2 読み出し試験実施

837 以下の流れに沿って読み出しを行ない、結果をローカルデータベースに保存した。

- 838 1. デジタル回路読み出し (`std_digitalscan`).
- 839 2. アナログ回路読み出し (`std_analogscan`).
- 840 3. 調整前 Threshold 測定.
- 841 4. Threshold 調整.
- 842 5. ToT 調整.
- 843 6. Threshold 再調整.
- 844 7. 調整後 Threshold 測定 (`std_thresholdscan`).
- 845 8. ToT 測定 (`std_totscan`).
- 846 9. ノイズ測定 (`std_noisescan`).

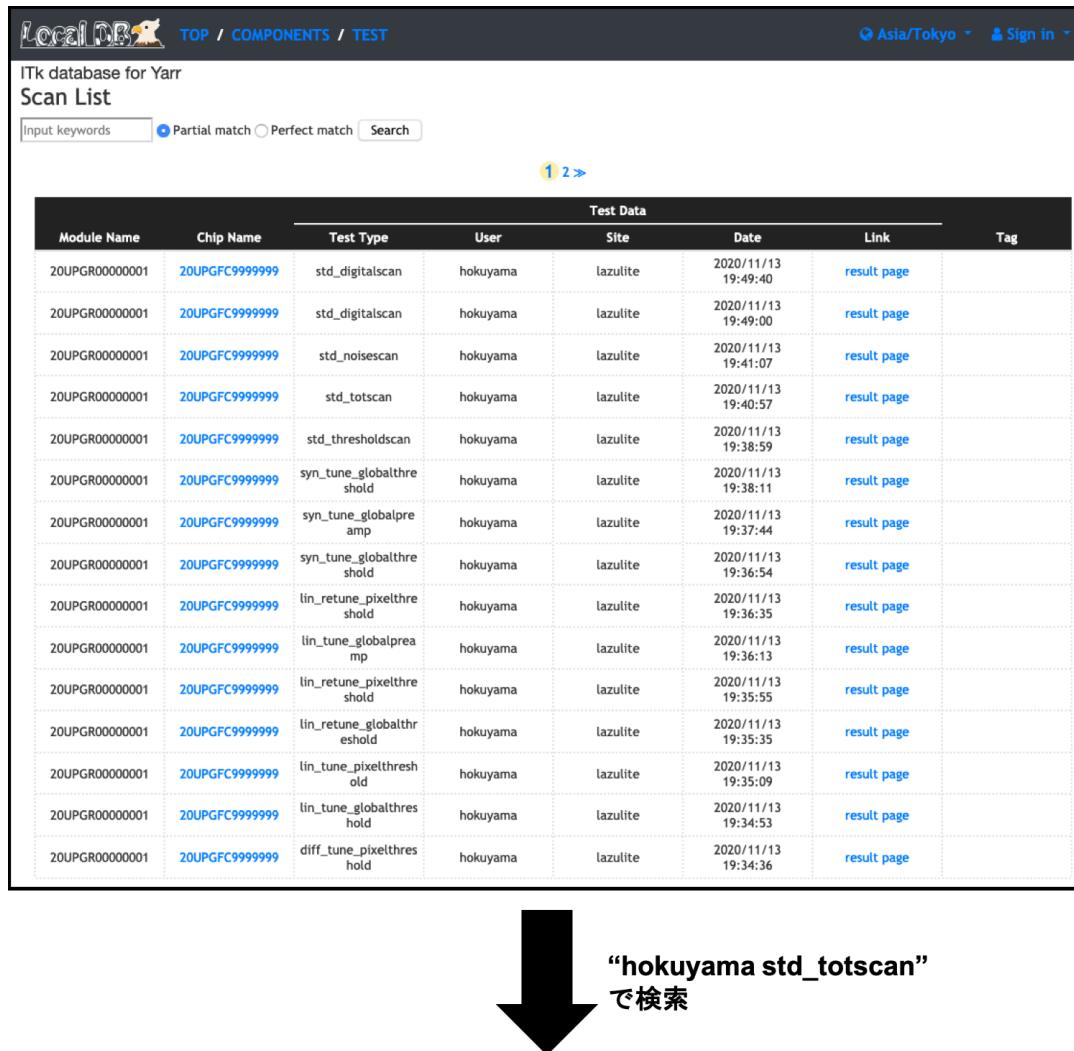
847 また読み出し試験を通して DCS 情報は InfluxDB を用いて監視し、試験結果と同様にローカルデータベースに保存した。



図 5.9 DCS 情報のモニタリングの様子。図において横軸は時間であり、縦軸は上から温度(青)、電流(緑)、電圧(黄)を示す。それぞれデータの監視が行えていることが分かる。温度に関して、電源オン、オフの付近で変化が大きいことが分かる。電流に関して、読み出し(チューニング)を行っている途中も変化が見られる。

5.4.3 DCS 情報の監視

読み出し試験は、DCS 情報を監視しながら行った。それぞれのデータは対応するソフトウェアを用いて InfluxDB にアップロードし、Grafana を使って監視をした。その様子を図 5.9 に示す。

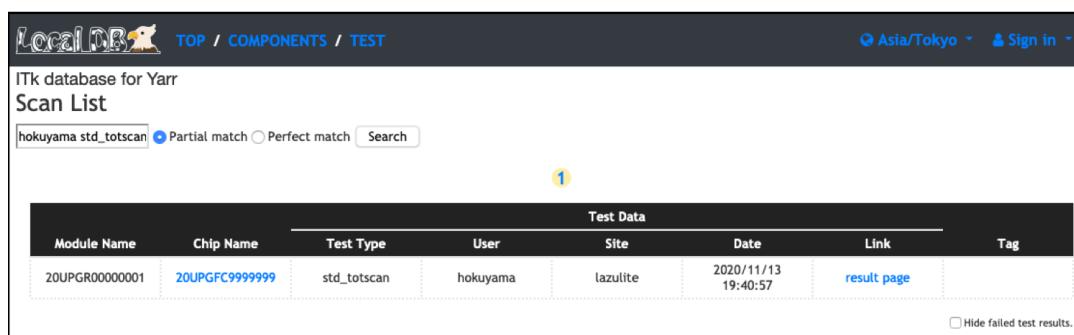


The screenshot shows the LocalDB Scan List page. At the top, there is a navigation bar with the LocalDB logo, 'TOP / COMPONENTS / TEST', and account information ('Asia/Tokyo', 'Sign in'). Below the navigation is a search bar with the placeholder 'Input keywords' and search options for 'Partial match' or 'Perfect match'. The main area is titled 'Scan List' and contains a table of 'Test Data'. The table has columns: Module Name, Chip Name, Test Type, User, Site, Date, Link, and Tag. There are 15 rows of data, each representing a different test entry. A large black arrow points downwards from the top table to the bottom table.

Test Data							
Module Name	Chip Name	Test Type	User	Site	Date	Link	Tag
20UPGR00000001	20UPGFC999999	std_digitalscan	hokuyama	lazulite	2020/11/13 19:49:40	result page	
20UPGR00000001	20UPGFC999999	std_digitalscan	hokuyama	lazulite	2020/11/13 19:49:00	result page	
20UPGR00000001	20UPGFC999999	std_noisescan	hokuyama	lazulite	2020/11/13 19:41:07	result page	
20UPGR00000001	20UPGFC999999	std_totscan	hokuyama	lazulite	2020/11/13 19:40:57	result page	
20UPGR00000001	20UPGFC999999	std_thresholdscan	hokuyama	lazulite	2020/11/13 19:38:59	result page	
20UPGR00000001	20UPGFC999999	syn_tune_globalthreshold	hokuyama	lazulite	2020/11/13 19:38:11	result page	
20UPGR00000001	20UPGFC999999	syn_tune_globalpreamp	hokuyama	lazulite	2020/11/13 19:37:44	result page	
20UPGR00000001	20UPGFC999999	syn_tune_globalthreshold	hokuyama	lazulite	2020/11/13 19:36:54	result page	
20UPGR00000001	20UPGFC999999	lin_retune_pixelthreshold	hokuyama	lazulite	2020/11/13 19:36:35	result page	
20UPGR00000001	20UPGFC999999	lin_tune_globalpreamp	hokuyama	lazulite	2020/11/13 19:36:13	result page	
20UPGR00000001	20UPGFC999999	lin_retune_pixelthreshold	hokuyama	lazulite	2020/11/13 19:35:55	result page	
20UPGR00000001	20UPGFC999999	lin_retune_globalthreshold	hokuyama	lazulite	2020/11/13 19:35:35	result page	
20UPGR00000001	20UPGFC999999	lin_tune_pixelthreshold	hokuyama	lazulite	2020/11/13 19:35:09	result page	
20UPGR00000001	20UPGFC999999	lin_tune_globalthreshold	hokuyama	lazulite	2020/11/13 19:34:53	result page	
20UPGR00000001	20UPGFC999999	diff_tune_pixelthreshold	hokuyama	lazulite	2020/11/13 19:34:36	result page	

↓

**“hokuyama std_totscan”
で検索**



The screenshot shows the LocalDB Scan List page after a search. The search bar now contains the query 'hokuyama std_totscan'. The search results table shows only one row of data, which corresponds to the 5th row in the first screenshot. This row has a 'Link' column value of '[result page](#)'.

Test Data							
Module Name	Chip Name	Test Type	User	Site	Date	Link	Tag
20UPGR00000001	20UPGFC999999	std_totscan	hokuyama	lazulite	2020/11/13 19:40:57	result page	

図 5.10 検索機能確認の様子。図は検索実行前の試験結果一覧（上図）と実行後（下図）を示す。図の例で“hokuyama std_totscan”的2つのキーワードで検索を行っており、実行後は対応する試験結果1つが表示されていることが分かる。この試験結果について試験実施者は“hokuyama”、試験項目は“std_totscan”であるため、検索機能が正常に動いていることが分かる。

5.4.4 検索機能

853 検索機能の確認を行い、正常に使用できることを確認した。検索機能実行の様子を図 5.10 に示す。

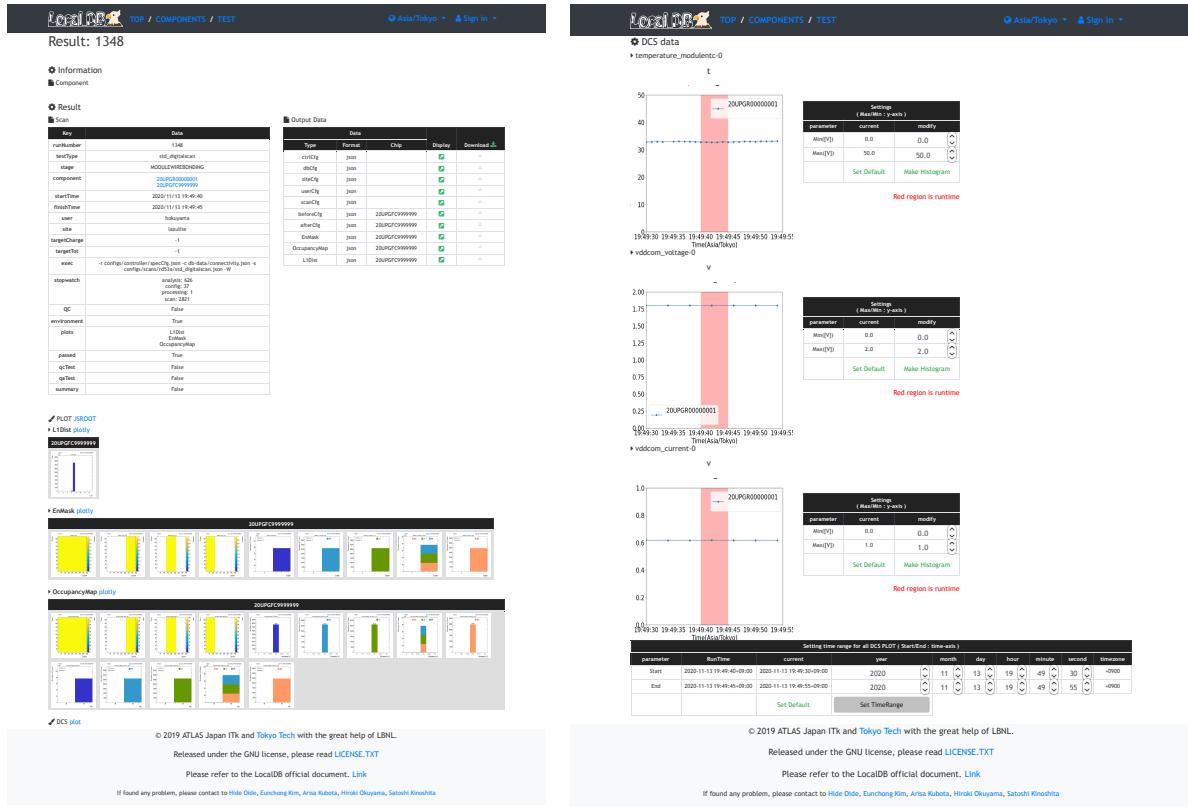


図 5.11 試験結果の閲覧。図は試験結果(左図)と試験におけるDCS情報のグラフ(右図)を示しており、上から温度、電圧、電流となっている。図は std_digitalscan の結果であり、試験情報及び結果のグラフが確認できる。また右図よりDCS情報も正常にローカルデータベース上に保存され、表示されていることが分かる。

854 5.4.5 試験結果閲覧

855 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 5.11 に示す。

856 5.4.6 結果選択とピクセル解析

857 読み出し試験結果を選択し、ピクセル解析を行なった。結果選択画面を図 5.12 に示す。このデモンス
858 トレーションにおける不良評価基準は 3 章で述べた表 3.5 の中から、現時点でシステムに実装している以
859 下の項目を抜粋した。

- 860 1. Digital Dead.
- 861 2. Digital Bad.
- 862 3. Analog Dead.
- 863 4. Analog Bad.
- 864 5. Tuning Failed.
- 865 6. Tuning Bad for Threshold.
- 866 7. Tuning Bad for ToT.
- 867 8. High ENC.
- 868 9. Noisy.

869 解析結果を図 5.13、それぞれの評価基準における不良ピクセルの分布を図 5.14 に示す。Synchronous
870 フロントエンドに不良ピクセルが多く分布していることが分かった。

871 5.4.7 中央データベースへ試験結果のアップロード

872 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、情報が正しくアップロードされていること
873 を確認した。図 5.15 に中央データベースのウェブページを示す。結果ファイルや解析結果が正しくアッ
874 プロードされていることが分かる。

875 5.5 本章のまとめ

876 本章では品質試験のデモンストレーションとして、読み出し試験の 5 項目を行なった。機能の確認を目
877 的とし、本研究で確立した品質試験におけるデータベース操作の流れに沿って確認を行なった。

878 本デモンストレーションにおいて達成した項目を以下に示す。

- 879 1. モジュールを実際に中央データベースに登録し、その情報をローカルデータベース上へダウンロー
880 ドできることを確認した。
- 881 2. 読み出しシステムをセットアップし、試験結果のアップロード及びモジュール情報への紐付けがで
882 きることを確認した。
- 883 3. DCS 情報取得システムを製作し、読み出し試験を通して監視した。
- 884 4. 試験結果検索、閲覧機能を確認した。検索機能については 2 キーワードでの動作が正常に動くこと
885 を確認した。
- 886 5. ピクセル解析ツールが正常に動くことを確認した。本デモンストレーションにおいては syn-
887 chronous フロントエンド上に不良ピクセルが多く見られた。
- 888 6. 読み出し試験結果を中央データベースにアップロードし、中央データベースのウェブページ上で結
889 果を確認できた。

Selection Page
→Select scan results for QC result in this stage.

Component: 20UPGR00000001
 Current Stage: MODULEWIREBONDING
 Please select one result for each test items.

►std_digitalscan

Test Data										
run Number	Module Name	Chip Name	Stage	Test Type	User	Site	Date	Link	Tag	Select
1392	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_digitalscan	hokuyama	tokyo_institute_of_technology	2020/12/08 22:35:47	result page		<input type="radio"/>
1391	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_digitalscan	hokuyama	tokyo_institute_of_technology	2020/12/08 22:29:39	result page		<input type="radio"/>
1351	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_digitalscan	hokuyama	tokyo_institute_of_technology	2020/12/03 14:54:43	result page		<input type="radio"/>
1350	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_digitalscan	hokuyama	tokyo_institute_of_technology	2020/11/30 19:57:19	result page		<input type="radio"/>
1348	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_digitalscan	hokuyama	lazulite	2020/11/13 19:49:40	result page		<input type="radio"/>
1347	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_digitalscan	hokuyama	lazulite	2020/11/13 19:49:00	result page		<input type="radio"/>
1328	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_digitalscan	hokuyama	lazulite	2020/11/13 19:31:02	result page		<input checked="" type="radio"/>
1327	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_digitalscan	hokuyama	lazulite	2020/11/13 19:29:24	result page		<input type="radio"/>
1326	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_digitalscan	hokuyama	lazulite	2020/11/13 19:28:38	result page		<input type="radio"/>

►std_analogscan

Test Data										
run Number	Module Name	Chip Name	Stage	Test Type	User	Site	Date	Link	Tag	Select
1329	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_analogscan	hokuyama	lazulite	2020/11/13 19:31:13	result page		<input checked="" type="radio"/>

►std_thresholdscan

Test Data										
run Number	Module Name	Chip Name	Stage	Test Type	User	Site	Date	Link	Tag	Select
1344	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_thresholdscan	hokuyama	lazulite	2020/11/13 19:38:59	result page		<input checked="" type="radio"/>
1330	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_thresholdscan	hokuyama	lazulite	2020/11/13 19:31:24	result page		<input type="radio"/>

►std_totscan

Test Data										
run Number	Module Name	Chip Name	Stage	Test Type	User	Site	Date	Link	Tag	Select
1345	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_totscan	hokuyama	lazulite	2020/11/13 19:40:57	result page		<input checked="" type="radio"/>

►std_noisescan

Test Data										
run Number	Module Name	Chip Name	Stage	Test Type	User	Site	Date	Link	Tag	Select
1346	20UPGR00000001	20UPGFC9999999	MODULEWIRE BONDING	std_noisescan	hokuyama	lazulite	2020/11/13 19:41:07	result page		<input checked="" type="radio"/>

Proceed

[Back to the component page](#)

© 2019 ATLAS Japan ITk and Tokyo Tech with the great help of LBNL.
 Released under the GNU license, please read [LICENSE.TXT](#)
 Please refer to the LocalDB official document. [Link](#)
 If found any problem, please contact to [Hide Olde](#), [Eunchong Kim](#), [Arisa Kubota](#), [Hiroki Okuyama](#), [Satoshi Kinoshita](#)

図 5.12 読み出し試験結果の選択ページ。読み出し試験実施後、ピクセル解析処理と中央データベースとの同期を実行するために試験結果を選択する必要がある。図では std_digitalscan、std_analogscan、std_thresholdscan、std_totscan、std_noisescan の 5 項目を選択している。

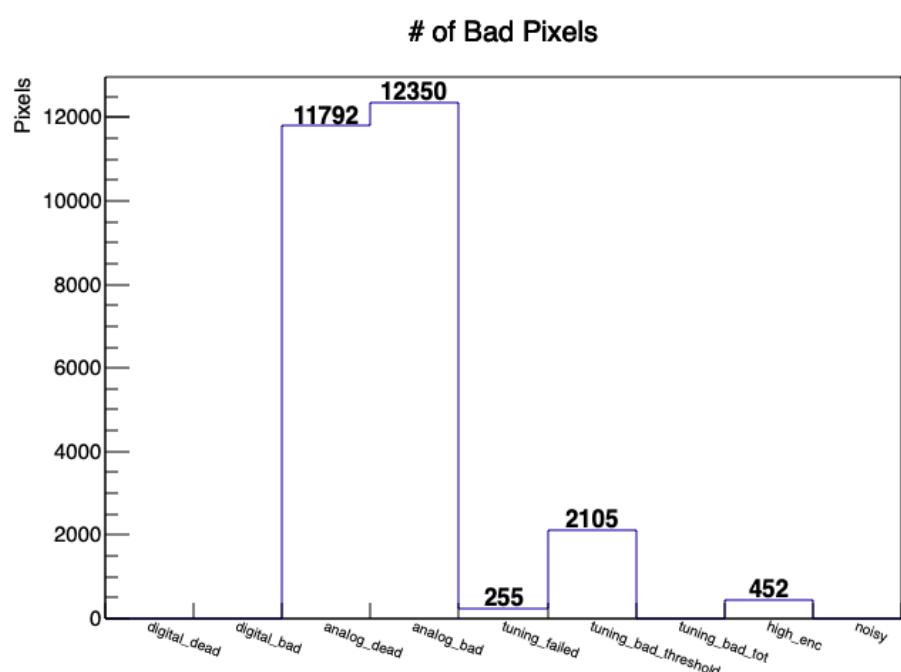


図 5.13 ピクセル解析結果。図において横軸は評価基準、縦軸は該当するピクセル数を表す。
analog_dead、analog_bad の割合が多いいため、アナログ回路読み出しに失敗しているピクセルが多いことが読み取れる。

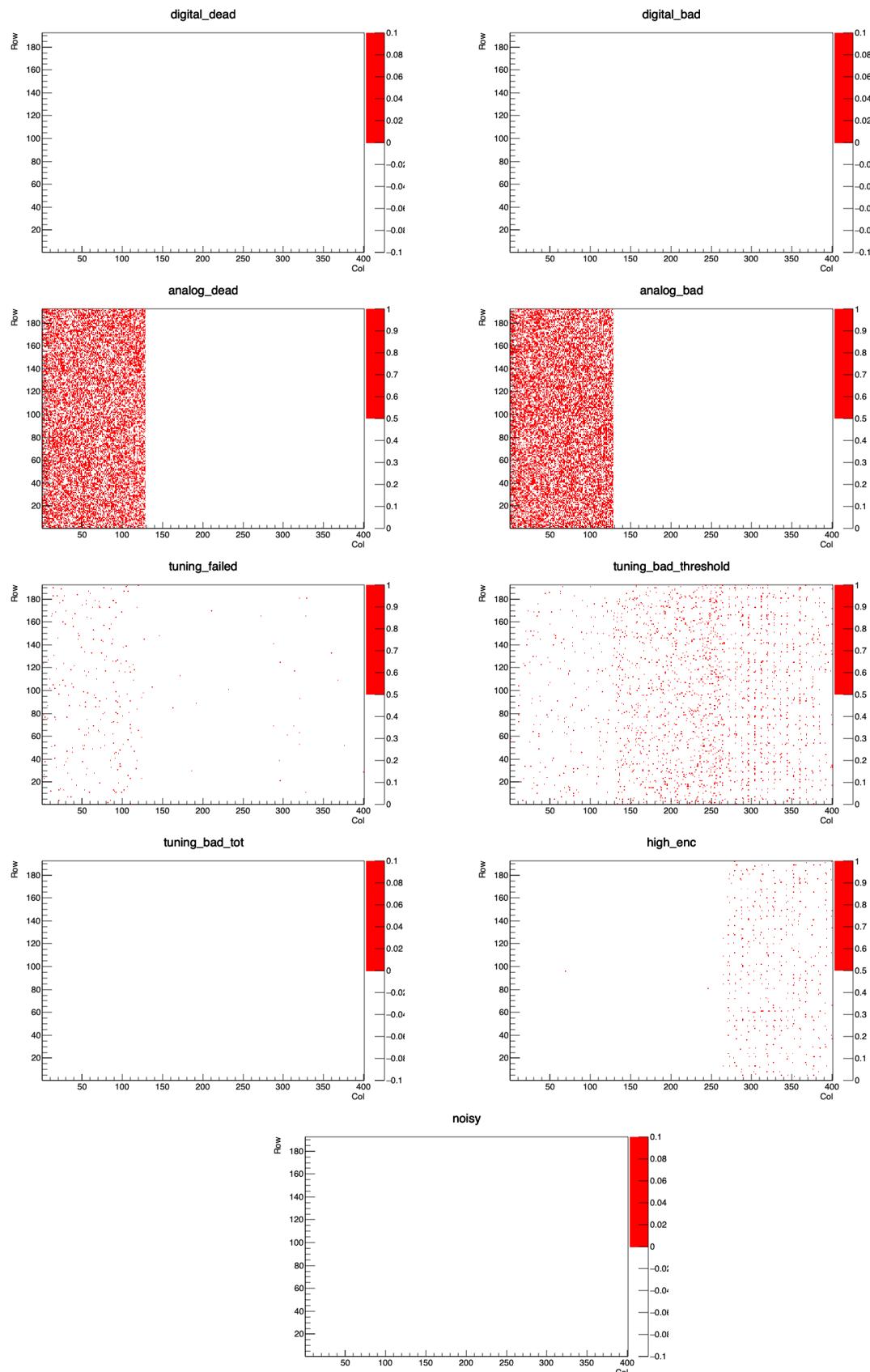


図 5.14 各評価基準における不良ピクセルの分布。各図は二次元ヒストグラムであり、横軸が FE チップにおける各ピクセルの列番号、縦軸が行番号を示しており、赤い領域が不良ピクセルを表している。図 5.13 においてアナログ回路読み出しに失敗しているピクセルが多いことがわかったが、その不良ピクセルは Synchronous フロントエンド上に存在していることが分かる。また Threshold 調整に失敗しているピクセル (tuning_bad_threshold) は全体的に分布していることが分かる。

The screenshot shows the 'Test Run Details' page for a 'ModuleQC::PixelFailureTest'. The top navigation bar includes 'ATLAS ITk Production Database' and 'Test' tabs, and a user profile for 'Hiroki Okuyama'. The main content area displays basic properties like INSTITUTE (Tokyo Institute of Technology), TEST DATE (13.11.2020 19:31), and TEST TYPE (ModuleQC::PixelFailureTest). Below these are sections for Associated Components, Properties (QC stage: MODULEWIREBONDING), and Results. The Results section lists various failure types with their counts: Noisy (0), High ENC (452), Tuning Bad ToT (0), Tuning Bad Threshold (2105), Tuning Failed (255), Analog Bad (12350), Analog Dead (11792), Digital Bad (0), and Digital Dead (0). The Attachments section shows several zip files: std_digitalscan_datafiles.zip, std_digitalscan_configfiles.zip, std_analogscan_datafiles.zip, std_analogscan_configfiles.zip, std_thresholdscan_datafiles.zip, std_thresholdscan_configfiles.zip, std_totscan_datafiles.zip, std_totscan_configfiles.zip, std_noisescan_datafiles.zip, and std_noisescan_configfiles.zip. A Comments section at the bottom has a plus sign icon.

図 5.15 中央データベースにおける読み出し試験結果ページ。読み出し試験結果を中央データベースにアップロードした。図は中央データベースのウェブページにおける、アップロードされた試験結果ページである。図の中央部に各評価基準における不良ピクセルの数、下部に各試験項目ごとの結果ファイルをまとめた Zip ファイルが表示されていることが分かる。試験結果のアップロードが正常に完了し、中央データベース上で結果を確認できることが分かる。

890 第6章

891 ローカルデータベースにおける検索機能

892 モジュール組み立て工程において、読み出し試験は複数回行われ、一回の試験で行う項目も複数存在す
 893 る。そのため、生産時には試験結果が数多くデータベースにアップロードされることになる。節4.3.6で
 894 説べたように、必要な試験結果を素早く取得することを目的として検索機能を実装した。詳細を以下に
 895 示す。

896 6.1 実装方法

897 一般的にウェブで用いられているフリーワードの検索エンジンのような機能を実装しようと考えた。
 898 ユーザの操作を最小限にし、直感的かつ柔軟な検索ができるようにするためである。

899 読み出し試験において、対象とする検索キーワードを以下の項目に絞った。読み出し試験に関するデータベース構造及び保存する情報は決まっている。そのため、ユーザが対象とする検索キーワードは限定
 900 される。ここでは以下の項目にしぼり、これらを検索キーワードとしてサポートすれば十分であると考えた。

- 903 • モジュール及びFEチップのID.
- 904 • 読み出し試験項目名(例:std_digitalscan).
- 905 • 試験者.
- 906 • 試験場所.
- 907 • 試験日時.
- 908 • タグ機能を用いてつけられたタグ.

909 そこで実装方法として、以下の2つを考えた。

- 910 1. 各試験に関する情報をプログラム上で配列(Pythonリスト)に保持し、検索キーワードが含まれる
 911 かを確認する方法.
 - 912 2. 各試験に関する情報を持つドキュメント、コレクションを予め作成、それを参照し検索を行う方法.
- 913 これらについて以下で詳細を説明する。

914 6.1.1 方法1: Pythonリストを用いた一致確認

915 Pythonリストを使う実装の場合、以下のような流れで検索処理を行う。

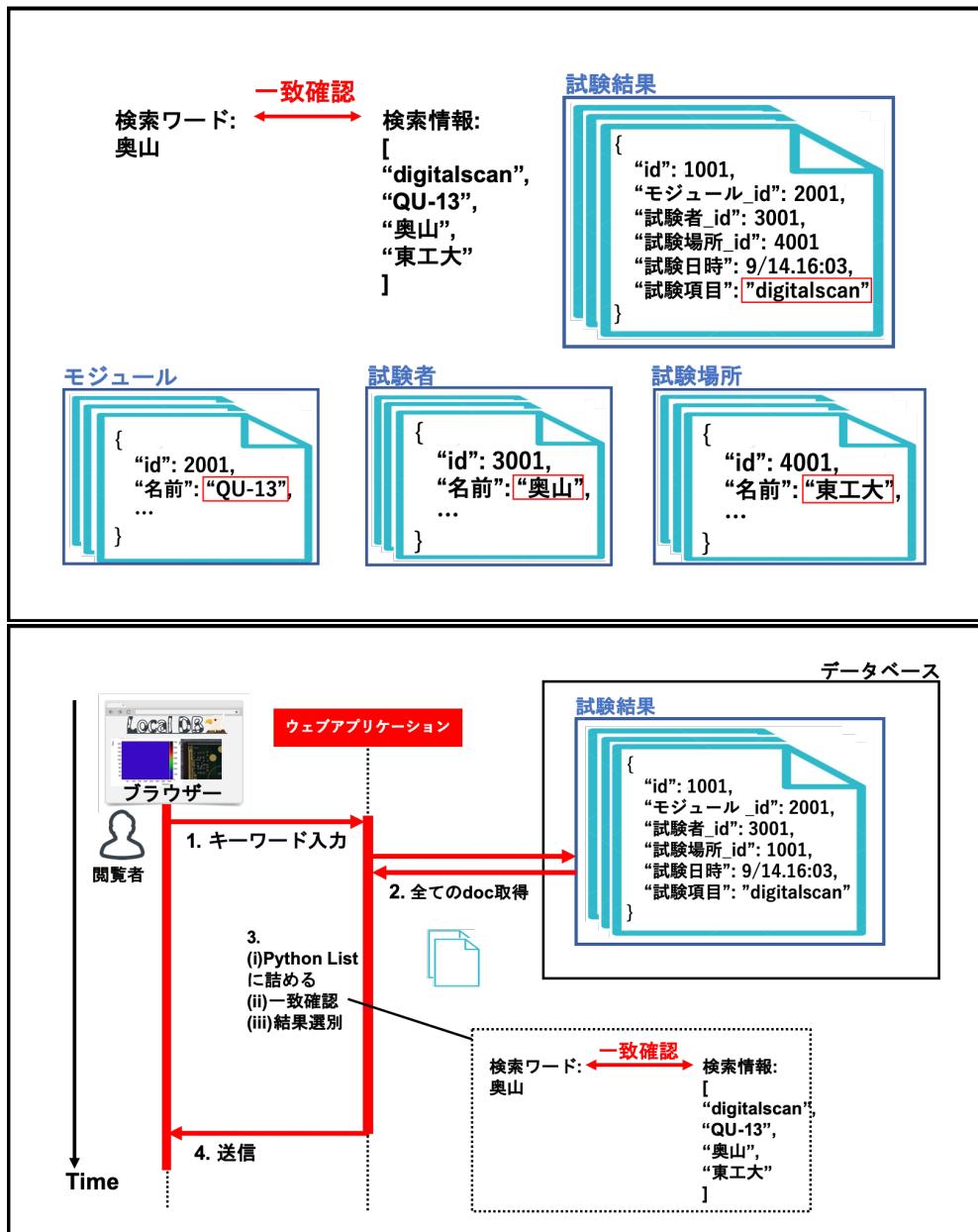


図 6.1 方法 1:Python リストを用いた場合の検索処理のイメージ図。上図は各コレクションと保存されている情報の例を示しており、下図は実際に検索を行った時の処理の流れを表している。上図中の赤枠で囲われた情報のように検索対象となる名前情報は複数のコレクションにまたがって保存されている。各試験結果に対してこれらの情報を集めリスト内に保持し、各要素とキーワードとの一致を確認することで検索処理を行う。

- 916 1. ユーザが検索キーワードを入力し、処理を実行。
 917 2. アップロードされた全ての読み出し試験結果に関する情報を取得。
 918 3. 各試験結果に関する情報を Python リストに保持、検索キーワードとの一致を確認、試験を選別。
 919 4. ブラウザに送信。

920 アルゴリズムのイメージを図 6.1 に示す。この方法では、データベース内の試験結果とアプリケーション内の関数だけで全ての処理を行うことが可能なため、シンプルな実装方法であると言え、始めに思いつく方法であった。

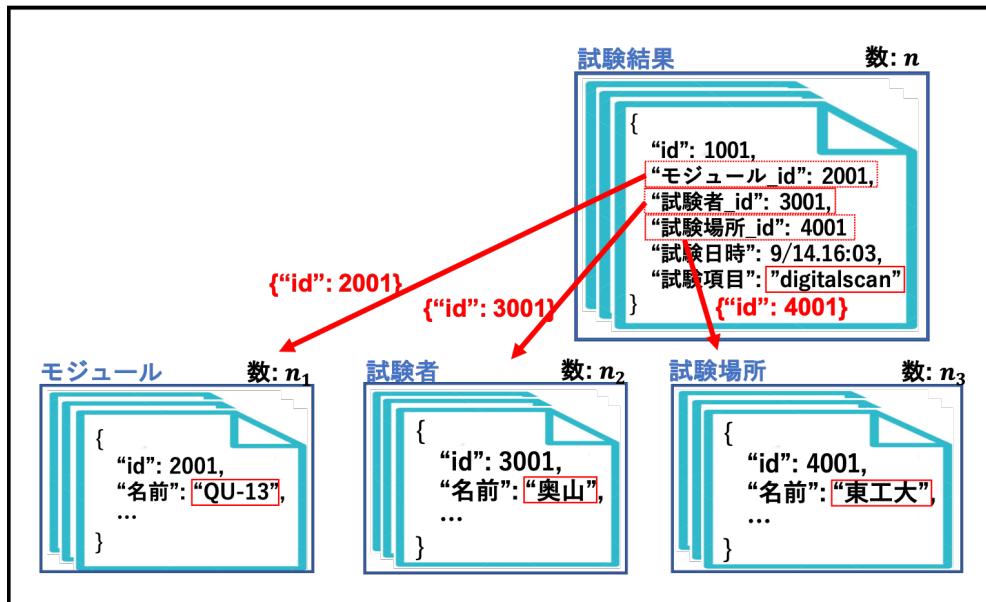


図 6.2 方法 1 における問題点のイメージ図。図 6.1 でも述べたように、検索対象となる情報は複数のコレクションにまたがって保存される。そのため、コレクションを超えて検索を行うような場合は試験結果の数以上に、処理に時間がかかるてしまう。この図の例の場合、あるコレクション検索にかかる時間がドキュメント数に対して線形とすると、 $O(n * (n_1 + n_2 + n_3))$ の処理時間がかかると考えられる。

しかし、この方法を試験実装したところ、ドキュメント数の増加に伴い検索処理時間を大きく要してしまう問題が発生した。図 6.2 のようにデータベース内の構造は複数のコレクションを跨いで情報を保持しているため、試験結果全てに対してリアルタイムでこの処理を行うと、検索時間が大きくかかるてしまう。このシステムのデータ構造においては、表 4.1 において各試験結果の情報を保存する testRun、component と testRun の関係を保存する componentTestRun の構造による遅延であると考えられる。試験結果数を n とすると、testRun が n 、componentTestRun がそれぞれ $O(n)$ のドキュメント数を持つことになる。これらのドキュメントを全て検索し、一致確認を行うと全体で $O(n^2)$ の時間がかかる。イメージを図 6.3 に示す。この実装方法について、ドキュメント数と処理時間の関係を測定したところ、図 6.4 のようになり、確かにドキュメント数に対して 2 次的に増加していた。測定の詳細については後述する。

6.1.2 方法 2: 検索情報を持つドキュメントを作成、使用

検索機能を改善するため方法 2 を考案し、実装を行った。ここで読み出し試験のデータ構造及び使用している Python フレームワークの変更はせずに処理時間を改善することを試みた。

改善策として、各試験に対応する検索キーワードを別のドキュメントに予め保持しておき、処理実行時にはそれを参照することで検索を行う方法を考案し、試験を行った。アルゴリズムのイメージを図 6.5 に示す。検索情報コレクションに入るドキュメント数は、試験結果数と同じである。そのため試験結果数に対する処理時間は $O(n)$ と考えられ、方法 1 に比べて検索コストを削減できると考えた。

なお、このコレクションはウェブアプリケーションの立ち上げ時に生成するシステムとした。コレクション生成処理の計算コストの見積もりとしてアプリケーション立ち上げにかかる時間を測定した。これに関しては後述する(図 6.7)。

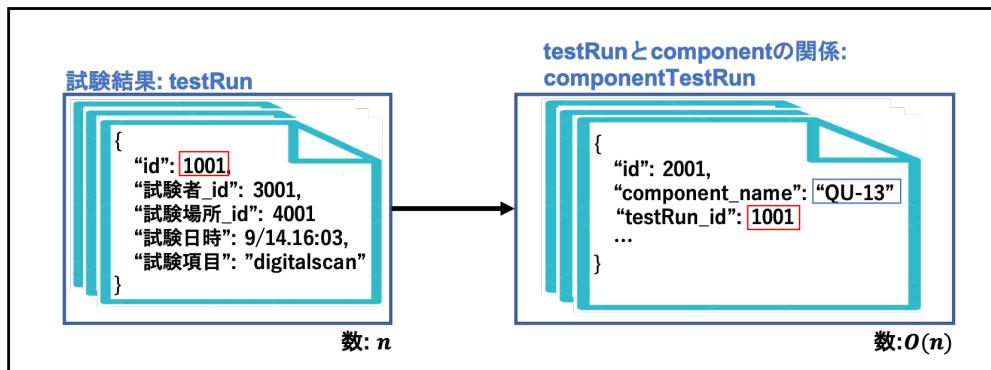


図 6.3 処理時間のボトルネックとなっているデータ構造の図。読み出し試験におけるデータ構造(図 4.4)の1つに、各試験結果情報を保存する testRun、component と testRun の関係性を保持する componentTestRun という構造が存在する。試験結果数を n とすると、testRun のドキュメント数は n 、componentTestRun の数は試験数と component の数の積となるため $O(n^2)$ となる。結果的に検索処理に $O(n^2)$ の時間がかかるてしまう。

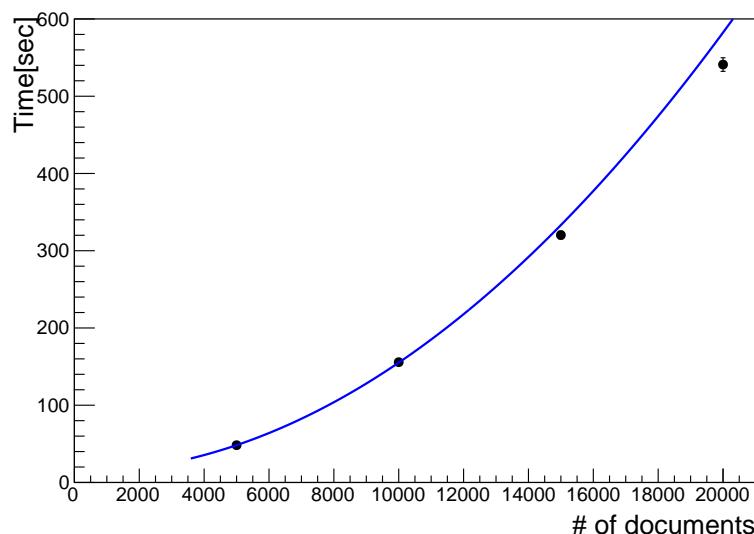


図 6.4 方法 1 における検索処理速度測定結果。横軸が試験結果のドキュメント数、縦軸が処理時間を表している。図 6.3 で述べたように、方法 1 の検索処理では試験結果数 n に対して $O(n^2)$ の検索時間がかかるため、試験結果のドキュメント数に対して処理時間は二次的に増加している。フィット関数や生産時における処理時間の見積もりに関しては後述する。

943

検索情報ドキュメントの例をコード 6.1 に示す。

ソースコード 6.1 検索情報コレクションに入るドキュメントの例。このように試験結果の ID("runId")、試験日時 ("timeStamp")、検索対象となる名前情報 ("data") が保存される。

```

944
945 1 {
946 2   "_id" : ObjectId("5fd489f60e2ca70557e44a8b"),
947 3   "runId" : "5fc4d027b1ef7c6297c91040",
948 4   "timeStamp" : ISODate("2020-11-30T10:57:19Z"),
949 5   "data" : [
950 6     "20UPGR00000001",
951 7     "20UPGFC9999999",
952 8     "std_digitalscan",
953 9     "hokuyama",
954 10    "tokyo_institute_of_technology",
955 11    "2020/12/09"

```

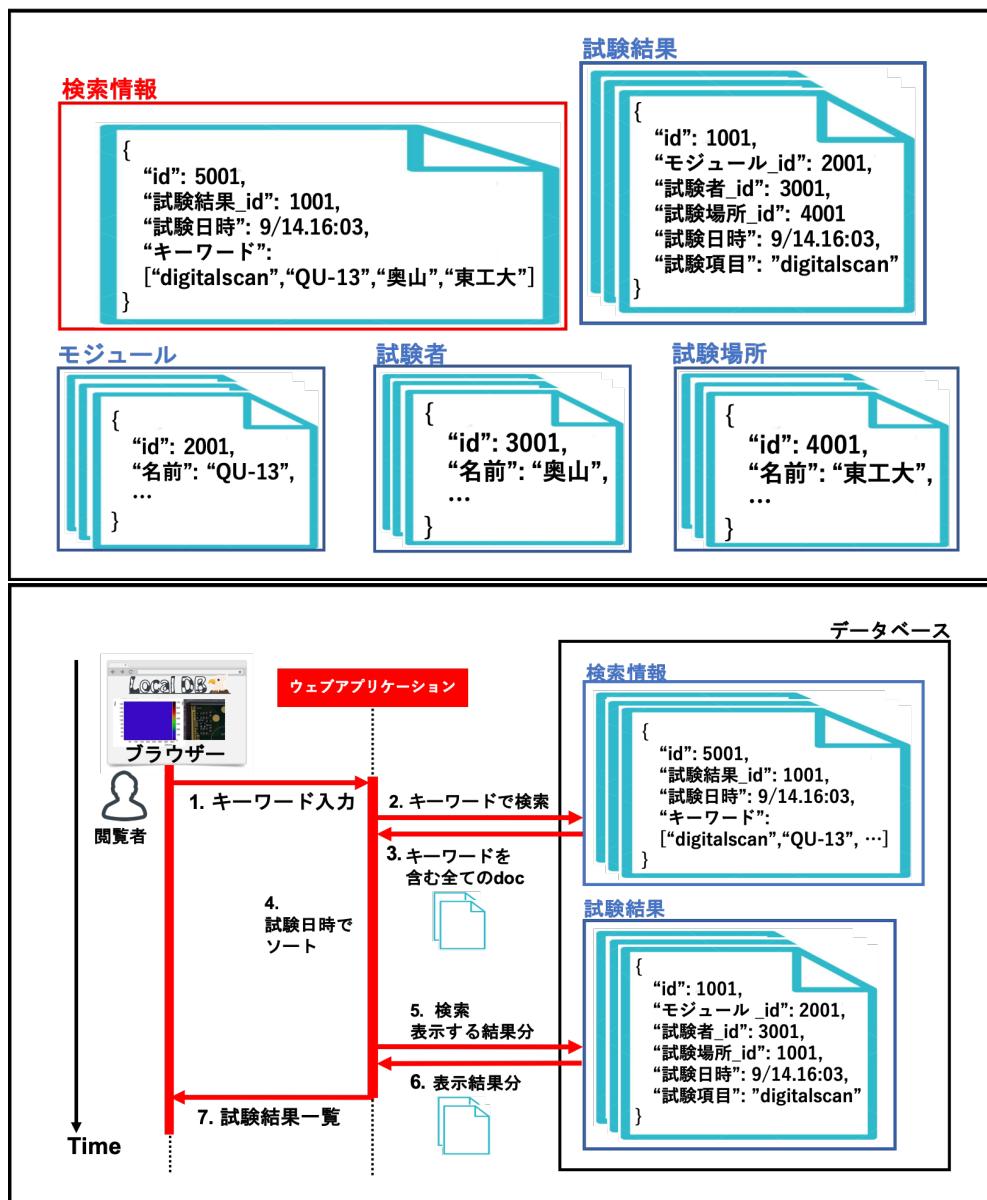


図 6.5 方法 2:検索キーワード専用コレクションを用いた場合のイメージ図。上図は各コレクションと保存されている情報の例を示しており、下図は実際に検索を行った時の処理の流れを表している。方法 2 では上図の赤枠で囲っている検索情報のコレクションを新たに設け、これを参照することで検索処理を行う。このコレクション内には各試験結果に対応したドキュメントが 1 つ保存され、全検索情報と試験結果の ID を持つものとなっている。処理の流れとしては下図のように、検索情報コレクションよりキーワードに対応する試験結果を抽出する処理と、表示の際に必要な情報を試験結果コレクションから取得する処理の 2 つに分かれた構造となっている。このような処理をすることにより、各検索処理にかかる時間は試験結果数に対して $O(n)$ になると考えられる。

956 12]
957 13 }

表 6.1 測定に使用したノート PC(MacBookAir(13-inch,2017)) の性能。検索処理時間の測定に個人的に使用しているノート PC を使用した。

種類	CPU	Type	Core	Thread	Clock speed[GHz]	Memory [GB]	Disk [GB]
MacBookAir(13-inch,2017)	Intel Core i5	2	4	1.8		8	256

表 6.2 検索機能処理時間測定の詳細。測定を行った試験結果数、回数、キーワード、検索モード、検索情報の詳細を示している。

試験結果数	5000, 10000, 15000, 20000
測定回数	各測定点に対して 20 回
検索キーワード	okuyama
検索モード	部分一致
各試験結果が持つ検索情報	全試験結果に対して同じ、以下に詳細
検索情報一覧	モジュール名: 20UPGR10000005 FE チップ名: 20UPGTU9000000 試験項目: std_digitalscan 試験者: okuyama 試験場所: default_host 試験日時: 2020/12/06

6.2 処理時間測定

考案した方法を実装し、検索処理時間の測定を行った。詳細を以下に示す。また方法 1 で行った処理時間測定も同様の条件で行った。

使用した装置

測定には個人的に使用しているノート PC(MacBookAir(13-inch,2017)) を用いた。性能を表 6.1 に示す。

測定内容

PC 上で MongoDB、ウェブアプリケーションを起動し、試験結果を MongoDB にアップロードした。コマンドプロンプトから以下のコマンドを実行し、ウェブアプリケーションのある試験結果ページのリクエストに対するアプリケーションのレスポンス時間を測定した。ここでは検索キーワードは “okuyama” とし、検索モードは部分一致としている。実際にアプリケーションを使用する際には、ブラウザに一覧表示をする時間が今回の測定時間に加算されることになる。

```
1 curl "http://127.0.0.1:5000/localdb/scan?keywords=okuyama&match=partial" -o /dev/null -w "%{time\_total}\n" 2> /dev/null -s
```

その他測定に関する詳細を表 6.2 に示す。

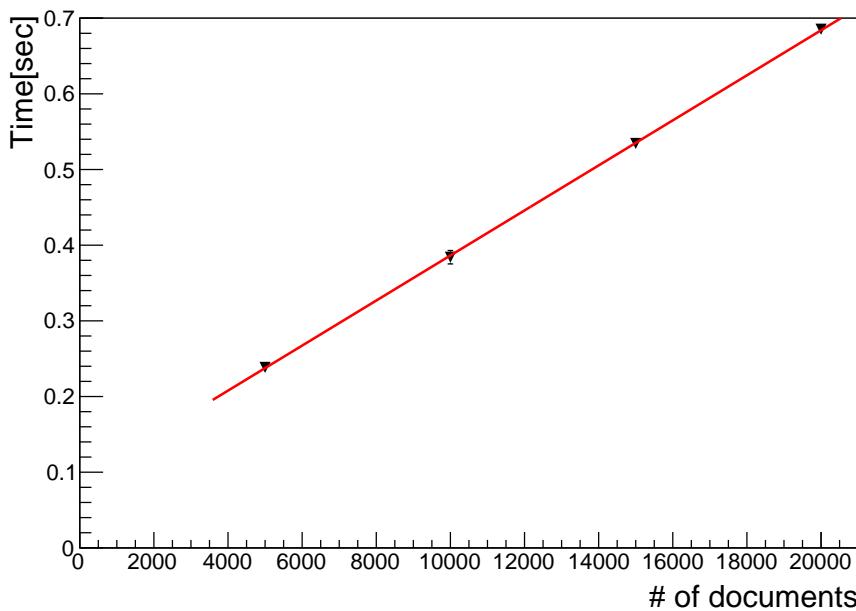


図 6.6 方法 2 における検索処理時間測定結果。横軸が試験結果のドキュメント数、縦軸が処理時間を表している。線形性を示していることが確認でき、検索実行時における処理時間は方法 1 に比べて優れている。

976 測定結果

977 方法 2 を用いて得られた結果を図 6.6 に示す。方法 1 の結果に関しては、既に図 6.4 で示した。比較す
978 ると方法 2 のほうが圧倒的に速く図での比較が難しいため、方法 2 と方法 1 の結果を同時に示すことは
979 していない。方法 1、方法 2 で得られた近似関数を式 6.1、6.2 に示す。方法 1 に対して、方法 2 はアルゴ
980 リズムの改善が見られる。

$$y = a_1 x^2 + b_1 \quad (6.1)$$

$$a_1 = (1.4 \pm 0.0) \times 10^{-6}$$

$$b_1 = 13 \pm 0$$

$$y = a_2 x + b_2 \quad (6.2)$$

$$a_2 = (3.0 \pm 0.1) \times 10^{-5}$$

$$b_2 = (8.9 \pm 0.8) \times 10^{-2}$$

981 現在は方法 2 で検索機能を実装し、サービスの 1 つとして提供している。

982 方法 2 では、アプリケーションの起動時に検索情報コレクションを生成するアルゴリズムとしている。
983 ここで検索情報コレクションの生成にかかる処理時間として、ドキュメント数に対するアプリケーション
984 の起動時間を測定した。結果を図 6.7 に示す。基本的には方法 1 と同じ操作を行うことで検索情報を収集
985 し、ドキュメントを作成する。そのため、処理時間は $O(n^2)$ となる。この処理を前もって行なうことで、
986 検索実行時の処理性能を上げている。また再起動の際には全てのドキュメントの削除、生成をする処理と

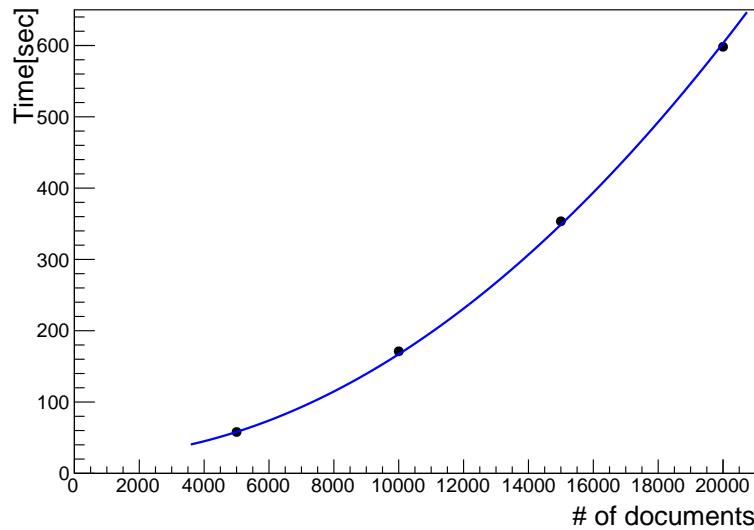


図 6.7 方法 2 における検索情報コレクションの生成時間測定結果。検索情報コレクションの生成にかかる処理時間として、ドキュメント数に対するアプリケーションの起動時間を測定した。横軸が試験結果のドキュメント数、縦軸が時間を示している。生成には、方法 1 と同様の処理を行なっているため $O(n^2)$ となっているのが分かる。ドキュメント数が 20,000 件の場合にはアプリケーションの起動に 10 分程度かかることが分かる。

987 なっている。これはスクリプト内でのドキュメント管理を簡略化するためである。実際の運用時に、起動
988 が遅い等のフィードバックを受ける場合にはこのドキュメント管理の処理も実装する必要がある。

989 生産時における検索時間の見積もり

990 各方法について、生産時における処理時間の見積もりを行う。簡単のため今回使用したデバイスと生産
991 時に使うサーバーの性能差は無視する。ここでデータベースで管理するモジュール数は日本が最多とし、
992 その数を予定している 2,000 とする。保存する読み出し試験数は 3 章で述べた項目の合計とし、1 つのモ
993 ジュールあたり 42 とする。全ての生産が終了した際の検索処理時間を見積もる。検索処理実行時間は、
994 上で得られた関係式を用いて方法 1、2 に対して式 6.3、6.4 のように見積もることができる。

$$\{(1.4 \pm 0.0) \times 10^{-6}\} \times (2000 \times 42)^2 + (13 \pm 0) = (9.8 \pm 0) \times 10^3 [\text{sec}] \quad (6.3)$$

995

$$\{(3.0 \pm 0.1) \times 10^{-5}\} \times (2000 \times 42) + \{(8.9 \pm 0.8) \times 10^{-2}\} = 2.6 \pm 0.1 [\text{sec}] \quad (6.4)$$

996 方法 1 では 1 回の検索に対して約 2.7 時間と見積もられ、生産時には検索機能として有用でないことが
997 わかる。方法 2 では生産終了時点においても数秒で処理を終えることができるため、生産を通して使用で
998 きると考えられる。

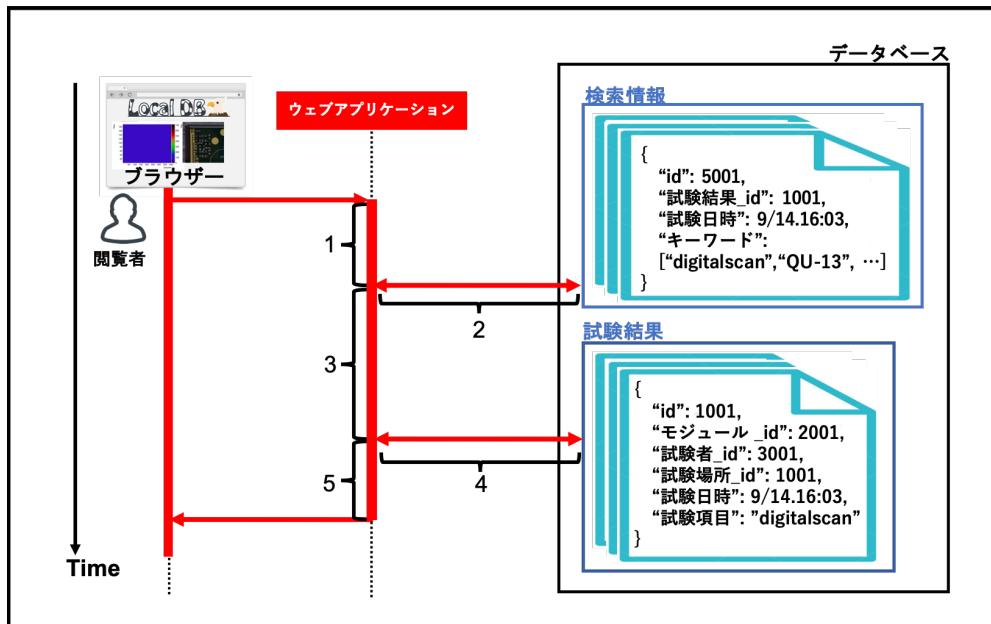


図 6.8 方法 2 の検索における詳細処理。図のように全体の流れにおけるアプリケーション内部での各処理、データベースから情報取得の各処理に 1 から 5 の番号をつけそれぞれにかかる処理時間を測定した。

999 6.3 改善方法と処理時間測定

1000 より処理時間を短くすることを目的として、新たな検索処理アルゴリズムの考案と測定を行った。詳細
1001 について以下に示す。

1002 6.3.1 方法 2 における検索処理時間の詳細調査

1003 先述したように、方法 2 では処理時間が改善した。この方法 2 について、処理時間の詳細を知るために
1004 追加で測定を行った。アプリケーション層での各処理について、以下のように番号をつける。

- 1005 1. キーワードを受け取り、検索情報コレクションに検索をかけるまでの処理。
- 1006 2. 検索情報コレクションに検索をかけ、情報を受け取る処理。
- 1007 3. ドキュメントを受け取り該当する試験結果 ID をまとめ、試験結果に対して検索をかけるまでの
1008 処理。
- 1009 4. 試験結果コレクションに検索をかけ、情報を受け取る処理。
- 1010 5. ドキュメントを受け取りデータを整形、ブラウザにレスポンスを返すまでの処理。

1011 イメージを図 6.8 に示す。

1012 ポトルネックとなっている処理を測定するために、上述した各処理にかかる時間の測定を行った。測定
1013 は試験結果数が 10,000 の場合に行った。

1014 結果を図 6.9 に示す。

1015 図より処理 3、5 の割合が大きいことがわかる。これらの処理について、特に以下の処理の割合が大き

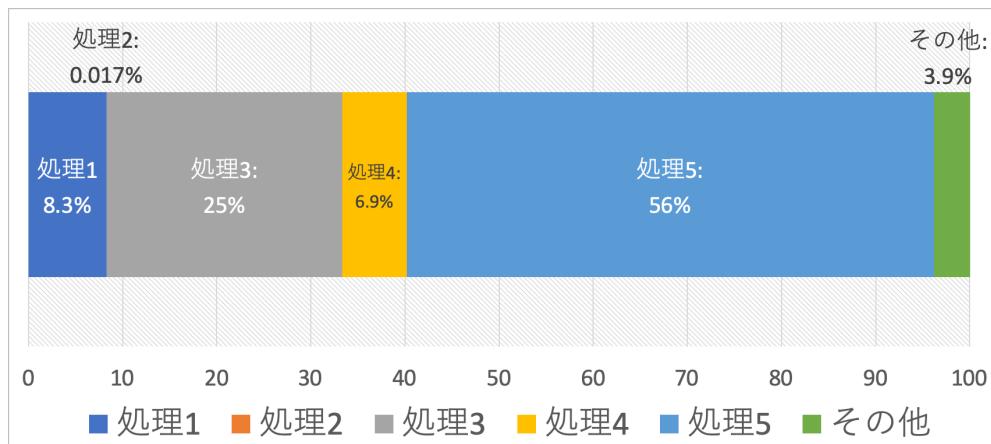


図 6.9 方法 2 における詳細処理時間の測定結果。図 6.8 における 5 つの詳細処理にかかる時間の割合を表している。処理 3、5 にあたるコレクション検索実行後のアプリケーション内での処理に多く時間がかかっていることが分かる。

表 6.3 処理 3,5 における型変換処理 6.5 の割合。処理 3、5 について、表の割合より、型変換処理 6.5 が支配的であることが分かる。

処理	全体 [sec]	処理 6.5[sec]	割合 [%]
3	0.091 ± 0.011	0.089 ± 0.005	97 ± 0
5	0.21 ± 0.00	0.18 ± 0.00	86 ± 1

1016 いことがわかった。

取得した複数ドキュメントを Python リストへ型変換する処理. (6.5)

1017 図 6.8 における処理 3、5 において、型変換処理 6.5 の割合を表 6.3 まとめた
 1018 この型変換処理について、あるコレクションにおける全ドキュメント数に対する Python リストへの型
 1019 変換処理時間の関係を測定した。結果を図 6.10 に示す。全ドキュメント数に対して線形性を示している
 1020 ことがわかる。方法 2 の検索処理については型変換処理 6.5 が支配的であることが分かった。

1021 6.3.2 改善点

1022 測定を踏まえ、改善方法として以下の項目を検討した。ここでは、上述したように使用しているデータ
 1023 構造や使用しているフレームワークの変更はせずに処理時間を改善することを前提としている。

- 1024 ● コレクション検索処理の回数を減らす.
- 1025 ● 検索対象コレクションのドキュメント数を減らす.

1026 上述した 2 つを目的として、以下の 2 つの方法を新しく考え処理時間測定を行った。

- 1027 3. 検索情報のコレクションに一覧表示に必要な情報を保持、参照.
- 1028 4. 方法 3 に付け加えて、検索情報ドキュメントを複数コレクションに分散、マルチスレッドを用いた
 1029 検索処理の並列化.

1030 方法 3 については一覧表示に必要な情報を検索情報のドキュメントが持つことで、データベースに対す

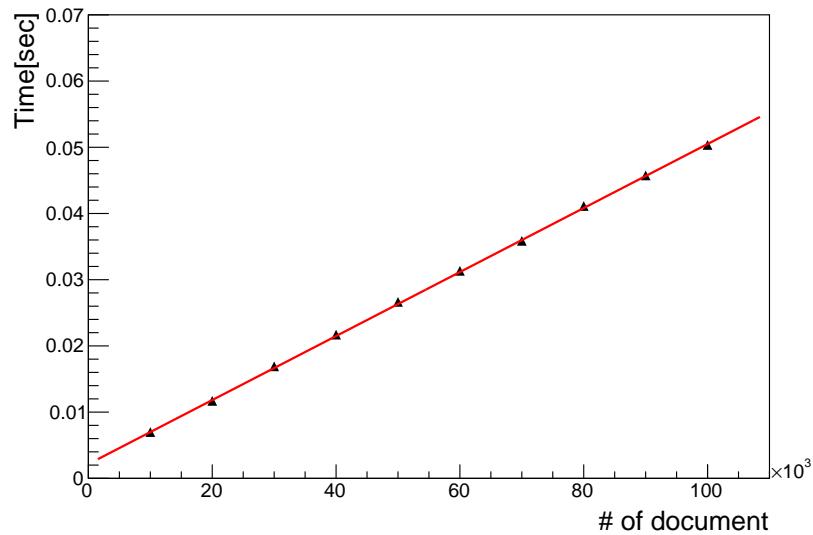


図 6.10 ドキュメント数に対する型変換処理時間の関係。コレクション検索後の型変換に要する処理時間は、図のようにドキュメント数に対して線形となっていることが分かる。

1031 る検索の回数を減らすことを目的としている。方法 4 については方法 3 の検索処理の数を減らすことに加
1032 えて、ドキュメントの数を分散し並列処理をすることで処理時間の改善を図っている。イメージをそれぞ
1033 れ図 6.11、6.12 に示す。

1034 方法 3、4 について、節 6.2 と同じ内容の測定を行った。方法 2 のものと合わせた結果を図 6.13 に示
1035 す。方法 4 について、分散するコレクション数は 10 個、スレッド数には 2 とした。方法 2 に比べて、方
1036 法 3、4 共に処理時間が改善していることがわかる。

1037 方法 3、4 を比べると傾きに差が見られる。この測定の環境では、方法 4 はドキュメント数が多くなっ
1038 た時に有効であると考えられる。方法 3 と 4 の有効性については、コンピュータの性能や OS にも依存す
1039 ると考えられるので、実際に運用を行うサーバーの環境で検討をする必要がある。

1040 方法 4 に関しては今回はコレクション数を 10、スレッド数を 2 としたが、それぞれ最適な数を検討す
1041 ることで更なる改善ができる可能性がある。

1042 得られた方法 3,4 に関する関係を式 6.6、6.7 に示す。

$$y = a_3 x + b_3 \quad (6.6)$$

$$a_3 = (2.9 \pm 0.1) \times 10^{-5}$$

$$b_3 = (5.0 \pm 1.1) \times 10^{-3}$$

$$y = ax + b \quad (6.7)$$

$$a = (2.7 \pm 0.1) \times 10^{-5}$$

$$b = (2.2 \pm 1.0) \times 10^{-2}$$

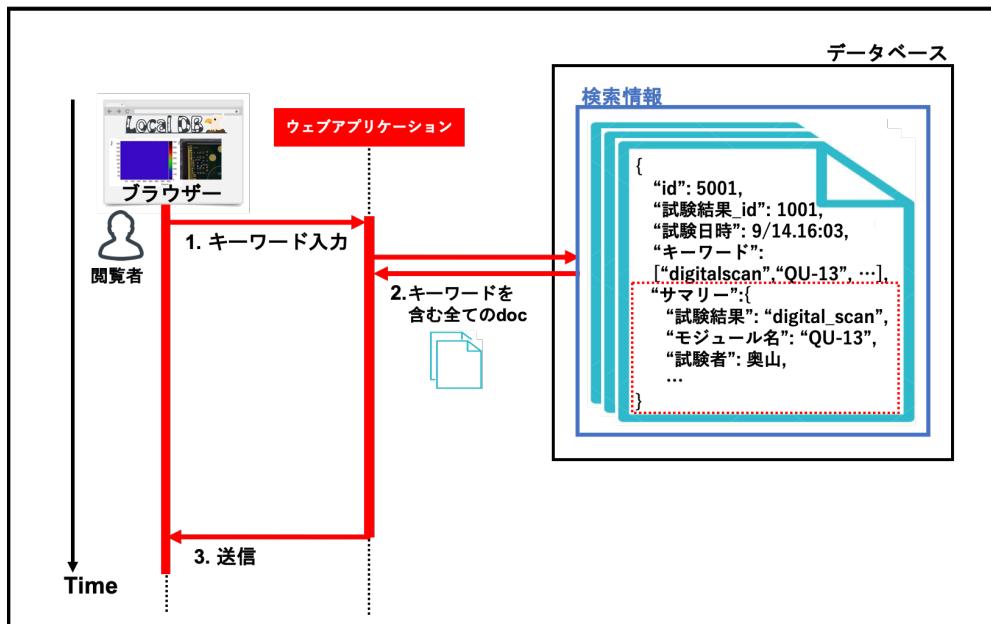


図 6.11 方法 3:検索情報と共に一覧表示に必要な情報を保持、参照。方法 2 では検索情報コレクションより、条件に一致する試験結果 ID を取得し、実際の試験結果に対して再度検索をかける流れとなっていたが、この方法では一覧表示に必要な情報も全て検索情報のコレクション内で保持する。こうすることで検索機能において必要な情報の全てが 1 つのコレクションにまとまり、コレクション検索の処理が 1 回で済むため、処理時間が改善すると考えられる。

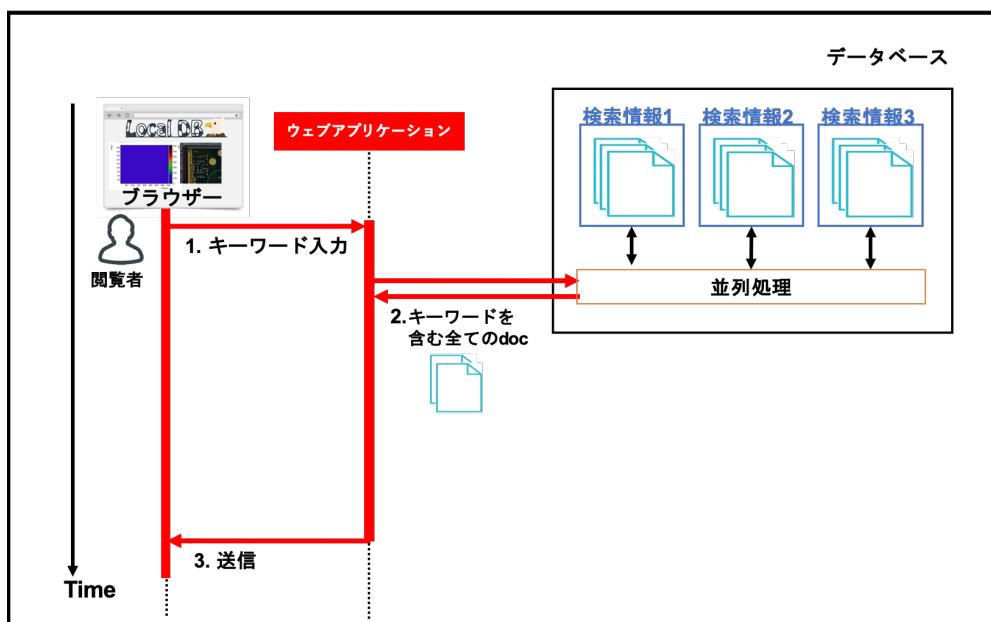


図 6.12 方法 4:検索情報コレクションを分散、マルチスレッドを使用。方法 3 に加えて、検索情報コレクションを分散し並列処理を行うことで、1 つのコレクションあたりに含まれるドキュメント数を減らし、コレクション検索にかかる時間を削減できると考えられる。

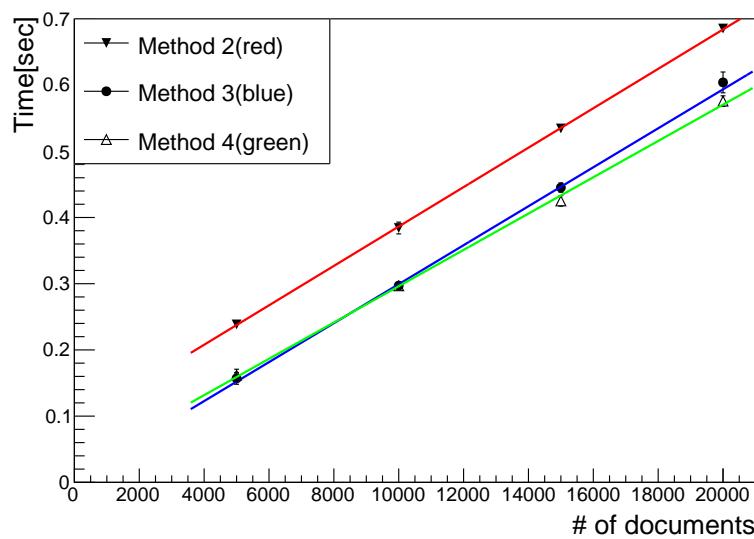


図 6.13 方法 3、4 に対する処理時間測定結果。横軸が試験結果のドキュメント数、縦軸が処理時間を表している。赤、青、緑がそれぞれ方法 2、3、4 を用いたものである。方法 2 に比べて 3、4 共に改善していることが分かる。方法 3、4 を比べると傾きに差があることが分かる。これより本測定における環境においては、ドキュメント数が大きい場合に方法 4 は有効である。

1043 6.4 本章のまとめ

1044 本章では、品質試験に特化した機能の 1 つである試験結果検索機能について、実装の方法と処理時間に
 1045 ついて述べた。方法 1 では試験結果数に対して検索処理時間が二次的に增加了。生産時に想定される
 1046 データ数である 84,000 件に対して約 2.7 時間かかる見積もりとなってしまい、検索機能としては実用に
 1047 向かない機能であった。そこで各試験に関わる検索情報を前もってあるコレクションに格納しておき、検
 1048 索処理実行時にはこれを参照する方法 2 を考案し実装した。改善後の方では試験結果数に対して処理時
 1049 間が線形となり、改善前の方法に比べ圧倒的に速くなった。84,000 件に対して約 2.6 秒と見積られ、生
 1050 産を通して使用することができる機能であることを確認した。方法 2 の機能を現在は提供している。

1051 さらなる改善に向けて、処理時間の詳細を調査したところデータベースのドキュメント取得時における
 1052 Python リスト型への変換処理に大きく時間を要していた。そこで改善策として、検索情報とともに一覧
 1053 表示に必要な情報を保持しておく方法 3、加えてマルチスレッドを使用し処理を分散する方法 4 の 2 つ考
 1054 案し、処理時間測定を行なった。どちらも現在の方法に比べて処理時間の改善が見られた。マルチスレッ
 1055 ドの方法はコンピュータの性能や OS に依存するため、実際に運用を行うサーバーによって処理時間が変
 1056 化すると考えられる。

1057 第7章

1058 中央データベースとローカルデータベースの同期

1059

1060 中央データベースとローカルデータベースの同期機能に関しての調査と機能開発を行った。詳細について
1061 以下で説明する。

1062 7.1 サーバーの設置場所による処理時間の違い

1063 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。そのため通信に要する処理時間
1064 は、組み立て機関の場所に依存すると考えられる。世界的に同期ツールが不自由なく動くことを向けて開
1065 発、改善に役立てることを目的として、データベース間の通信にかかる処理時間を測定した。組み立て機
1066 関及びローカルデータベースの設置場所はヨーロッパ、アメリカ、日本の3つの地域に分布している(付
1067 錄C)。それぞれにおける代表機関として以下の3つに設置されたサーバーを用いて調査を行った。

- 1068 • 日本、高エネルギー加速器研究所 (KEK)
- 1069 • アメリカ、ローレンス・バークレー国立研究所 [43](LBNL)
- 1070 • スイス、欧州原子核研究機構 (CERN)

1071 各サーバーの性能を表7.1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7.1に示す。測
1072 定について、各サーバーに作成した個人のアカウント上で作業を行いデータを取得した。

1073 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
1074 ネットワーク環境も生産時と同じであると仮定している。

表7.1 各ローカルデータベースサーバーの性能一覧。今回の調査に利用したサーバーの性能を示す。

KEK(日本)、LBNL(アメリカ)、CERN(スイス)に設置されたサーバーを用いた。

設置機関	CPU Type	Core	Thread	Clock speed[GHz]	Memory [GB]	Disk [GB]
KEK(日本)	Intel(R) Core(TM) i7-9700K	8	16	3.6	32.66	1800 + 1800
LBNL(アメリカ)	Intel(R) Core(TM) i7-8700	6	12	3.7	32.63	233
CERN(スイス)	Intel(R) Core(TM) i7-4790	4	8	3.6	32.69	238.5 + 3700 + 3700



図 7.1 各サーバーの設置場所。赤点で示しているのがそれぞれローカルデータベースサーバーの設置位置であり、オレンジで示しているのが中央データベースである。距離としては CERN が一番近く、 LBNL、KEK の順番となっている。

表 7.2 同期ツールの中で使用する中央データベースの主な API 一覧。同期ツールにおいて、中央データベースの情報取得には提供されているいくつかの API を用いており代表的なものを示す。この API を Python を用いて実行することで、情報取得や試験結果のアップロードをすることができる。

関数名	処理の内容	本ツールでの使用用途
getComponent	登録した部品情報の取得	主にダウンロード時におけるモジュールや FE チップの情報取得に用いる。
listComponents	登録した部品情報一覧の取得	主にダウンロード時におけるモジュール情報一覧取得に用いる。
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	ある試験結果に対するバイナリファイルの添付	読み出し試験結果ファイルを添付する際に用いる。

7.1.1 同期ツールに使用する API

中央データベースとの同期には、中央データベースで開発された API を使用している。ローカルデータベースとの同期ツールの中で使用している主な API を表 7.2 に示す。

7.1.2 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、まとめたものを表 7.3 に示す。ファイル容量と処理時間の関係を図 7.2 に示す。ここで、どの場合においても KEK における処理時間が最も長いことがわかる。そのため本研究では、開発した同期ツールの処理確認及び処理時間測定を KEK サーバーで行い、その有用性を評価することとした。

表 7.3 中央データベース API 実行時の処理時間測定結果。左の結果は表 7.2 における “getComponent” を用いてモジュール 1 つの情報を取得するのにかかった時間、右は “createTestRunAttachment” を用いて 1Byte のファイル送信にかかった時間である。どのサーバーにおいても 0.28 秒以上の処理時間がかかっていることがわかり、データベースへの接続、情報取得にかかる時間が読み取れる。3 つのサーバーを比べると、どちらの場合も KEK サーバーでの処理時間が一番大きいことが分かる。

サーバー	処理時間 [秒]	サーバー	処理時間 [秒]
KEK	0.47 ± 0.01	KEK	0.83 ± 0.01
LBNL	0.37 ± 0.02	LBNL	0.34 ± 0.03
CERN	0.28 ± 0.01	CERN	0.48 ± 0.03

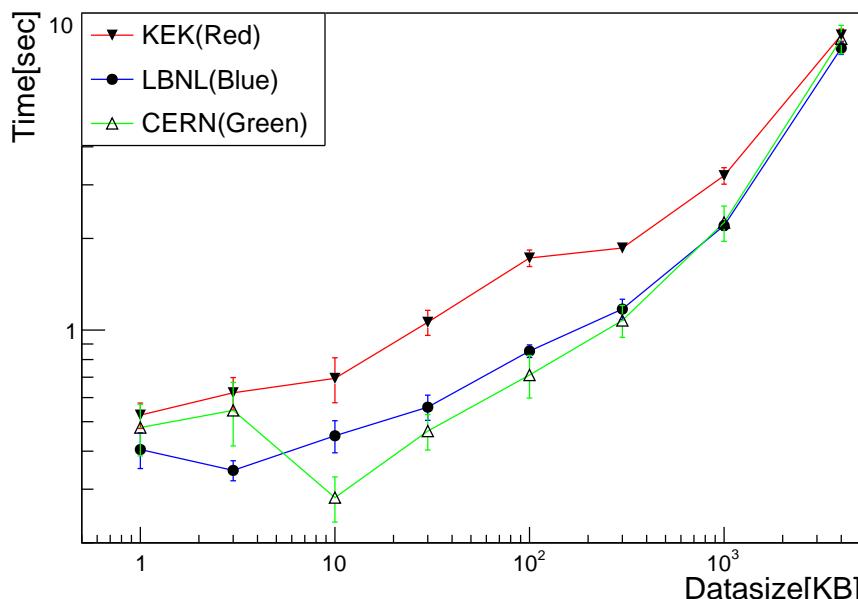


図 7.2 “createTestRunAttachment” を用いた添付処理におけるファイル容量と処理時間の関係。それぞれのサーバーで 1、3、10、30、100、300 KB、1、4 MB のファイル送信にかかる時間を測定した。どの点においても KEK サーバーが最も処理時間を要していることが分かる。このグラフについての考察を付録 E に記す。

1089 7.1.3 中央データベースウェブページ接続にかかる時間

1090 KEK、LBNL のサーバーにおいてネットワークの混雑状況等の理由による処理時間遅延の影響を調べ
1091 るために、中央データベースウェブページへの通信時間を 1 週間通して測定した。測定は 1 月 10 日午
1092 後 8 時から 1 月 18 日午後 16 時の、偶数時間 (0,2,4,6,8,10,12,14,16,18,20,22 時) に行なった。CERN の
1093 サーバーにおいては、サーバーのダウンによりデータ取得に失敗した。測定結果を図 7.3 に示す。KEK、
1094 LBNL 共に処理時間の変化は小さく、どの時間帯においても KEK の方が遅いことが分かる。

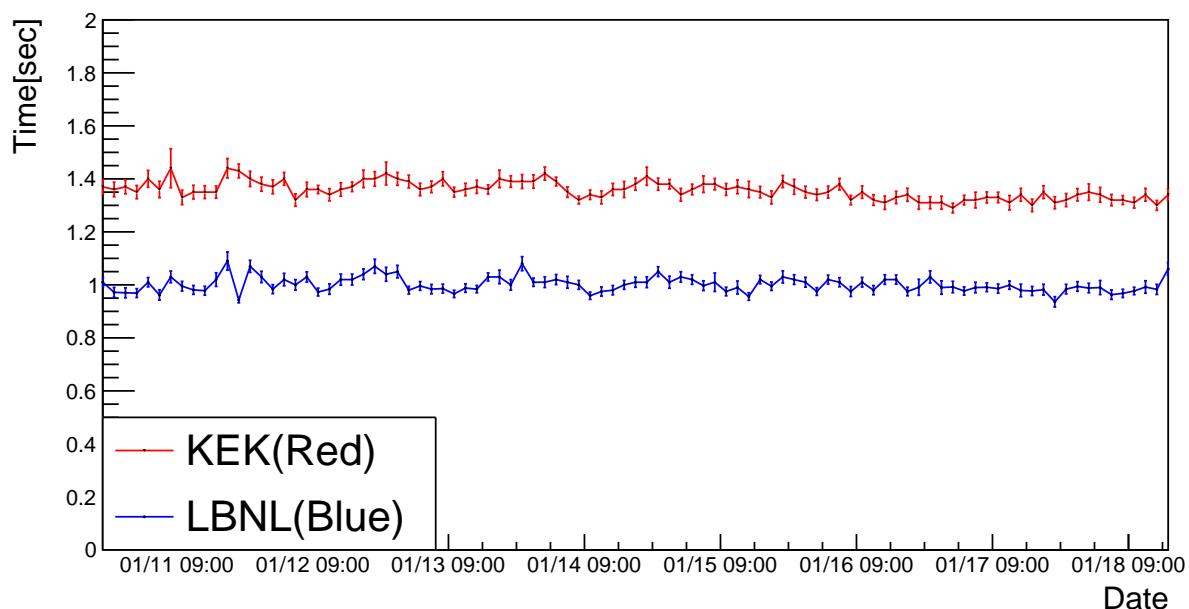


図 7.3 中央データベースウェブページへの通信時間の測定結果。横軸は測定日時、縦軸は中央データベースウェブページへの通信時間を表す。1月10日午後8時から1月18日午後16時の期間で測定を行い、各日の偶数時間にデータ取得を行なった。赤線、青線はそれぞれ KEK、LBNL のサーバーを用いた測定結果である。どちらのサーバーにおいても時間帯による通信時間の大きな変化は見られず、全ての時間帯において KEK の通信時間の方が遅いことが分かる。

表 7.4 ダウンロード機能を用いて保存する情報一覧。ダウンロード機能を用いて中央データベースからローカルデータベースに保存する情報の一覧を示している。保存の際にはモジュール、FE チップの情報がそれぞれ別のドキュメントに保存される。

部品	情報
モジュール	シリアルナンバー 搭載 FE チップの種類 登録機関 搭載 FE チップの枚数
FE チップ	シリアルナンバー FE チップ ID(モジュール上の位置を表す情報) 種類 登録機関

7.2 モジュール情報のダウンロード

7.2.1 ダウンロードする情報と構造

中央データベースから、モジュール及び FE チップの情報をダウンロードする機能を開発、実装した。ダウンロードする情報の詳細について表 7.4 に示す。

ダウンロードされたモジュール、FE チップのドキュメントの例をコード 7.1、7.2 に示す。

ソースコード 7.1 ダウンロードしたモジュール情報のドキュメントの例。ドキュメントが表 7.4 の情報を持つことが分かる。

```

100 1 {
101 2     "_id" : ObjectId("5fa79114e615fa000a1a5976"),
102 3     "name" : "20UPGR000000001",
103 4     "chipType" : "RD53A",
104 5     "serialNumber" : "20UPGR000000001",
105 6     "chipId" : -1,
106 7     "componentType" : "module",
107 8     "address" : "5fd597fdf7339bbf26b87fb2",
108 9     "children" : 1,
109 10    "sys" : {
110 11        "mts" : ISODate("2020-12-13T04:26:37.989Z"),
111 12        "cts" : ISODate("2020-12-13T04:26:37.989Z"),
112 13        "rev" : 0
113 14    },
114 15    "dbVersion" : 1.01,
115 16    "user_id" : -1,
116 17    "proDB" : true
117 18 }
```

ソースコード 7.2 ダウンロードした FE チップ情報のドキュメントの例。ドキュメントが表 7.4 の情報を持つことが分かる。

```

1120 1 {
1121 2     "_id" : ObjectId("5fa79560e615fa000a1a5a16"),
1122 3     "name" : "20UPGFC99999999",
1123 4     "chipType" : "RD53A",
1124 5     "serialNumber" : "20UPGFC99999999",
1125 6     "chipId" : 0,
1126 7     "componentType" : "front-end_chip",
1127 8     "address" : "5fd597fdf7339bbf26b87fb2",
1128 9     "children" : -1,
1129 10    "sys" : {
1130 11        "mts" : ISODate("2020-12-13T04:26:37.984Z"),
1131 12        "cts" : ISODate("2020-12-13T04:26:37.984Z"),
1132 13        "rev" : 0
1133 14    },
```

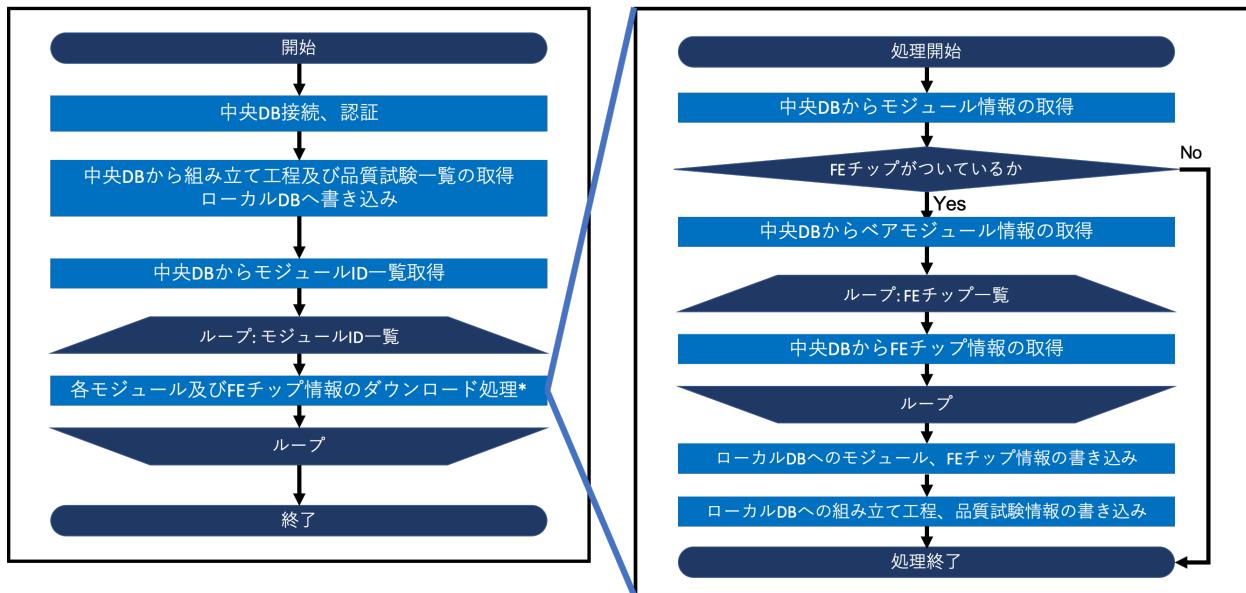


図 7.4 ダウンロード処理における流れのイメージ図。流れの中で複数回中央データベースに接続し、モジュールや FE チップの情報取得を行う。

```

1135      15     "dbVersion" : 1.01,
1136      16     "user_id" : -1,
1137      17     "proDB" : true
1138  }

```

1140 7.2.2 処理の流れ

1141 ダウンロード機能における処理の流れのイメージを図 7.4 に示す。

1142 7.2.3 機能確認

1143 KEK で組み立てられた 6 台の Quad 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 7.5、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 7.6 に示す。

Module	Bare Module	Sensor	PCB	Carrier	FE chip
20UPGM20030004	20UPGB40500019	20UPGS83300002	20UPGPQ0030004	20UPGMC0210000	20UPGFC0014659 20UPGFC0014658 20UPGFC0014675 20UPGFC0014691 20UPGFC0014644 20UPGFC0014628 20UPGFC0014660 20UPGFC0014708 20UPGFC0014677 20UPGFC0014629 20UPGFC0014693 20UPGFC0014646 20UPGFC0016282 20UPGFC0016281 20UPGFC0016279 20UPGFC0016280 20UPGFC0016278 20UPGFC0016267 20UPGFC0016235 20UPGFC0016242 20UPGFC0016276 20UPGFC0016266 20UPGFC0016220 20UPGFC0016227
20UPGM20030001	20UPGB40500020	20UPGS83300003	20UPGPQ0030001	20UPGMC0210001	
20UPGM20030003	20UPGB40500021	20UPGS83300004	20UPGPQ0030003	20UPGMC0210002	
20UPGM20030006	20UPGB40500022	20UPGS83300001	20UPGPQ0030006	20UPGMC0210003	
20UPGM20030022	20UPGB40500023	20UPGS83300005	20UPGPQ0030022	20UPGMC0210004	
20UPGM20030024	20UPGB40500024	20UPGS83300006	20UPGPQ0030024	20UPGMC0210005	

図 7.5 登録した Quad モジュールとその構成部品のシリアルナンバー一覧。左から、登録したモジュール、搭載されているベアモジュール、シリコンセンサー、PCB、モジュールキャリア、FE チップの中央データベース内でのシリアルナンバーを示している。Quad モジュールであるため、FE チップをそれぞれ 4 つ搭載している。

ProdDB web

IN PROGRESS	Module - Outer system quad module	20UPGM20030001
IN PROGRESS	Module - Outer system quad module	20UPGM20030024
IN PROGRESS	Module - Outer system quad module	20UPGM20030022
IN PROGRESS	Module - Outer system quad module	20UPGM20030006
IN PROGRESS	Module - Outer system quad module	20UPGM20030003
IN PROGRESS	Module - Outer system quad module	20UPGM20030004

 Download

LocalDBX TOP / COMPONENTS / TEST

ITk database for Yarr Component List

Input keywords Partial match Perfect match

RD53A (11 modules)

Module Name	Chip Name	Latest Result						Tag
		Current Stage	Test Type	User	Site	Date	Link	
20UPGR90020026	20UPGFC0028965 20UPGFC0028950 20UPGFC0028951 20UPGFC0028952	None	None	None	None	None		
20UPGR90000004	20UPGFC0008036 20UPGFC0007994 20UPGFC0007972 20UPGFC0008035	None	None	None	None	None		
20UPGR30000001	20UPGR33000000 20UPGR33000001 20UPGR33000002 20UPGR33000003	None	None	None	None	None		
20UPGR10099999	20UPGFC999995 20UPGFC999996 20UPGFC999997 20UPGFC999998	None	None	None	None	None		
20UPGR00000001	20UPGFC999999	None	None	None	None	None		
20UPGM20030024	20UPGFC0016276 20UPGFC0016266 20UPGFC0016220 20UPGFC0016227	None	None	None	None	None		
20UPGM20030022	20UPGFC0016278 20UPGFC0016267 20UPGFC0016235 20UPGFC0016242	None	None	None	None	None		
20UPGM20030006	20UPGFC0016282 20UPGFC0016281 20UPGFC0016279 20UPGFC0016280	None	None	None	None	None		
20UPGM20030004	20UPGFC0014659 20UPGFC0014658 20UPGFC0014675 20UPGFC0014691	None	None	None	None	None		
20UPGM20030003	20UPGFC0014677 20UPGFC0014629 20UPGFC0014693 20UPGFC0014646	None	None	None	None	None		
20UPGM20030001	20UPGFC0014644 20UPGFC0014628 20UPGFC0014660 20UPGFC0014708	None	None	None	None	None		

図 7.6 登録した Quad モジュールのダウンロードの様子。上図が中央データベースのウェブページを表しており、下図がローカルデータベースのものである。上図で登録したモジュール一覧を確認でき、赤枠で囲っているところでシリアルナンバーを見ることができる。ダウンロード実行後は下図のようにローカルデータベースで対応するモジュールを確認することができる。ローカルデータベースではモジュール情報に加えて FE チップの情報も取得するため、下図の表ではこれらのシリアルナンバーも確認できる。

表 7.5 登録したモジュールのダウンロード処理時間測定結果。登録したそれぞれのモジュールについてダウンロードにかかる時間を測定した。表より 1 つあたり平均 4 秒の時間がかかっていることが分かる。

モジュール	処理時間
20UPGM20030004	3.8
20UPGM20030001	3.7
20UPGM20030003	5.9
20UPGM20030006	3.6
20UPGM20030022	3.8
20UPGM20030024	3.3
平均	4.0 ± 0.4

1145 7.2.4 処理時間測定

1146 ダウンロードの処理時間を測定した。これについてまとめたものを表 7.5 に示す。Quad モジュール 1
 1147 つに対して平均して 4.0 ± 0.4 秒の処理時間がかかっている。

1148 7.2.5 処理時間詳細

1149 ダウンロード処理時間の改善に向けて、処理時間の詳細について以下の測定した。

- 1150 1. 中央データベースからモジュール情報の取得.
- 1151 2. データベースでの FE チップ確認処理.
- 1152 3. 中央データベースからベアモジュール情報の取得.
- 1153 4. 中央データベースから FE チップ情報の取得 (4 枚分).
- 1154 5. ローカルデータベースへの情報の書き込み (モジュール、FE チップ、品質試験情報).

1155 情報取得のイメージを図 7.7 に示す。このように Quad モジュールの場合、ダウンロードの流れの中で
 1156 合計して 7 回、データベース API を用いて情報取得を行う。

1157 結果を表 7.6 に示す。

1158 この結果より、各構成部品情報の取得 (モジュール、ベアモジュール、FE チップ) の取得にそれぞれ均
 1159 等に処理時間がかかっていることがわかった。

1160 7.2.6 生産時における見積もり

1161 現在ダウンロード機能のオプションとして、以下の 2 つを実装している。

- 1162 1. モジュール 1 つをダウンロードする機能.
- 1163 2. 登録されている全てのモジュールの一括ダウンロード機能.

1164 オプション 1 の見積もり値は、表 7.5 の平均値として 4.0 ± 0.4 [sec] となる。オプション 2 の見積もり
 1165 値は、生産時には最大で 10,000 台のモジュールが中央データベースに登録されることから、以下のよう

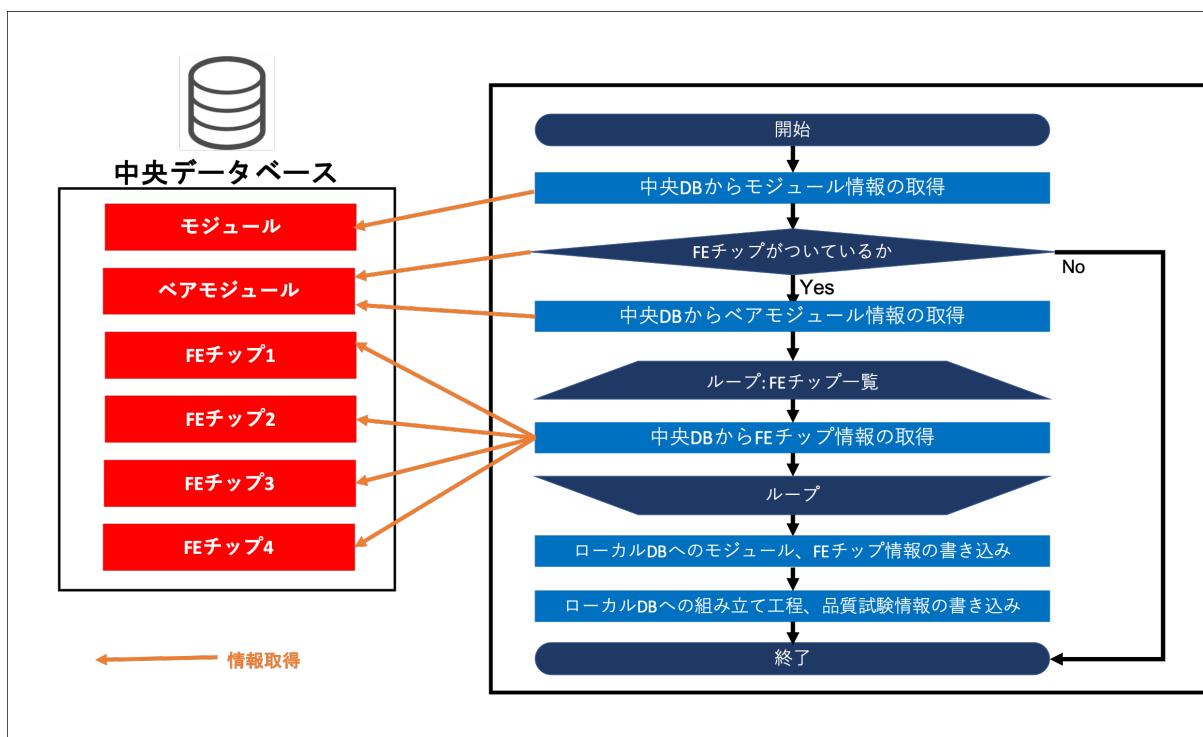


図 7.7 モジュール及び構成部品情報取得のイメージ。図のように処理の流れの中で合計 7 回中央データベースに接続し、モジュール、ペアモジュール、FE チップの情報取得を行っている。

表 7.6 ダウンロード機能における詳細処理にかかる時間測定。図 7.7 より、中央データベースに接続、情報取得を合計 7 回行っており、それが処理 1 から 4 に対応する。どの処理においても 0.5 秒程度の時間がかかることが分かる。処理 5 はローカルデータベースへの書き込み処理であるが、他の処理に比べて十分に小さいことが分かる。

処理	時間
1	0.60 ± 0.07
2	0.55 ± 0.07
3	0.61 ± 0.04
4	0.46 ± 0.04 0.71 ± 0.19 0.51 ± 0.04 0.57 ± 0.12
5	0.0025 ± 0.0011
合計	4.0 ± 0.1

1. 中央データベースからモジュール情報の取得.
2. データベースでの FE チップ確認処理.
3. 中央データベースからペアモジュール情報の取得.
4. 中央データベースから FE チップ情報の取得 (4 枚分).
5. ローカルデータベースへの情報の書き込み (モジュール、FE チップ、品質試験情報).

1166 になる。

$$(4.0 \pm 0.4)[\text{sec}] \times 10,000 = 11 \pm 1[\text{hour}] \quad (7.1)$$

1167 データベース操作の流れにおいて(図4.18)、各機関で各モジュールの組み立てを始める際に中央データベースへのモジュール登録及びローカルデータベースへモジュール情報のダウンロードを行うことを想
1168 定している。これは1つずつ行うため、このような場合はオプション1を用いる。

1169 オプション2の使用例として、モジュールの組み立て工程の途中で、複数のモジュール($O(100)$ 程度)
1170 を機関1から機関2に輸送する場合をあげる。機関2では機関1で登録された全てのモジュール情報を
1171 ダウンロードしてくる必要があるため、このオプションを使用する。輸送に要する時間として1週間と仮
1172 定すると、この時間を利用してダウンロード処理を行う必要がある。また何らかの不具合で実行途中で落
1173 ちてしまった場合、再会した際には落ちた時点の処理から行われるのが望ましい。この機能は現在実装を
1174 していないため、今後改良する必要がある。

1176 7.2.7 改善点の考案と見積もり

1177 オプション2について、11時間の処理時間を要する見積もりとなっている。より円滑な同期処理に向
1178 けて、処理時間の改善方法について考える。

1179 一括ダウンロード機能については以下の改善点が考えられる。

- 1180 1. モジュールの現在位置に対応したものののみのダウンロード.
- 1181 2. FEチップの登録機関を取得しない.
- 1182 3. モジュールのプロパティとして、ダウンロードに必要な情報を全て保存.
- 1183 4. データベースAPIを改良し、モジュール一覧取得の際に構成要素の情報を取得できるようにする.

1184 これらについて詳細と処理時間の見積もりを以下で行う。

1185 改善案1: モジュールの現在位置に対応したものののみのダウンロード

1186 中央データベースではモジュールの現在位置情報を保持している。全てのモジュール情報をダウンロー
1187 ドする必要はなく、機能実行者と位置が同じもののみをダウンロードするアルゴリズムにすれば処理時間
1188 を改善できる。見積もりとしては、ダウンロード対象となるモジュール数を n とすると、以下のように
1189 なる。

$$(11 \pm 1) \times \frac{n}{10000}[\text{hour}] \quad (7.2)$$

1190 改善案2: FEチップの登録機関を取得しない

1191 表7.7よりダウンロードの際に、FEチップの情報取得を行っている。これはFEチップ登録機関の情
1192 報を取得しローカルデータベースに保存するためであるが、この情報は組み立て機関で必ずしも保持して
1193 おきたい情報ではない。そのため、現段階ではFEチップのデータ取得処理は割愛することができる。こ
1194 れにかかる処理時間は表7.6より、合計して $2.3 \pm 0.2[\text{sec}]$ となるため、その場合オプション2の処理時
1195 間の見積もりは、以下のようになる。

$$\{(4.0 \pm 0.4) - (2.3 \pm 0.2)\}[\text{sec}] \times 10,000 = 4.9 \pm 0.8[\text{hour}] \quad (7.3)$$

1196 この改善策のデメリットとしては、FE チップの情報取得処理を省くとローカルデータベースに保存した
1197 い情報が将来的にできた場合に、保存できないことである。例えば各 FE チップの最適動作電圧のように
1198 モジュール読み出しに対して有益な情報は保存し、迅速に確認したいというシステム方針になることがあ
1199 げられる。

1200 改善案 3: モジュールのプロパティとして、ダウンロードに必要な情報を全て保存

1201 モジュールのプロパティとして、FE チップの名前等のダウンロードに必要な情報を書いておくと、表
1202 7.2 における listComponents によるモジュール一覧取得の際にその情報を参照することができる。こう
1203 することで、表 7.7 において、ペアモジュールや FE チップの情報取得を省くことができる。これらの処
1204 理時間は、合計して $2.9 \pm 0.2[\text{sec}]$ となるため、その場合オプション 2 の処理時間の見積もりは、

$$\{(4.0 \pm 0.4) - (2.9 \pm 0.2)\}[\text{sec}] \times 10,000 = 3.1 \pm 0.8[\text{hour}] \quad (7.4)$$

1205 これのデメリットは、データベースの中でデータが冗長になってしまうことである。FE チップの名前
1206 情報がモジュールのプロパティにも保存されていると、データベース内部で冗長性を持ってしまい、編集
1207 が加えられた場合などこれを管理するのが難しくなる。

1208 改善案 4: データベース API を改良し、モジュール一覧取得の際に構成要素の情報を取得できるようにする

1209 現在、表 7.2 の listComponents を用いた時にはモジュール一覧の情報は取得できるが、各モジュール
1210 に対して構成要素の情報は取得できない。そのため表 7.7 のようにモジュールごとに中央データベースに
1211 接続し、部品情報を取得している。ダウンロードに必要な情報を listComponents で一括で取得できるよ
1212 うな仕様に API の変更を行えば、中央データベースへの接続は一回ですみ、処理時間を削減できると考
1213 えられる。この場合、中央データベースの内部構造を知り、一括で取得しデータ送信をする場合にどれだ
1214 けの時間を要するかを見積もり、今の場合と比較する必要がある。

1215
1216 現段階ではモジュール組み立ての試験段階であり、組み立て機関で必要な情報、世界各地での組み立て
1217 工程の流れ等を検討している段階である。ここで述べたような改善策を組み合わせ、変更を加えていく必
1218 要がある。また中央データベースとの同期におけるネットワーク速度は地理的な距離により異なるため、
1219 この点も考慮に入れる必要がある。今回の測定は日本で行われたものであるため、他の国においては処理
1220 速度は改善すると考えられる。

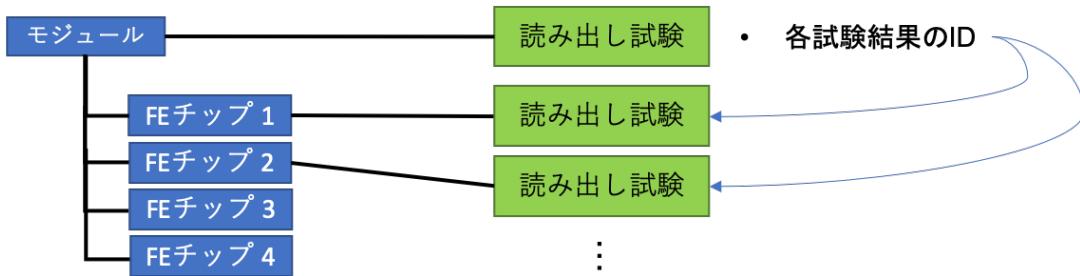


図 7.8 中央データベースにおける読み出し試験結果の構造。YARR の出力ファイル及びローカルデータベースのデータ構造において、読み出し試験結果は全て FE チップに紐つけられている。そのため、図のように中央データベースにおいてもこのデータ構造を保持する形でアップロードを行う。モジュールの結果は各 FE チップの試験結果に対する ID を持つことで紐付けを行っている。

表 7.7 中央データベースにおける読み出し試験結果に関する情報一覧。モジュール及び FE チップが中央データベース内で持つ試験結果の情報を示している。

部品	試験情報、結果	添付ファイル
モジュール	モジュール環境温度 FE チップにつく読み出し試験結果の ID	
FE チップ	ピクセル解析結果	試験結果データファイル 読み出し設定ファイル その他設定ファイル

1221 7.3 読み出し試験結果のアップロード

1222 4 章で述べたように、読み出し試験結果について中央データベースへアップロードするツールを開発し
1223 た。以下で詳細を述べる。

1224 7.3.1 アップロードする情報とその構造

1225 読み出し試験結果について、中央データベースにアップロードする情報を以下に記す。

- 1226 • 試験日時.
- 1227 • モジュール周りの環境温度.
- 1228 • ピクセル解析結果.
- 1229 • 各試験結果データファイル.
- 1230 • 読み出し設定ファイル.
- 1231 • その他設定ファイル (DB、ユーザ、組み立て機関等).

1232 中央データベースにおける読み出し試験の構造に関して、YARR を用いて行った読み出し結果は全て
1233 FE チップ毎に取得、保存される。そのため、データベースの内部でも FE チップに読み出し試験結果を
1234 紐づける構造を設け、モジュールの結果では各 FE チップの結果ページの ID を持つ構造とした。イメー
1235 ジを図 7.8 に示す。

1236 中央データベースにおいてモジュール、FE チップの試験結果が持つ情報を表 7.7 にまとめた。

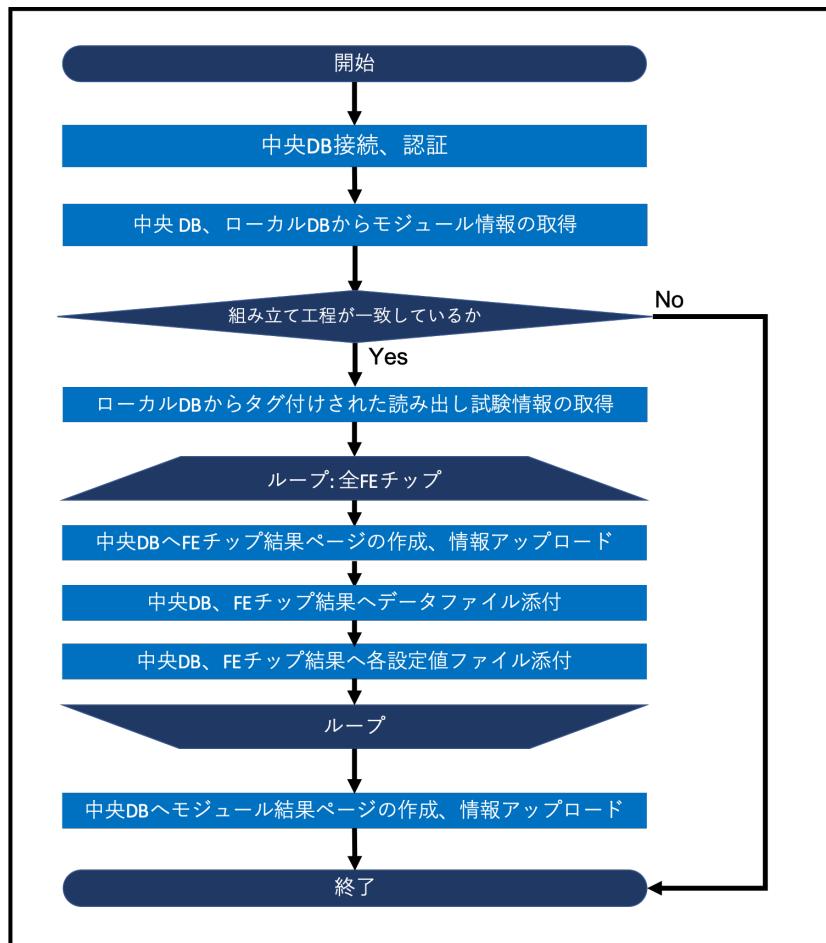


図 7.9 アップロード処理に関する流れのイメージ図。流れの中には FE チップに関するループ構造があり、ここで FE チップの結果生成、結果ファイルのアップロードを行う。最後にモジュールに対する結果生成とアップロードを行う。

1237 7.3.2 処理の流れ

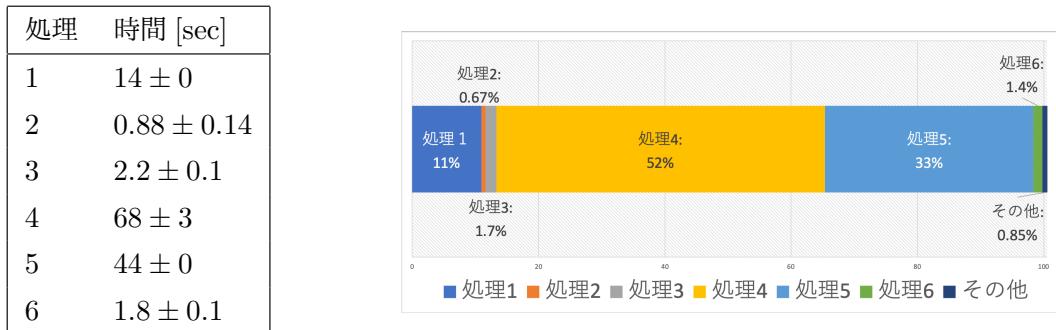
1238 アップロード機能における処理の流れのイメージを図 7.9 に示す。流れの中には共通して行われる処理
1239 と、各 FE チップに対して行われる処理がある。

1240 7.3.3 処理時間測定

1241 上述した処理のツールを開発し、処理時間測定を行った。ここで行ったアップロードする試験項目は 5
1242 章で行ったデモンストレーションのものと同じで以下の項目とする。

- 1243 • デジタル回路読み出し.
- 1244 • アナログ回路読み出し.
- 1245 • Threshold 測定.
- 1246 • ToT 測定.
- 1247 • ノイズ占有率測定.

1248 これらの項目についてアップロードを行い、その処理時間を測定した。KEK のサーバーを用いてアッ



1. 中央データベース接続、認証.
2. 中央データベース、ローカルデータベースからモジュール情報の取得.
3. 中央データベースに FE チップ結果ページの作成。結果情報をアップロード.
4. 3 で作成した結果に対して、各データファイルを添付.
5. 3 で作成した結果に対して、各設定値ファイルを添付.
6. 中央データベースにモジュールの結果ページの作成、結果情報をアップロード.

図 7.10 アップロード機能における詳細処理測定結果。アップロード機能において、各詳細処理時間を測定した結果である。左図は測定値であり、右図はそれぞれの割合を示したものである。右図より、処理 4、5 の結果ファイルの添付、設定値ファイルの添付に多く時間がかかっていることが分かる。

₁₂₄₉ プロード処理を 20 回を行い、5 項目全体でかかる時間を測定した。以下のようになった。

$$(1.3 \pm 0.0) \times 10^2 [\text{sec}] \quad (7.5)$$

₁₂₅₀ ここで処理流れの図 7.9 より特に以下の詳細処理を抜粋し、それぞれにかかる時間を測定した。

1. 中央データベース接続、認証.
2. 中央データベース、ローカルデータベースからモジュール情報の取得.
3. 中央データベースに FE チップ結果ページの作成。結果情報をアップロード.
4. 3 で作成した結果に対して、各データファイルを添付.
5. 3 で作成した結果に対して、各設定値ファイルを添付.
6. 中央データベースにモジュールの結果ページの作成、結果情報をアップロード.

₁₂₅₇ 測定値と割合をグラフにしたものと図 7.10 を示す。結果データや各設定値のファイル添付に大きく時間がかかっていることが分かった。

₁₂₅₉ これらの処理について詳細を調査するために、各ファイルにおける添付処理時間の測定を行った。測定は上述したものと同様に KEK サーバーを用いて合計 20 回行った。添付する結果ファイル、設定ファイルの種類とデータ容量、添付処理実行結果、処理時間を表 7.8 に示す。ここで 4 MB を超える容量のファイル添付は失敗していることがわかり、アップロード機能の問題点を発見した。

表 7.8: アップロード処理における結果、設定値ファイル添付実行結果と処理時間。図 7.10 より、読み出し試験に対して出力される各ファイルのアップロード実行結果、データ容量と処理時間をまとめた。ここで扱うファイルサイズの合計は 94 MB である。図よりファイルのデータ容量が大きいほど処理時間が長いことが分かる。`std_thresholdscan` のようにファイル数が多い項目の場合、合計して大きい処理時間を要することが分かる。全項目において読み出しの設定ファイルにあたる `beforeCfg_chipCfg.json`, `afterCfg_chipCfg.json` のアップロードは、中央データベースの容量制限により失敗していることが分かる。

読み出し項目	ファイル名	実行結果	容量 [KB]	処理時間 [sec]	全体 [sec]
std_digitalscan	EnMask.json	Ok	1,300	3.3 ± 0.1	17±0
	OccupancyMap.json	Ok	1,500	2.9 ± 0.2	
	L1Dist.json	Ok	0.53	0.75 ± 0.12	
	ctrlCfg_ctrlCfg.json	Ok	0.46	0.61 ± 0.08	
	dbCfg_dbCfg.json	Ok	0.60	0.69 ± 0.16	
	siteCfg_siteCfg.json	Ok	0.033	0.61 ± 0.06	
	userCfg_userCfg.json	Ok	0.14	0.66 ± 0.09	
	scanCfg_std_digitalscan.json	Ok	2.2	0.55 ± 0.06	
	beforeCfg_chipCfg.json	Error	7,200	3.0 ± 0.2	
	afterCfg_chipCfg.json	Error	7,200	4.0 ± 0.2	
std_analogscan	EnMask.json	Ok	1,300	3.9 ± 0.1	17±0
	OccupancyMap.json	Ok	1,400	2.6 ± 0.1	
	L1Dist.json	Ok	0.60	0.69 ± 0.16	
	ctrlCfg_ctrlCfg.json	Ok	0.46	0.54 ± 0.05	
	dbCfg_dbCfg.json	Ok	0.60	0.49 ± 0.04	
	siteCfg_siteCfg.json	Ok	0.033	0.48 ± 0.04	
	userCfg_userCfg.json	Ok	0.14	0.58 ± 0.08	
	scanCfg_std_analogscan.json	Ok	2.1	0.45 ± 0.03	
	beforeCfg_chipCfg.json	Error	7,200	2.9 ± 0.2	
	afterCfg_chipCfg.json	Error	7,200	3.9 ± 0.3	
std_thresholdscan	Scurve-30-96.json	Ok	0.98	1.3 ± 0.1	49±1
	Scurve-110-96.json	Ok	0.98	0.45 ± 0.03	
	Scurve-70-96.json	Ok	0.98	0.47 ± 0.04	
	Scurve-150-96.json	Ok	1.0	0.46 ± 0.04	
	Scurve-190-96.json	Ok	1.0	0.64 ± 0.12	
	Scurve-230-96.json	Ok	1.0	0.49 ± 0.03	
	Scurve-270-96.json	Ok	1.0	0.47 ± 0.04	
	Scurve-310-96.json	Ok	1.0	0.47 ± 0.04	
	Scurve-350-96.json	Ok	1.0	0.49 ± 0.03	
	Scurve-390-96.json	Ok	1.0	0.52 ± 0.06	
	Scurve-40-96.json	Ok	1.0	0.46 ± 0.03	
	Scurve-80-96.json	Ok	0.99	0.68 ± 0.13	
	Scurve-120-96.json	Ok	1.0	0.54 ± 0.07	
	Scurve-160-96.json	Ok	1.0	0.51 ± 0.05	
	Scurve-200-96.json	Ok	1.0	0.49 ± 0.04	
	Scurve-240-96.json	Ok	1.0	0.50 ± 0.05	
	Scurve-280-96.json	Ok	1.0	0.48 ± 0.04	
	Scurve-320-96.json	Ok	1.0	0.49 ± 0.05	
	Scurve-360-96.json	Ok	1.0	0.49 ± 0.06	
	Scurve-400-96.json	Ok	1.0	0.45 ± 0.05	
	Scurve-10-96.json	Ok	1.0	0.42 ± 0.03	
	Scurve-50-96.json	Ok	0.99	0.49 ± 0.05	
	Scurve-90-96.json	Ok	0.99	0.46 ± 0.05	
	Scurve-130-96.json	Ok	1.0	0.47 ± 0.05	
	Scurve-170-96.json	Ok	1.0	0.52 ± 0.04	
	Scurve-210-96.json	Ok	1.0	0.51 ± 0.04	
	Scurve-250-96.json	Ok	1.0	0.58 ± 0.10	
	Scurve-290-96.json	Ok	1.0	0.64 ± 0.13	
	Scurve-330-96.json	Ok	1.0	0.64 ± 0.09	
	Scurve-370-96.json	Ok	1.0	0.49 ± 0.06	
	Scurve-60-96.json	Ok	0.99	0.51 ± 0.06	
	Scurve-100-96.json	Ok	1.0	0.48 ± 0.05	
	Scurve-140-96.json	Ok	1.0	0.48 ± 0.06	
	Scurve-180-96.json	Ok	1.0	0.52 ± 0.06	
	Scurve-220-96.json	Ok	1.0	0.54 ± 0.05	
	Scurve-260-96.json	Ok	1.0	0.51 ± 0.05	
	Scurve-300-96.json	Ok	1.0	0.66 ± 0.09	
	Scurve-340-96.json	Ok	1.0	0.51 ± 0.06	
	Scurve-380-96.json	Ok	1.0	0.55 ± 0.05	
	sCurve-0.json	Ok	49	1.0 ± 0.1	
	ThresholdDist-0.json	Ok	4.6	0.56 ± 0.06	
	ThresholdMap-0.json	Ok	2,200	4.3 ± 0.1	
	NoiseDist-0.json	Ok	2.3	0.42 ± 0.04	
	Chi2Map-0.json	Ok	2,300	4.5 ± 0.1	
	StatusMap-0.json	Ok	1,300	2.8 ± 0.1	
	StatusDist-0.json	Ok	0.49	0.48 ± 0.04	
	NoiseMap-0.json	Ok	2,200	4.0 ± 0.2	
	Chi2Dist-0.json	Ok	1.1	0.50 ± 0.04	
	TimePerFitDist-0.json	Ok	3.1	0.56 ± 0.12	
	ctrlCfg_ctrlCfg.json	Ok	0.46	0.49 ± 0.05	
	dbCfg_dbCfg.json	Ok	0.60	0.52 ± 0.06	

表 7.8: アップロード処理における結果、設定値ファイル添付実行結果と処理時間。図 7.10 より、読み出し試験に対して出力される各ファイルのアップロード実行結果、データ容量と処理時間をまとめた。ここで扱うファイルサイズの合計は 94 MB である。図よりファイルのデータ容量が大きいほど処理時間が長いことが分かる。`std.thresholdscan` のようにファイル数が多い項目の場合、合計して大きい処理時間を要することが分かる。全項目において読み出しの設定ファイルにあたる `beforeCfg.chipCfg.json`, `afterCfg.chipCfg.json` のアップロードは、中央データベースの容量制限により失敗していることが分かる。

	siteCfg_siteCfg.json	Ok	0.033	0.54 ± 0.07	
	userCfg-userCfg.json	Ok	0.14	0.60 ± 0.07	
	scanCfg_std_thresholdscan.json	Ok	2.2	0.46 ± 0.03	
	beforeCfg.chipCfg.json	Error	7,200	3.2 ± 0.2	
	afterCfg.chipCfg.json	Error	7,200	3.5 ± 0.1	
std_totscan	MeanTotMap-0.json	Ok	1,900	4.8 ± 0.1	20±0
	SigmaTotMap-0.json	Ok	2,200	4.1 ± 0.2	
	MeanTotDist-0.json	Ok	0.59	0.54 ± 0.07	
	SigmaTotDist-0.json	Ok	1.8	0.47 ± 0.03	
	L1Dist.json	Ok	0.59	0.60 ± 0.08	
	ctrlCfg_ctrlCfg.json	Ok	0.46	0.47 ± 0.03	
	dbCfg_dbCfg.json	Ok	0.60	0.67 ± 0.24	
	siteCfg_siteCfg.json	Ok	0.033	0.59 ± 0.10	
	userCfg-userCfg.json	Ok	0.14	0.56 ± 0.05	
	scanCfg_std_totscan.json	Ok	2.0	0.59 ± 0.10	
	beforeCfg.chipCfg.json	Error	7,200	3.0 ± 0.1	
	afterCfg.chipCfg.json	Error	7,200	3.9 ± 0.3	
std_noisescan	Occupancy.json	Ok	1,300	4.0 ± 0.1	18±0
	NoiseOccupancy.json	Ok	1,300	2.5 ± 0.1	
	NoiseMask.json	Ok	1,300	2.4 ± 0.1	
	ctrlCfg_ctrlCfg.json	Ok	0.46	0.57 ± 0.08	
	dbCfg_dbCfg.json	Ok	0.60	0.55 ± 0.06	
	siteCfg_siteCfg.json	Ok	0.033	0.53 ± 0.04	
	userCfg-userCfg.json	Ok	0.14	0.61 ± 0.07	
	scanCfg_std_noisescan.json	Ok	1.4	0.53 ± 0.05	
	beforeCfg.chipCfg.json	Error	7,200	2.8 ± 0.2	
	afterCfg.chipCfg.json	Error	7,200	3.5 ± 0.1	

1263 生産時におけるアップロード処理時間の見積もり

1264 Quad モジュールにおける読み出し試験結果アップロード処理合計時間の見積もりを行った。上述した
 1265 測定は SCC であるため FE チップに対する処理(処理 3、4、5)は 1 回であるため、Quad モジュールの
 1266 場合は、図 7.10 を用いて以下のように計算できる。

$$\begin{aligned} \text{FE チップ処理} &: \left\{ (2.2 + 68 + 44) \pm \sqrt{(0.1)^2 + 3^2 + 0^2} \right\} \times 4 \\ &= (4.6 \pm 0.1) \times 10^2 [\text{sec}] \end{aligned} \quad (7.6)$$

$$\begin{aligned} \text{合計} &: (4.6 \pm 0.1) \times 10^2 [\text{sec}] + (14 \pm 0) + (0.88 \pm 0.14) + (1.8 \pm 0.1) \\ &= 7.9 \pm 0.1 [\text{min}] \end{aligned} \quad (7.7)$$

1267 モジュール読み出し試験 1 回に対して、約 8 分程度かかる見積もりとなった。円滑なモジュール組み立
 1268 て、データ管理を行うために同期は速やかに行うことが要求されること、外観検査や平坦性測定のように
 1269 他の品質試験も同期する必要があることを考慮し、処理時間の改善を試みた。詳細を以下で示す。

1270 7.3.4 改善策

1271 表 7.8 より、データ容量の大きいものにアップロード時間がかかっていることがわかる。また添付処理
 1272 を行うオフセットがあることから、std_thresholdscan のように各容量が大きくなくてもファイル数が
 1273 多いものにはアップロード時間が合計して多くかかってしまうことがわかる。

1274 これらのことと 4 MB 以上の容量を持つファイル添付処理の失敗をなくすために、次のような改善策を
 1275 考えた。

- 1276 各試験項目に対する結果データ、設定ファイルをそれぞれ Zip ファイルに統合し、圧縮後にアップ
 1277 ロードを行う。

1278 こうすることで、アップロードするファイルの容量、数共に削減することができる。圧縮率によっては
 1279 アップロード処理の失敗もなくすことができると考えた。

1280 これを踏まえアップロードツールを改良し、再び各ファイルの添付処理にかかる時間を測定した。SCC
 1281 を用いた 5 つの読み出し試験項目のアップロードにかかる合計処理時間は以下のようになつた。改善前は
 1282 $(1.3 \pm 0.0) \times 10^2 [\text{sec}]$ (式 7.5) であった。

$$36 \pm 1 [\text{sec}] \quad (7.8)$$

1283 また各ファイル添付の処理結果、時間を表 7.9 に示す。全てのファイルのアップロードに成功した。

表 7.9: アップロード処理改善後における結果、設定値ファイル添付実行結果と処理時間。各試験結果毎に結果ファイル、設定値ファイルを Zip ファイルにまとめアップロードする処理とした。これにより、ファイル数、容量の削減に成功し、アップロード時間が改善した。この時、ファイル容量の合計は 3.9 MB となつた。全てのファイルのアップロードに成功していることが分かる。

読み出し項目	ファイル名	実行結果	容量 [KB]	処理時間 [sec]	全体 [sec]
std_digitalscan	std_digitalscan_datafiles.zip	Ok	10	0.77 ± 0.18	1.9 ± 0.2
	std_digitalscan_configfiles.zip	Ok	56	1.1 ± 0.1	
std_analogscan	std_analogscan_datafiles.zip	Ok	46	1.0 ± 0.2	2.2 ± 0.3
	std_analogscan_configfiles.zip	Ok	58	1.2 ± 0.2	
std_thresholdscan	std_thresholdscan_datafiles.zip	Ok	1,500	2.6 ± 0.1	3.5 ± 0.1
	std_thresholdscan_configfiles.zip	Ok	190	0.86 ± 0.08	
std_totscan	std_totscan_datafiles.zip	Ok	730	1.7 ± 0.2	2.5 ± 0.2
	std_totscan_configfiles.zip	Ok	190	0.83 ± 0.15	
std_noisescan	std_noisescan_datafiles.zip	Ok	19	0.56 ± 0.07	1.7 ± 0.1
	std_noisescan_configfiles.zip	Ok	190	1.1 ± 0.1	

1284 改善後における処理時間の見積もり

1285 上記の見積もりと同様に、改善後のツールにおけるアップロード時間の見積もりを行った。表 7.9 より、結果ファイル、設定値ファイルの添付(処理 4、処理 5)にかかる時間はそれぞれ合計で 6.7 ± 1.1
1286 、 $5.1 \pm 0.3[\text{sec}]$ となった。中央データベースにおける FE チップの結果生成(処理 3)にかかる時間は改
1287 善前と変わらないとし、図 7.10 より $2.2 \pm 0.1[\text{sec}]$ とすると、4 枚の FE チップにかかる処理時間は合計
1288 で以下のように計算できる。

$$\begin{aligned} & \left\{ (2.2 + 6.7 + 5.1) \pm \sqrt{(0.1)^2 + (1.1)^2 + (0.3)^2} \right\} \times 4 \\ & = 56 \pm 5[\text{sec}] \end{aligned} \quad (7.9)$$

(7.10)

1290 その他の処理も合わせて全体に要する処理時間は以下のようになる。

$$\begin{aligned} & (56 \pm 5)[\text{sec}] + (14 \pm 0) + (0.88 \pm 0.14) + (1.8 \pm 0.1) \\ & = 1.2 \pm 0.1[\text{min}] \end{aligned} \quad (7.11)$$

1291 約 1 分でアップロードを完了できる見積もりとなった。改善前は $7.9 \pm 0.1[\text{min}]$ (式 7.7) であり、約
1292 15% の処理時間となった。現在は改善後の方針を用いたツールを提供している。

1293 その他の項目における処理時間

1294 改善策として最も処理時間を要していたファイル添付に関しての検討を行なったが、以下のその他の処
1295 理における処理時間について述べる。

- 1296 1. ローカルデータベース内部処理.
- 1297 2. 中央データベースへの接続、認証処理.
- 1298 3. ファイル添付以外の処理に関して、中央データベース API 使用.
- 1299 4. ネットワーク速度の改善.

1300 項目 1 に関して、読み出し試験におけるローカルデータベースの内部構造は世界的に使われているもの
1301 である。内部構造の変更をすることなく改善を図りたいという開発方針から、この項目に関して本研究で
1302 は処理時間改善の検討をしなかった。また図 7.10 よりローカルデータベース内部処理は十分に小さいも
1303 のであったため、検討の必要がないと考えた。

1304 項目 2 に関して、中央データベースへの接続、認証処理は中央データベースの API 及び中央データ
1305 ベースの内部構造によるものである。改善には、これらの変更を検討する必要があるが本研究では開発を
1306 行なっていない範囲である。

1307 項目 3 に関して、アップロード機能ではファイル添付以外に、以下の処理の際に中央データベースの
1308 API を使用している。

- 1309 • モジュール情報の取得.
- 1310 • 結果ページの作成(モジュールと FE チップ).

1311 これらの処理は必要であるため、API の使用回数の減らし処理時間を削減することは難しい。

1312 項目4に関して、中央データベースと通信する全ての処理はネットワーク速度に依存していると考えら
1313 れる(付録E)。サーバーが置かれている位置やネットワーク環境の改善により、本ツールの処理速度も改
1314 善すると考えられる。

1315 7.4 本章のまとめ

1316 本章では中央データベースとローカルデータベースの同期ツールに関する調査、開発の詳細と処理時間
1317 測定について述べた。

1318 同期ツールの調査に関して、KEK、LBNL、CERNのサーバーを用いて中央データベースAPIの使用
1319 にかかる処理時間測定を行なった。どの場合においてもKEKのサーバーにおける処理時間が長かった。
1320 さらに、KEK、LBNLのサーバーを用いて、ネットワークの通信時間の変化を調べるために中央データ
1321 ベース通信時間測定を1週間通して行なった。時間帯における通信時間の大きな変化は見られず、どの
1322 時間帯においてもKEKのサーバーの方が遅かった。以上よりツールの処理時間に関して、KEKのサー
1323 バーで有用性を検討した。

1324 同期ツール開発の詳細と処理時間測定に関して、同期ツールとしてモジュール情報のダウンロード、読み
1325 出し試験結果のアップロード機能を開発した。

1326 モジュール情報ダウンロード機能に関して、KEKで実際に作られた6つのモジュールを中央データ
1327 ベースに登録し、ダウンロード機能が正常に動作することを確認した。この際に処理時間を測定し、モ
1328 ジュール1つあたり平均 $4.0 \pm 0.4[\text{sec}]$ であった。処理時間の改善策をいくつか考案し、それについて
1329 の見積もりを行なった。

1330 読み出し試験結果アップロード機能に関して、5章で行なったデモンストレーションの読み出し試験結
1331 果を用いて処理時間測定とQuadモジュールにおける見積もりを行なった。初めは処理時間の見積もり値
1332 が $7.9 \pm 0.1[\text{min}]$ となった。処理時間の詳細を調査したところファイル添付処理に大きく時間を要してお
1333 り、ファイル容量、ファイル数に依存していた。そのため、添付ファイルを1つのZipファイルにまとめ
1334 圧縮し添付を行うことで処理時間の改善を図った。圧縮前後のファイル容量は94、3.9[MB/FEチップ]
1335 となった。処理時間を測定し、Quadモジュールについて見積もりを行なったところ $1.2 \pm 0.1[\text{min}]$ とな
1336 り処理時間の改善に成功した。またファイル添付以外の処理における処理時間について述べた。

1337 第8章

1338 まとめ

1339 8.1 本論文のまとめ

1340 HL-LHCに向けてATLAS内部飛跡検出器の総入れ替えを予定しており、これに向けてピクセルモ
 1341 デュールを世界で約10,000台生産する予定である。各モジュールに対して品質試験を行い、全てのモ
 1342 デュール及び品質試験の結果は中央データベースに保存する。

1343 本研究では、この生産及び品質試験に向けてデータベースシステムの構築を行った。各組み立て機関に
 1344 てデータ管理をするローカルデータベースを確立し、品質試験結果検索や中央データベースとの同期機能
 1345 など、生産時に必要となる諸ツールの開発を行った。

1346 開発した諸ツールを含め、生産において必要な機能の確認を行った。本番を想定したソフトウェア、
 1347 ハードウェアのセットアップと各ツールの処理を実行し、機能が使用可能であることを確認した。

1348 主な開発項目の1つ目として品質試験検索機能を述べた。開発当初はデータベース内部構造により試験
 1349 結果数 n に対して処理時間が $O(n^2)$ となり、運用時における見積もりが $(9.8 \pm 0) \times 10^3[\text{sec}]$ となって
 1350 しまい、実用に向かない機能であった。MongoDB内に新しいコレクションを設け検索に必要な情報を予
 1351 めドキュメントに保持しておくことにより、処理時間の改善に成功した。実際に処理時間の測定を行い、
 1352 データ数の増加に対しても検索機能が不都合なく使えることを確認した。本番を想定した見積もりを行
 1353 い、84,000件のデータ数に対して $2.6 \pm 0.1[\text{sec}]$ で処理が実行できる見込みであり、生産時において有用
 1354 な機能であることを確認した。

1355 2つ目に中央データベースとローカルデータベースの同期ツールを開発した。世界的に使われるツール
 1356 であり、全ての組み立て機関でこのツールをサポートするために中央データベースへの通信処理時間調査
 1357 をKEK、LBNL、CERNのサーバーを用いて行った。KEKのサーバーを用いた場合に最も時間がかかる
 1358 ことを確認した。

1359 同期ツールに関して、モジュール情報のダウンロード、読み出し試験結果のアップロード機能を開発
 1360 した。モジュール情報のダウンロード機能に関して、KEKサーバーを用いてその処理時間測定を行い、
 1361 Quadモジュール1つあたり $4.0 \pm 0.4[\text{sec}]$ の処理時間がかかるることを確認した。処理の詳細を調査する
 1362 と、モジュールや構成部品の情報取得するために行っている中央データベースAPIの使用に時間がか
 1363 かっていた。処理時間の改善策をいくつか考案し、それぞれについて見積もりを行った。読み出し試験
 1364 結果のアップロード機能に関して、開発当初は読み出し試験結果5項目のアップロードにおける処理時
 1365 間の見積もり値が $7.9 \pm 0.1[\text{min}]$ であった。処理時間改善に向けて処理の詳細を調査したところ、結果
 1366 ファイルの添付処理に大きく時間が要していることが分かった。ファイル添付についての詳細を調べる
 1367 と、添付処理時間がファイル数とファイル容量に依存していた。このことから結果ファイルをZIPファ

1368 イルにまとめ、圧縮しアップロードを行うことで処理時間の改善を図った。ここで圧縮前後のファイル容
1369 量は、それぞれ 94、3.9[MB/FE チップ] となった。結果として処理時間の改善に成功し、その見積もり
1370 値が $1.2 \pm 0.1[\text{min}]$ となった。

1371 8.2 現状と今後の課題

1372 8.2.1 ソフトウェアリリースとユーザサポート

1373 本論文で述べたツールの他に、読み出し試験コマンド統括ソフト、品質試験結果アップロード用ソフト
1374 などの開発もチームとして行っている。全てのソフトウェアを含めて、品質試験のデータ管理を達成する
1375 ようなアプリケーションスイートを目指している。2020 年 12 月 9 日にファーストバージョンのリリース
1376 を行い、いくつかの機関で全体のシステム及びソフトウェアが使われている現状である。

1377 また CERN で行ったチュートリアルを経て、世界的に機能普及が進んでいる。そのためユーザサポー
1378 トとしてソフトウェア使用のためのドキュメント [44] の作成、整備も行っている。開発者の連絡先やロー
1379 カルデータベース専用掲示板へのリンクもドキュメントに記している。何か問題が生じた時などに簡単に
1380 質問が可能となる仕組みを整えている。

1381 8.2.2 開発課題

1382 本研究では検索機能や同期機能など、読み出し試験を対象とした機能を重点的に開発した。ローカル
1383 データベース開発は、読み出し試験の結果を管理したいという要求から始まり、現在はそれ以外の品質試
1384 験も含め、全ての試験結果や組み立て工程の管理も目標としている。今後の開発課題として以下の機能を
1385 あげる。

- 1386 • 読み出し試験以外 (外観検査、平坦性測定等) の結果同期機能.
- 1387 • 中央データベースからローカルデータベースへ品質試験結果の同期.
- 1388 • 品質試験結果解析とモジュール選別機能.
- 1389 • 組み立て工程管理を世界的にサポート.

1390 最後の項目に関して、モジュールの組み立て工程は各機関ごとに異なるため、全ての現場における工程
1391 を調査しそれをサポートするシステムを実装する必要がある。例えば、日本では「ペアモジュール・フレ
1392 キシブル基板貼り付け」工程の後に、モジュール冷却のための構造である「セル搭載」を行う予定であり、
1393 これは他の組み立て機関とは異なる。各地域における柔軟な生産を許しているため、組み立て工程は世界
1394 的に細かく統一されていない。データベースシステムでは多様な組み立て工程に対応できるシステムとす
1395 る必要がある。

1396 柔軟な組み立て工程のサポートに向けて、中央データベースには選択可能な組み立て工程を定義する機
1397 能が存在し、これを用いることを考えている。そのイメージを図 8.1 に示す。

1398 以下の開発項目をあげる。

- 1399 • 中央データベースにおいて選択可能な工程定義機能を使い、全ての組み立て機関における工程をサ
1400 ポートする構造を設計、実装.
- 1401 • 本研究で開発した同期ツールを拡張、ローカルデータベース上にも中央データベースと同様の組み
1402 立て工程構造を保持.

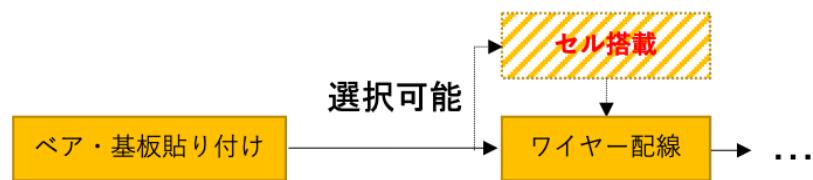


図 8.1 中央データベースにおける選択可能な組み立て工程のイメージ。図に示すように中央データベースでは選択可能な組み立て工程を定義することができる。図ではセル搭載が選択可能となっていて、ペア・基板貼り付け工程の後に、どちらの工程に進むのかを選択できる。この機能を用いて世界的に多様な組み立て工程をサポートすることを考えている。

1403 これらを達成することにより、全ての機関における組み立て工程情報の管理ができるようになると考え
1404 ている。

1405 付録 A

1406 シリコン検出器の原理

1407 A.1 半導体

1408 この説では半導体 [15] について述べる。固体は、絶縁体、半導体、導体の 3 つに大別できる。物質の電
1409 気伝導度に関して、絶縁体は非常に低い値 ($10^{-18} \sim 10^{-8}$ S/cm)、導体は高い値 ($10^4 \sim 10^6$ S/cm) を持
1410 つ。半導体の電気伝導度はこれらの中間であり、温度、光、磁界および微量の不純物に対し非常に敏感で
1411 ある。この特徴のために半導体はエレクトロニクスにおける最も重要な材料の 1 つになっている。半導体
1412 は元素半導体と化合物半導体に分けられ、多くの物質がその候補となる。元素半導体の中で代表的なもの
1413 として Si があげられ、ATLAS ピクセル検出器に使われる半導体は Si がベースとなっている。不純物が
1414 入っていない、全ての原子が Si の半導体を真性半導体と呼ぶ。真性半導体中の Si は隣り合う 4 つの Si
1415 と共有結合を構成し、結晶を作る。(図 A.1)

1416 真性半導体に対し、As などの最外殻電子を 5 つもつ原子を不純物としてドープしたものを n 型半導体、
1417 B などの 3 つのものをドープしたものを p 型半導体と呼ぶ。それぞれキャリアとして電子、ホールを持
1418 つことになり、キャリア移動の特性を組み合わせて様々なデバイスに応用することができる。

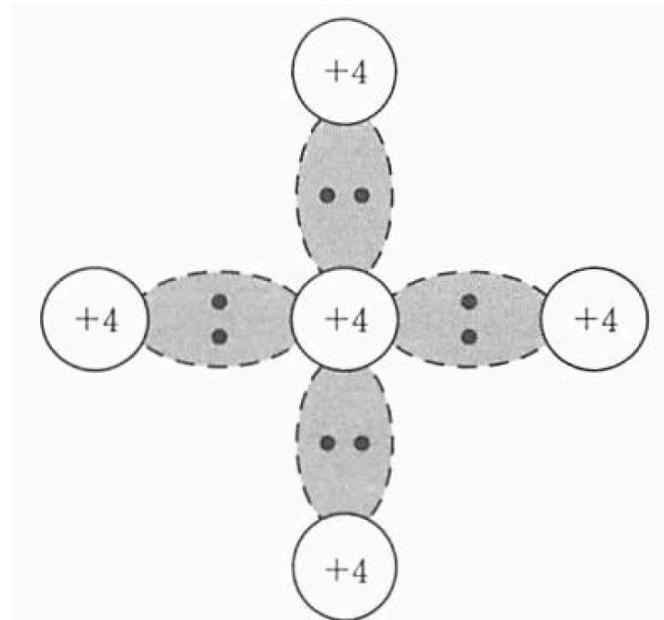


図 A.1 真性半導体中のシリコン [15]。

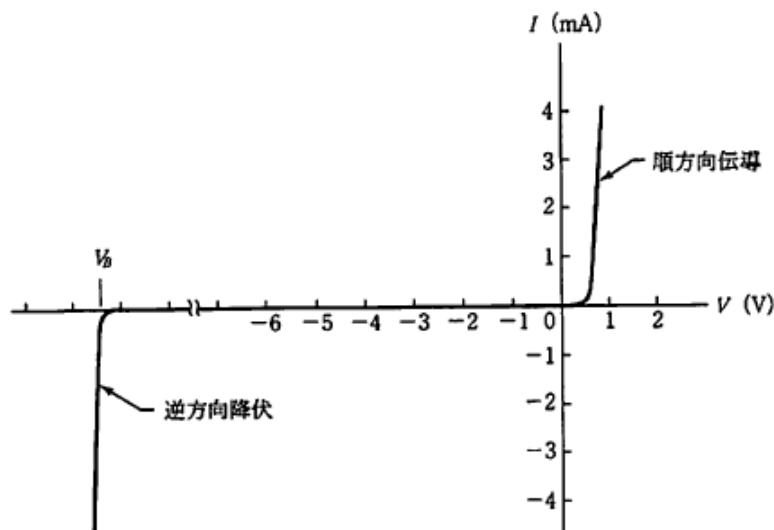


図 A.2 pn 接合の電流 - 電圧特性 [15]。



図 A.3 空乏層 [15]。

1419 A.1.1 pn 接合

1420 n 型半導体と p 型半導体を接合し、その接合部を pn 接合と呼ぶ。この接合は各種半導体素子で様々な
 1421 形で応用されており、ピクセル検出器にも用いられている。

1422 pn 接合の最も重要な特徴は特定の方向にだけ電流が流れやすい整流性である。図 A.2 に示すように正
 1423 電圧をかけると電流は急速に増加する。逆方向にかけた場合、始めのうちは電流はほとんど流れない。あ
 1424 る臨界電圧に達すると電流は急激に増大する。

1425 逆方向電圧をかけた場合、図 A.3 に示すように pn 接合付近はキャリアが存在しない空乏層領域が形成
 1426 される。この時、それぞれの半導体のエネルギー準位に差が生じている状態となっている。印加電圧 V
 1427 と空乏層幅 W は以下のようない関係がある。

$$W \propto \sqrt{V} \quad (\text{A.1})$$

1428 A.2 検出原理

1429 荷電粒子が物質中を通過するとき、以下の Bethe-Bloch の公式によってエネルギーを損失する [16]。

$$-\left\langle \frac{dE}{dx} \right\rangle = K z^2 \frac{Z}{A} \frac{1}{\beta^2} \left(\frac{1}{2} \ln \frac{2m_e c^2 \beta^2 \gamma^2 T_{\max}}{I^2} - \beta^2 + \dots \right) \quad (\text{A.2})$$

$\frac{dE}{dx}$: 荷電粒子のエネルギー損失量 [$\text{eV} \cdot \text{g}^{-1} \cdot \text{cm}^2$].

K : $4\pi N_A r_e^2 m_e c^2 = 0.307075 [\text{MeVcm}^2]$.

z : 荷電粒子の電荷量.

Z : 物質の原子番号 (Si 14).

A : 物質の原子量 (Si 28).

$m_e c^2$: 電子の静止エネルギー (0.511 MeV).

β : 光速を 1 とした入射粒子の速度.

γ : ローレンツ因子 $1/\sqrt{1 - \beta^2}$.

I : 励起エネルギーの期待値 (Si 137eV).

(A.3)

1430 また T_{\max} は質量 M の入射粒子による 1 つの電子への最大運動エネルギー移行であり、以下の式で書
1431 ける。

$$T_{\max} = \frac{2m_e c^2 \beta^2 \gamma^2}{1 + 2\gamma m_e / M + (m_e / M)^2} \quad (\text{A.4})$$

1432 荷電粒子が半導体を通過したとき、そのエネルギー損失に応じて電子・ホール対が生成する。

1433 ここで荷電粒子がセンサーを通過した時における電気信号の強さを計算する。例として、大きい速度
1434 ($\beta\gamma \gg 1$) を持つ荷電粒子が、 $300 \mu\text{m}$ の厚さを持つシリコン半導体を通過した時の生成キャリア数及び
1435 信号強度を計算する。

1436 荷電粒子の速度が十分に早い場合、式 A.2 は物質の密度 ρ を用いて以下のように近似できる [45]。

$$-\left\langle \frac{dE}{dx} \right\rangle = 1 - 2\rho [\text{MeV/cm}] \quad (\text{A.5})$$

1437 シリコンの密度を 2.33 g/cm^3 、シリコンにおける 1 キャリア対生成に必要な平均電離エネルギー
1438 3.62 eV より、 $300 \mu\text{m}$ の厚さを持つシリコン半導体を通過した時の生成キャリア数は以下のように計算
1439 できる。

$$1 - 2 \times 2.33 \times (3 \times 10^4) \times 1/3.62 \simeq 20,000 - 39,000 \quad (\text{A.6})$$

1440 素電荷 $e = 1.6 \times 10^{-19} \text{ C}$ より、信号強度は以下のようになる。

$$20,000 - 39,000 \times 1.6 \times 10^{-19} \simeq (3.2 - 5.1) \times 10^{-15} [\text{C}] \quad (\text{A.7})$$

₁₄₄₁ 付録 B

₁₄₄₂ RD53A とフレキシブル基板

₁₄₄₃ RD53A の各 Front-end が持つアナログ回路を図 B.1 に示す。

₁₄₄₄ B.1 アナログ回路とデジタル回路

₁₄₄₅ B.2 試験用電荷入射の方法

₁₄₄₆ RD53A の各ピクセルが持つ試験用電荷入射回路の簡略図を図 B.3 に示す。

₁₄₄₇ B.3 RD53A のデータフォーマット

₁₄₄₈ RD53A のデータフォーマットの模式図を図 B.4 に示す。

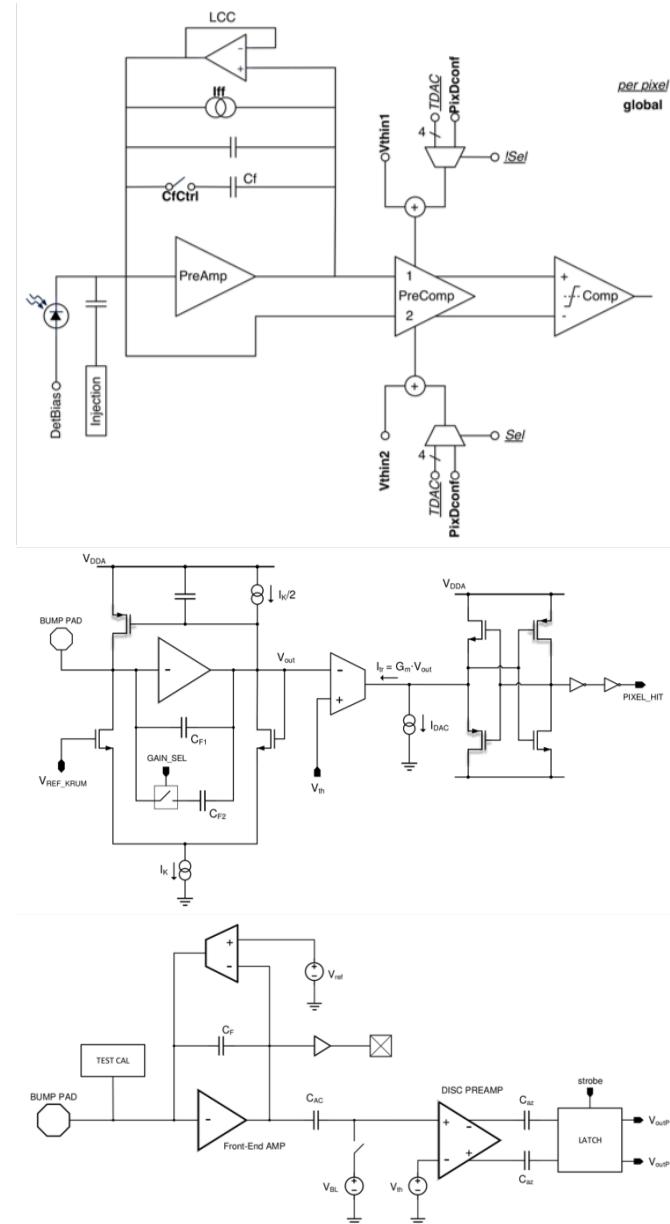


図 B.1 アナログ回路 [15]。上から順に Differential, Linear, Synchronous フロントエンドが持つアナログ回路図である。

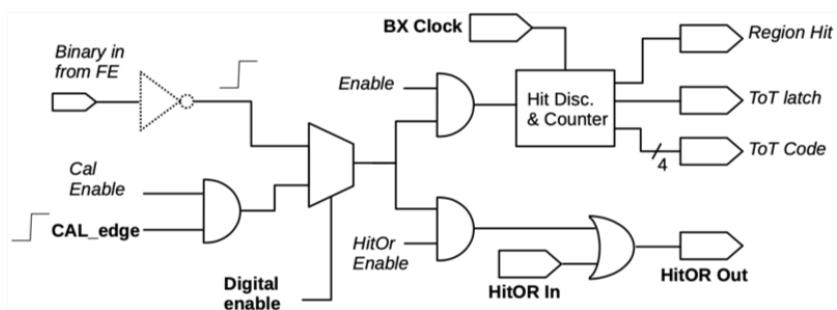


図 B.2 デジタル回路 [15]。

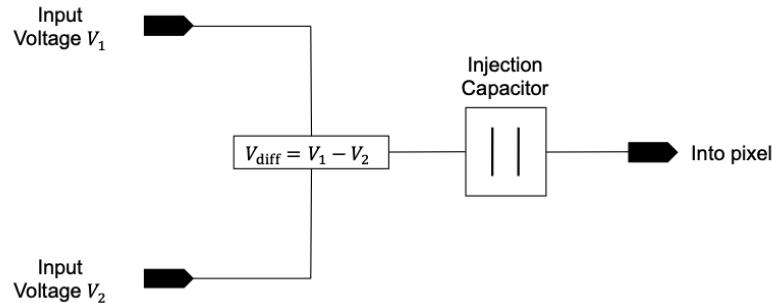


図 B.3 RD53A の各ピクセルが持つ試験用電荷入射回路の簡略図 [47]。図のように 2 つの電位を入力し、その差分の電圧を回路内のコンデンサにかける。これを開放することで、電荷をピクセル回路内に入力する。

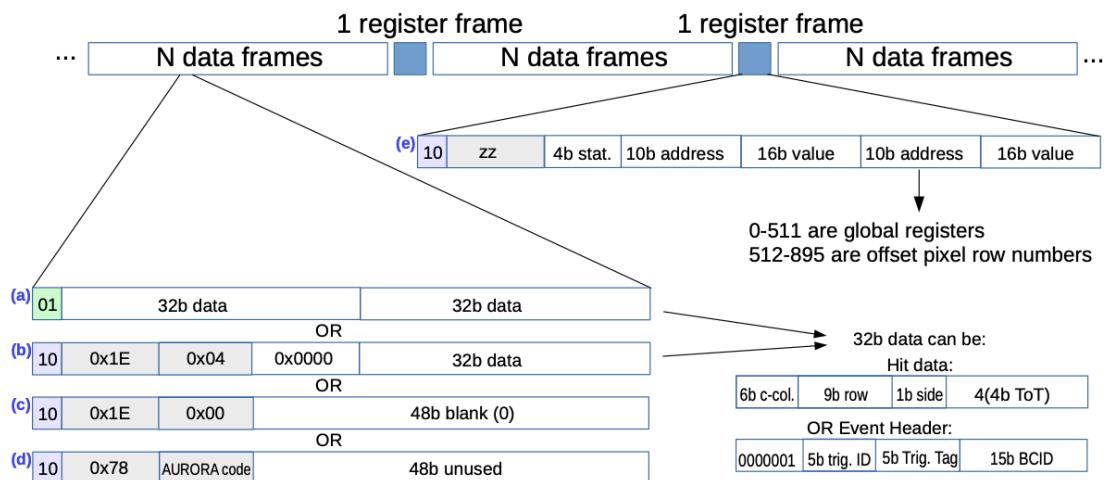


図 B.4 RD53A におけるデータフォーマットのイメージ [15]。RD53A のデータフォーマットは図のように、N 個の data frame と 1 個の register frame から成る。特にデータフレームに関しては header 部と hit data 部で構成され、header 部に trigger の情報が入っている。クロック信号や外部トリガーで送った情報はここに格納され、そのトリガーに応じてデータを取得する。

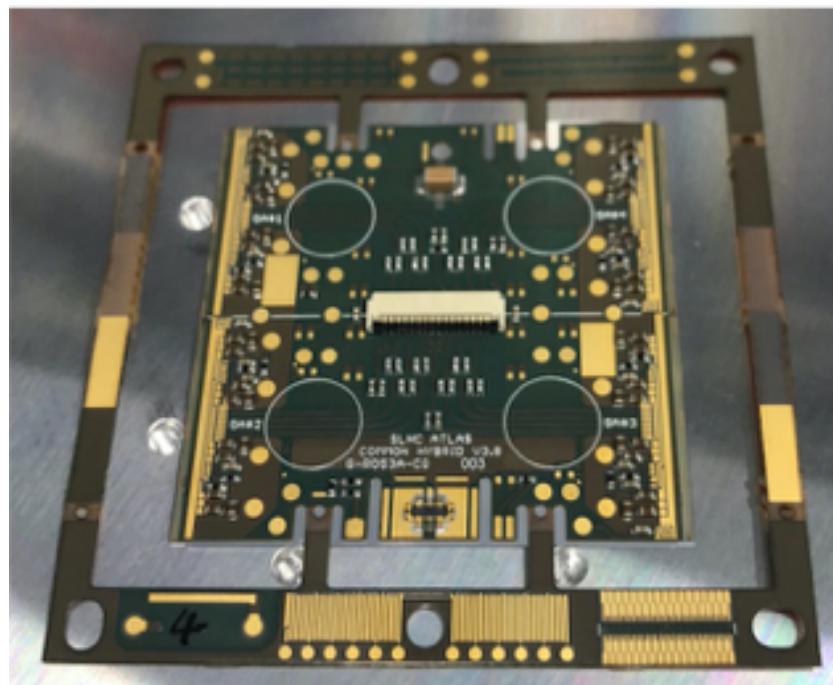


図 B.5 フレキシブル基板。

¹⁴⁴⁹ B.4 フレキシブル基板

¹⁴⁵⁰ フレキシブル基板の写真を図 B.5 に示す。

1451 付録 C

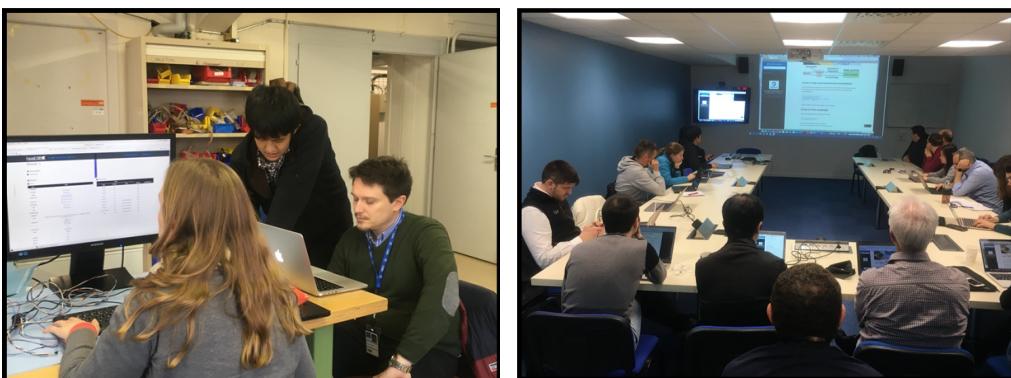
1452 ローカルデータベースのチュートリアル
1453 と普及状況

1454 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にてシステムのチュートリアルを行った。国際学会(ITk Upgrade week)が開かれていた期間である2020年2月3日から2月7日にかけて行い、世界的に各機関の研究者がCERNに集まっている機会を利用した。CERNだけでなく各国のローカルデータベースユーザに対して説明をすることができた。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。

- 1459 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)。
- 1460 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)。

1462 それぞれのセッションの様子を図C.1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。また品質試験の流れにおいて、一連の機能確認をすることができた。

1464 これを経て2020年11月現在、ローカルデータベースは世界18箇所にて導入され、試験運用が開始している。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



図C.1 ローカルデータベースシステムチュートリアルの様子。2020年2月にCERNでローカルデータベースシステムのチュートリアルを行った。参加者が実際にシステムの設置、機能実行を行うハンズオンセッション(左図)と、参加者の前で実際に機能の動作をみせ、議論を行うハンズオフセッション(右図)に分けて行った。システムに有益なフィードバックと共に、システムの機能普及に成功した。

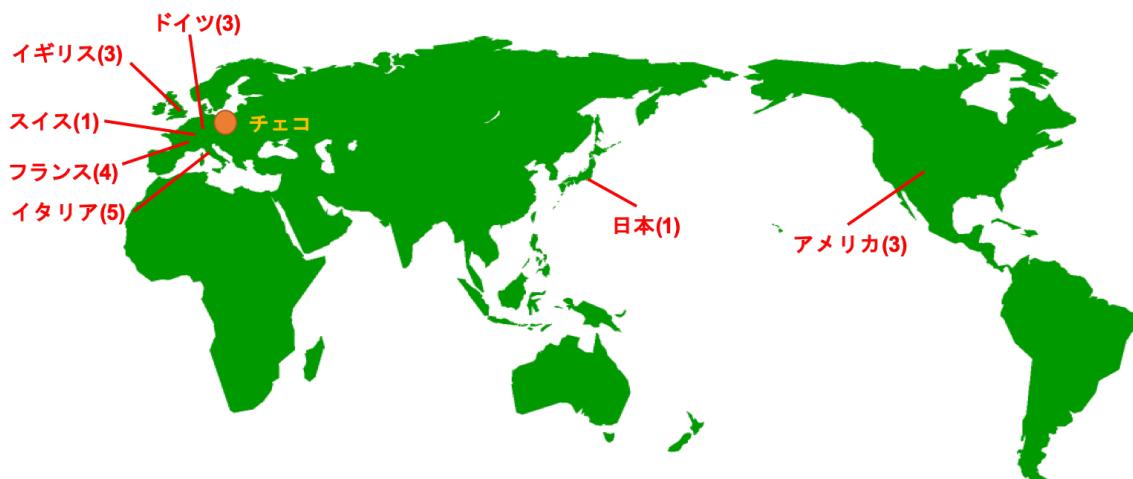


図 C.2 ローカルデータベースシステム導入及び試運転場所。赤文字は設置している地域、括弧内の数字はその地域におけるシステム導入場所の数を示している。2020 年 11 月現在、ローカルデータベースシステムは世界 18 の機関で試験運転がなされている。日本を除いてその多くはヨーロッパとアメリカに位置していることが分かる。

¹⁴⁶⁶ 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、導入及び試験運用を行っている機関を以下に示す。また世界地図を C.2 に示す。

- 高エネルギー加速器研究機構 (KEK), 日本
- 欧州原子核研究機構 (CERN), スイス
- University of Liverpool, イギリス
- University of Oxford, イギリス
- University of Glasgow, イギリス
- Paris-Saclay University, フランス
- パリ第 6 大学, フランス
- フランス国立科学研究中心, フランス
- University of Grenoble, フランス
- University of Gottingen, ドイツ
- University of Siegen, ドイツ
- University of Genoa, イタリア
- University of Salento, イタリア
- University of Milan, イタリア
- University of Udine, イタリア
- University of Trento, イタリア
- University of Oklahoma, アメリカ
- Argonne National Laboratory, アメリカ
- Lawrence Berkeley National Laboratory(LBL), アメリカ

付録 D

モジュール生産状況の解析

データベースシステムを使って、モジュール生産状況の解析を行うことができる。モジュールの組み立て工程は各組み立て機関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同期される。そのため全モジュールの現在工程を中央データベース上で取得できることができ、この情報を用いて世界的な生産状況の解析を行うことができる。現在は生産は行われていないが、想定している解析結果のイメージを図 D.1 に示す。

生産数や生産レートのモニタリングを行うことで、今後の生産計画や問題解決に役立てることが可能である。

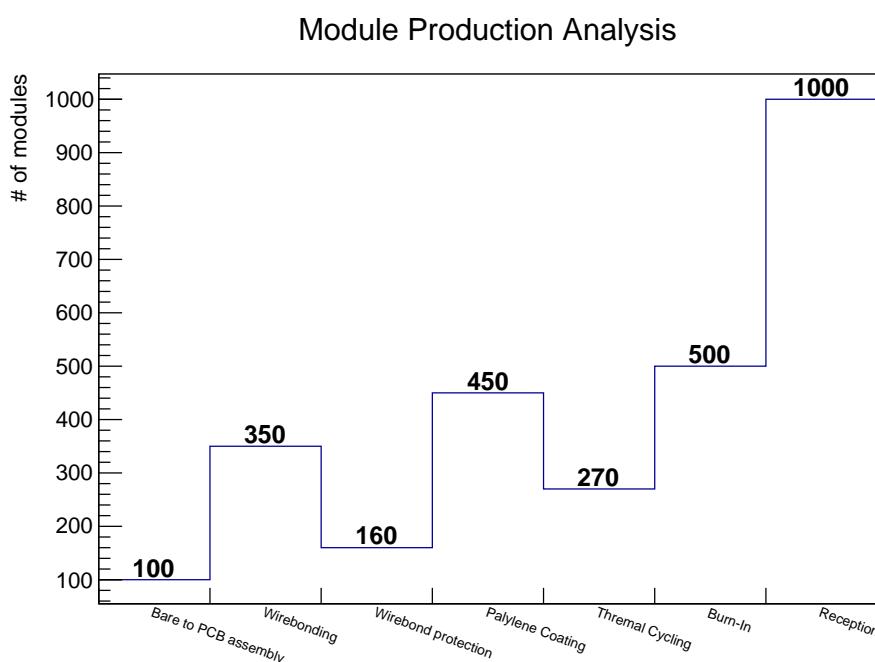


図 D.1 生産時のモジュール組み立て状況解析の例。ローカルデータベースにて組み立て工程は管理され、工程毎に中央データベースに同期されるため、生産時には全モジュールの現工程を中央データベースで取得できる。図の例のように各段階におけるモジュール数を見ることで生産状況の確認ができる。さらにある期間ごとに工程を取得することで生産レートなども計算することができ、今後の生産計画やモジュール選別の参考とすることができます。

付録 E

ファイル送信時におけるデータ容量と処理時間の考察

1499 始めに、処理速度遅延の原因として以下をあげる。

- 1500 ● サーバーの読み書き速度.
- 1501 ● サーバーのファイル転送アルゴリズム.
- 1502 ● サーバーのファイル転送時におけるパケットサイズ.
- 1503 ● サーバー間のネットワーク上の距離.
- 1504 ● サーバー間ネットワークの処理性能.

1505 KEK と LBNL に設置されているサーバーにおいて、中央データベースへのファイル送信処理時間に差
1506 が出る理由について考察する。

1507 ここで、ファイル送信時におけるデータ容量と処理時間の関係は、線形性を示さない。図 E.1 は KEK
1508 から LBNL のサーバーに scp コマンドを用いてファイル送信を行い、データ容量と処理時間の関係を取
1509 得したものである。赤線が線形フィットであるが、測定点は優位にずれている。これは TCP 通信におい
1510 てパケットの送信に輻輳制御 [48] と呼ばれる技術が使われており、データ送信量を変化させながら情報通
1511 信を行っているためである。

1512 scp によるファイル送信を以下の 2 つの場合に対して行い、ファイル容量と処理時間の関係を取得した。

- 1513 1. KEK から LBNL.
- 1514 2. LBNL から KEK.

1515 結果を図 E.2 に示す。ここで処理時間に差が生まれる原因について調査したものを以下に示す。

- 1516 ● 読み書き速度を測定したところ同程度であった。
- 1517 ● 輻輳制御アルゴリズムは Cubic をどちらも使用していた。
- 1518 ● パケットサイズは変わらなかった。
- 1519 ● ping による応答時間確認は同程度であった (111 msec)。

1520 よって各サーバが置かれているローカルネットワークの性能に差があると考えた。一般的には上りより
1521 下りの方が太いと考えると、KEK ローカルの上りネットワークの性能が、LBNL ローカルの上りネット
1522 ワークと比べて悪いと考えられる。これにより、ファイル送信時間に差が生まれている。

1523 次に、下の 2 つの場合に対して行い、ファイル容量と処理時間の関係を取得した。

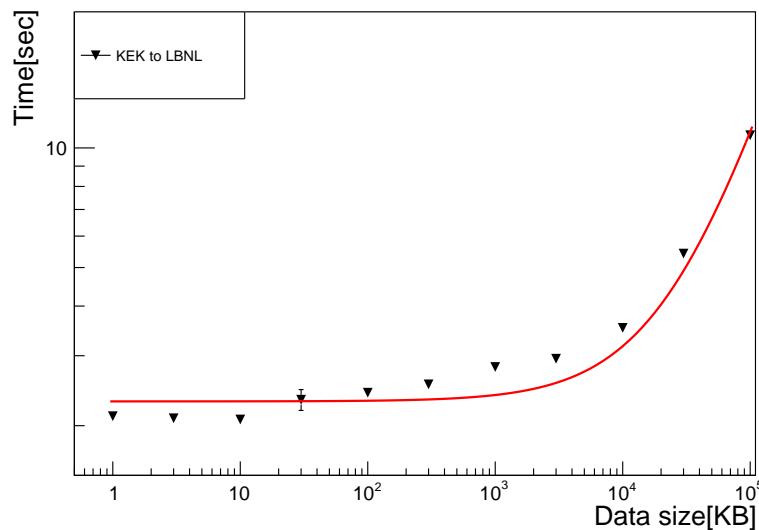


図 E.1 scp 送信におけるファイルサイズと処理時間の関係。横軸はファイルサイズ、縦軸は時間を示している。赤線は線形フィットを示している。測定点はフィット関数から優位に離れており、scp 送信においてファイルサイズと処理時間は線形性を示さないことが分かる。

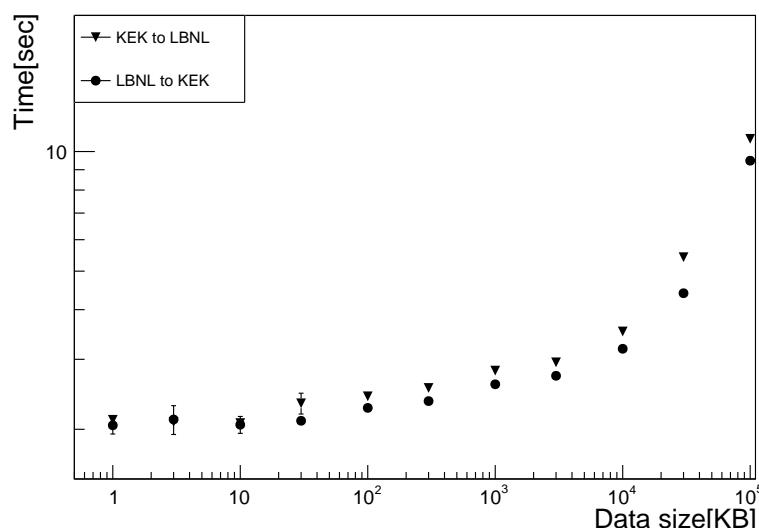


図 E.2 KEK、LBNL 間のファイル送信におけるファイルサイズと処理時間の関係。横軸はファイルサイズ、縦軸は時間を示している。三角は KEK から LBNL、丸は LBNL から KEK に送信した場合のものである。比較して KEK から LBNL へのファイル送信の方が時間がかかる。

1524 1. KEK から CERN(Lxplus).

1525 2. LBNL から CERN(Lxplus).

1526 結果を図 E.3 に示す。KEK ローカルの上りネットワークは細いため、処理時間に差が生まれる。加え
1527 てこの場合はサーバ間の距離による遅延も含まれていると考えられる。ping による応答時間が測定 1 で
1528 は 170 msec 程度なのに対し、測定 2 は 150 msec 程度であった。これも処理速度遅延に影響していると
1529 考えられる。

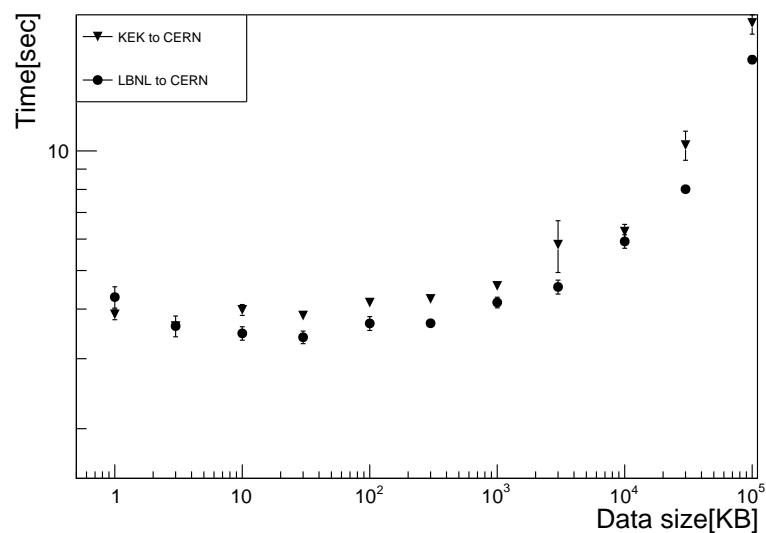


図 E.3 KEK、LBNL と CERN 間のファイル送信におけるファイルサイズと処理時間の関係。横軸はファイルサイズ、縦軸は時間を示している。三角は KEK から CERN、丸は LBNL から CERN に送信した場合のものである。比較して KEK から CERN へのファイル送信の方が時間がかかることがある。

1530 CERN と中央データベースが地理的に近い距離であることを考慮し、上述したことをまとめると KEK
 1531 と LBNL の間で処理時間の差が生まれる要因は以下であると考えた。

- 1532 ● ローカルネットワークの性能.
 1533 ● ネットワーク上の距離差.

参考文献

- 1534 [1] “現代素粒子物理実験的観点からみる標準理論”
 末包文彦・久世正弘・白井淳平・湯田春雄, 森北出版株式会社, 2016 年 12 月第 1 版第 1 刷発行.
- 1537 [2] “Quarks and Leptons”.
 Francis Halzen, Alan D.Martin. John Wiley & Sons Inc. 1984 年 9 月 1 日.
- 1539 [3] 藤井恵介. “自発的対称性の破れ”. 2000-2 <https://www-jlc.kek.jp/general/DOC/oho95-html/node6.html>, (2021-1)
- 1541 [4] ATLAS Collaboration. “Combined measurements of Higgs boson production and decay using up
 1542 to 80 fb^{-1} of proton–proton collision data at $\sqrt{s}= 13 \text{ TeV}$ collected with the ATLAS experiment”.
 1543 CERN Document Server. 2019-9 更新.
<http://cdsweb.cern.ch/record/2629412/>, (2021-1)
- 1544 [5] Stephen P. Martin. “A Supersymmetry Primer”. arXiv.org. 2016-1 更新.
<https://arxiv.org/abs/hep-ph/9709356>, (2021-1)
- 1547 [6] Damerau,H et al. “LHC Injectors Upgrade Technical Design Report”. CERN Document server.
 1548 2016-05
<https://cds.cern.ch/record/2153863>, (2020-12)
- 1550 [7] Georges Aad et al. “The ATLAS Experiment at the CERN Large Hadron Collider”. Semantic
 1551 Scholar. 2008-8
<https://www.semanticscholar.org/paper/The-ATLAS-Experiment-at-the-CERN-Large-Hadron-Aad-Gr>
 1553 7d771b20731969fe10c267465582ee60e9383db3, (2020-12)
- 1554 [8] ATLAS Collaboration. “The upgraded Pixel Detector of the ATLAS Experiment for Run 2 at
 1555 the Large Hadron Collider”. ScienceDirect. 2016-9
 1556 <https://doi.org/10.1016/j.nima.2016.05.018>, (2020-12)
- 1557 [9] L. Evans and P. Bryant. “LHC Machine”. Journal of Instrumentation. 2008-8.
 1558 <https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08001/pdf>, (2021-1)
- 1559 [10] Apollinari, G;Bjar Alonso, I; Brning, O; Lamont, M; Rossi, L. “High-Luminosity Large Hadron
 1560 Collider (HL-LHC) : Preliminary Design Report”. CERN Document Server. 2015-12
 1561 <https://cds.cern.ch/record/2116337>, (2020-12)
- 1562 [11] The HL-LHC project. “The HL-LHC project”. CERN Accelerating science. 2020-8
 1563 <https://hilumilhc.web.cern.ch/content/hl-lhc-project>, (2020-12)
- 1564 [12] ATLAS Collaboration. “Technical Design Report for the ATLAS Inner Tracker Pixel Detector”.
 1565 CERN Document Server. 2018-8
 1566 <https://cds.cern.ch/record/2285585>, (2020-12)

- 1567 [13] ATLAS Collaboration. “Projections for measurements of Higgs boson signal strengths and cou-
1568 pling parameters with the ATLAS detector at a HL-LHC”. CERN Document Server. 2020-5 更新
1569 新.
1570 [1571 \[14\] ATLAS Collaboration. “Observation and measurement of Higgs boson decays to \$WW\$ with the
1572 ATLAS detector”. CERN Document Server. 2020-6 更新.
1573 \[1574 \\[15\\] “Semiconductor Devices Physics and Technology”.
1575 S.M. Sze 著, 南日康夫・川辺光央・長谷川文夫訳, 産業図書, 2015 年 3 月第 2 版第 11 刷発行.
1576 \\[16\\] “Pixel Detectors”.
1577 Rossi, L., Fischer, P., Rohe, T., Wermes, N. Springer, 2006-7-8
1578 \\[17\\] “トータルドーズ効果”. 天文学辞典. 2018-3
1579 \\[1580 \\\[18\\\] Danilo Giugni. “General approach for thermal-mechanics QA and QC”. CERN Indico. 2019-12
1581 \\\[1582 \\\\[19\\\\] Johannes Weller. “Characterisation of Pixel Sensors for the ATLAS ITk Upgrade”. CERN
1583 Indico. 2020-7
1584 \\\\[1585 \\\\\[20\\\\\] Vasilije Perovic, CMS Tracker Group. “Serial powering in four-chip prototype RD53A modules
1586 for Phase 2 upgrade of the CMS pixel detector”. ScienceDirect. 2020-10.
1587 \\\\\[1588 \\\\\\[21\\\\\\] Timon Heim. “YARR - A PCIe based Readout Concept for Current and Future ATLAS Pixel
1589 Modules”. IOP Science. 2017
1590 \\\\\\[1591 \\\\\\\[22\\\\\\\] “物理数学 II”.
1592 西森秀穎 著, 丸善出版, 平成 27 年 9 月 30 日発行.
1593 \\\\\\\[23\\\\\\\] Meng, Lingxin. “RD53A Module Testing Document”. CERN Document server. 2020-9
1594 \\\\\\\[1595 \\\\\\\\[24\\\\\\\\] “Unicorn University - Unicorn College”. Unicorn University.
1596 \\\\\\\\[1597 \\\\\\\\\[25\\\\\\\\\] “MongoDB: The most popular database for modern apps”. MongoDB, Inc.
1598 \\\\\\\\\[1599 \\\\\\\\\\[26\\\\\\\\\\] “Welcome to Flask”. Flask Documentation.
1600 \\\\\\\\\\[1601 \\\\\\\\\\\[27\\\\\\\\\\\] 窪田ありさ. “HL-LHC ATLAS 実験用新型ピクセル検出器の系統評価と量産時に向けた試験管理シ
1602 ステムの開発” CERN Box. 2020-3
1603 \\\\\\\\\\\[1604 \\\\\\\\\\\\[28\\\\\\\\\\\\] “Databases and Collections”. MongoDB Manual.
1605 \\\\\\\\\\\\[1606 \\\\\\\\\\\\\[29\\\\\\\\\\\\\] “WiredTiger Storage Engine”. MongoDB, Inc.
1607\\\\\\\\\\\\]\\\\\\\\\\\\(https://docs.mongodb.com/manual/core/databases-and-collections/,\\\\\\\\\\\\(2020-12\\\\\\\\\\\\)\\\\\\\\\\\\)\\\\\\\\\\\]\\\\\\\\\\\(https://cernbox.cern.ch/index.php/s/BdhvSTAuuE5xHxt,\\\\\\\\\\\(2020-1\\\\\\\\\\\)\\\\\\\\\\\)\\\\\\\\\\]\\\\\\\\\\(https://flask.palletsprojects.com/en/1.1.x/,\\\\\\\\\\(2020-12\\\\\\\\\\)\\\\\\\\\\)\\\\\\\\\]\\\\\\\\\(https://www.mongodb.com/2,\\\\\\\\\(2020-12\\\\\\\\\)\\\\\\\\\)\\\\\\\\]\\\\\\\\(https://unicornuniversity.net/en/,\\\\\\\\(2020-1\\\\\\\\)\\\\\\\\)\\\\\\\]\\\\\\\(https://cds.cern.ch/record/2702738,\\\\\\\(2020-12\\\\\\\)\\\\\\\)\\\\\\]\\\\\\(https://iopscience.iop.org/article/10.1088/1742-6596/898/3/032053,\\\\\\(2020-12\\\\\\)\\\\\\)\\\\\]\\\\\(https://www.sciencedirect.com/science/article/pii/S0168900220308330,\\\\\(2021-1\\\\\)\\\\\)\\\\]\\\\(https://indico.cern.ch/event/935007/contributions/3928837/attachments/2075810/3485805/BA_pres_\\\\$jweller.pdf,\\\\(2020-1\\\\)\\\\)\\\]\\\(https://indico.cern.ch/event/860761/contributions/3661710/,\\\(2020-12\\\)\\\)\\]\\(https://astro-dic.jp/total-dose-effect/,\\(2020-12\\)\\)\]\(https://cds.cern.ch/record/1975394,\(2020-1\)\)](https://cds.cern.ch/record/1956710,(2020-1))

- 1608 <https://docs.mongodb.com/manual/core/wiredtiger/>,(2020-1)
- 1609 [30] “PyMongo 3.11.2 Documentation”. PyMongo 3.11.2 Documentation.
- 1610 <https://pymongo.readthedocs.io/en/stable/>,(2020-12)
- 1611 [31] “ROOT: analyzing petabytes of data, scientifically.”. ROOT Team.
- 1612 <https://root.cern/>,(2020-12)
- 1613 [32] “QC Software doc”. ITk Docs.
- 1614 https://itk.docs.cern.ch/pixels/qc__software/rd53a__demo__flow/,(2020-1)
- 1615 [33] “InfluxDB: Purpose-Built Open Source Time Series Database”. influxdata.
- 1616 <https://www.influxdata.com>,(2020-12)
- 1617 [34] “Grafana: The open observability platform”. Grafana Labs.
- 1618 <https://grafana.com>,(2020-12)
- 1619 [35] “PySerialComm”. GitLab.
- 1620 <https://gitlab.cern.ch/solans/PySerialComm>,(2020-12)
- 1621 [36] “RD53A Single Chip Card Configuration”. CERN twiki. 2018-5-14
- 1622 https://twiki.cern.ch/twiki/pub/RD53/RD53ATesting/RD53A__SCC__Configuration.pdf,(2020-12)
- 1624 [37] Renesas Electronics Corporation. “GPIO”. RENESAS.
- 1625 <https://www.renesas.com/jp/ja/support/engineer-school/mcu-programming-peripherals-01-gpio-12>
- 1627 [38] Microchip Technology. “2.7V Dual Channel 10-Bit A/D Converter with SPITM Serial Interface”. 秋月電子通商. 2006-08
- 1629 <https://akizukidensi.com/download/ds/microchip/mcp3002.pdf>,(2020-12)
- 1630 [39] “Raspberry Pi 3 Model B+”. RASPBERRY PI FOUNDATION.
- 1631 <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>,(2020-12)
- 1632 [40] “E3640A E3649A Programmable DC Power Supplies - Data Sheet”. Keysight Technologies. 2018-3
- 1634 <https://www.keysight.com/jp/ja/assets/7018-06827/data-sheets/5968-7355.pdf>,(2020-12)
- 1636 [41] “XpressK7-160-Gen2”. Mouser Electronics.
- 1637 <https://www.mouser.jp/ProductDetail/ReFLEX-CES/XpressK7-160-Gen2?qs=rrS6PyfT74eSJLUPLu1P5g%3D%3D>,(2020-12-22)
- 1639 [42] Wielers, Monika. “Pixel production database serial numbering scheme”. CERN Document Server. 2020-12-07
- 1641 <https://cds.cern.ch/record/2728364>,(2020-12)
- 1642 [43] “Berkeley Lab — Lawrence Berkeley National Laboratory”. Berkeley Lab.
- 1643 <https://www.lbl.gov>,(2020-1)
- 1644 [44] “LocalDB docs”. LocalDB docs.
- 1645 <https://localdb-docs.readthedocs.io/en/top/>,(2020-12)
- 1646 [45] “Selected Exercises in Particle and Nuclear Physics”.
- 1647 Lorenzo Bianchini, Springer, 2017-9 published.
- 1648 [46] Garcia-Sciveres, Maurice. “The RD53A Integrated Circuit”. CERN Document Server. 2017-10

- 1649 <https://cds.cern.ch/record/2287593>, (2020-12)
- 1650 [47] Patrick McCormack, Maurice Garcia-Sciveres, Timon Heim, Benjamin Nachman, Magne
1651 Lauritzen. “New Method for Silicon Sensor Charge Calibration Using Compton Scattering”.
1652 arXiv.org. 2020-8.
- 1653 <https://arxiv.org/abs/2008.11860>, (2020-1)
- 1654 [48] “TCP 技術入門-進化を続ける基本プロトコル”.
1655 安永遼真, 中山悠, 丸田一輝著, WEB+DB PRESS plus, 2019 年 7 月 6 日発売.

1656 謝辭

図目次

1657

1658	1.1	ヒッグスボテンシャル	2
1659	1.2	標準模型の素粒子の質量とヒッグス粒子との結合定数の関係	3
1660	1.3	加速器の全体像	5
1661	1.4	極角 θ と擬ラピディティ η の関係図	6
1662	1.5	ATLAS 検出器の全体像	7
1663	1.6	内部飛跡検出器の全体像	8
1664	1.7	ピクセル検出器の全体像	9
1665	1.8	ピクセルモジュールの全体像	9
1666	1.9	HL-LHC 運転計画	10
1667	1.10	LHC、HL-LHC における粒子の質量とヒッグス粒子の結合定数の関係と統計誤差の見積 もり	11
1668	1.11	パイルアップ数増加のイメージ	12
1669	1.12	ITk の全体像	12
1670	1.13	ITk の断面図	13
1671	1.14	VBF イベントの図	14
1672	1.15	VBF イベントにおけるジェットの運動方向の η 分布	15
1673	2.1	ピクセルモジュールの構成	16
1674	2.2	新型ピクセルモジュールに搭載するシリコンセンサーの断面図	17
1675	2.3	ToT の概念図	18
1676	2.4	RD53A	18
1677	2.5	ピクセルモジュールにおける信号伝達の様子	19
1678	2.6	搭載するモジュールのプロトタイプと ITk における配置。	20
1679	3.1	組み立て工程のイメージ図	22
1680	3.2	外観検査の様子	23
1681	3.3	平坦性測定の様子。	23
1682	3.4	センサー電流-電圧特性結果の例	24
1683	3.5	FE チップ電流-電圧特性試験結果の例。	24
1684	3.6	読み出し試験の様子	25
1685	3.7	デジタル回路読み出しにおける <i>Occupancy</i> 分布の例。	28
1686	3.8	入射電荷量と <i>Occupancy</i> の関係	29
1687	3.9	Threshold 値とノイズの分布。	29

1689	3.10	ノイズ占有率測定における <i>NoiseOccupancy</i> 分布の例。	31
1690	3.11	組み立て工程と対応する品質試験一覧	33
1691	4.1	データベースシステムの模式図	34
1692	4.2	ローカルデータベースシステムの概要	36
1693	4.3	MongoDB の構造の例	38
1694	4.4	先行研究により設計された読み出し試験結果の MongoDB 内部構造	39
1695	4.5	ウェブアプリケーション処理のイメージ	40
1696	4.6	品質試験結果ページの例	41
1697	4.7	中央データベースにおけるモジュールの種類と構造	42
1698	4.8	中央データベース内におけるモジュール構造の一例 (Quad モジュール)	43
1699	4.9	同期ツール処理のイメージ	44
1700	4.10	同期機能の概要	44
1701	4.11	ウェブアプリケーションにおけるコメント機能	46
1702	4.12	ウェブアプリケーションにおけるタグ機能	46
1703	4.13	結果選択画面及び組み立て工程表示の例	50
1704	4.14	ピクセル解析ツールにおけるファイル統合処理のイメージ	51
1705	4.15	Tree ファイルとそのデータ保持のイメージ	52
1706	4.16	ピクセル解析ツールの処理の流れ	53
1707	4.17	ウェブアプリケーションにおける検索機能の様子	54
1708	4.18	各モジュールにおけるデータベース操作の流れ	55
1709	5.1	読み出し試験に用いるソフトウェアの概要	58
1710	5.2	RD53A シングルモジュール (SCC)	59
1711	5.3	モジュール付属サーミスタを用いた温度読み出し回路	60
1712	5.4	用いた電源	60
1713	5.5	使用した FPGA ボード (XpressK7)	61
1714	5.6	使用した FMC-DisplayPort 変換カード	61
1715	5.7	ハードウェアセットアップ	61
1716	5.8	ダウンロードしたモジュール情報確認画面	63
1717	5.9	DCS 情報のモニタリングの様子	64
1718	5.10	検索機能確認の様子	65
1719	5.11	試験結果の閲覧	66
1720	5.12	読み出し試験結果の選択ページ	68
1721	5.13	ピクセル解析結果	69
1722	5.14	各評価基準における不良ピクセルの分布	70
1723	5.15	中央データベースにおける読み出し試験結果ページ	71
1724	6.1	方法 1:Python リストを用いた場合	73
1725	6.2	方法 1 における問題点のイメージ	74
1726	6.3	処理時間のボトルネックとなっているデータ構造	75
1727	6.4	方法 1 における検索処理速度測定結果	75

1728	6.5	方法 2:検索キーワード専用コレクションを用いた場合	76
1729	6.6	方法 2 における検索処理時間測定結果	79
1730	6.7	方法 2 における検索情報コレクションの生成時間測定結果	80
1731	6.8	方法 2 の検索における詳細処理	81
1732	6.9	方法 2 における詳細処理時間の測定結果	82
1733	6.10	ドキュメント数に対する型変換処理時間の関係	83
1734	6.11	方法 3:検索情報と共に一覧表示に必要な情報を保持、参照	84
1735	6.12	方法 4:検索情報コレクションを分散、マルチスレッドを使用	84
1736	6.13	方法 3、4 に対する処理時間測定結果	85
1737	7.1	各サーバーの設置場所	87
1738	7.2	”createTestRunAttachment” を用いた添付処理におけるファイル容量と処理時間の関係	88
1739	7.3	中央データベースウェブページへの通信時間の測定結果	89
1740	7.4	ダウンロード処理における流れのイメージ図	91
1741	7.5	登録した Quad モジュールと構成部品のシリアルナンバー一覧。	92
1742	7.6	登録した Quad モジュールのダウンロードの様子	93
1743	7.7	モジュール及び構成部品情報取得のイメージ図	95
1744	7.8	中央データベースにおける読み出し試験結果の構造	98
1745	7.9	アップロード処理に関する流れのイメージ図	99
1746	7.10	アップロード機能における詳細処理測定結果	100
1747	8.1	中央データベースにおける選択可能な組み立て工程のイメージ	108
1748	A.1	真性半導体中のシリコン	109
1749	A.2	pn 接合の電流 – 電圧特性	110
1750	A.3	空乏層	110
1751	B.1	アナログフロントエンド	113
1752	B.2	デジタルフロントエンド	113
1753	B.3	RD53A の各ピクセルが持つ試験用電荷入射回路の簡略図	114
1754	B.4	RD53A におけるデータフォーマットのイメージ図	114
1755	B.5	フレキシブル基板	115
1756	C.1	ローカルデータベースシステムチュートリアルの様子	116
1757	C.2	ローカルデータベースシステム導入及び試運転場所	117
1758	D.1	生産時のモジュール組み立て状況解析の例	118
1759	E.1	添付するファイルサイズと処理時間の関係	120
1760	E.2	KEK、LBL 間のファイル送信	120
1761	E.3	KEK、LBL と CERN 間のファイル送信	121

表目次

1762	表目次	
1763	1.1 標準模型のフェルミオン	2
1764	1.2 標準模型のボソン	2
1765	1.3 現行 LHC と HL-LHC の比較	10
1766	1.4 ピクセル検出器設置領域の比較	12
1767	1.5 搭載するピクセルモジュール数の比較	13
1768	1.6 VBF $H \rightarrow WW$ の測定における系統誤差の見積もり	14
1769	2.1 センサーのスペック	17
1770	2.2 RD53A のスペック	18
1771	3.1 デジタル回路読み出しにおける結果ファイル一覧	28
1772	3.2 Threshold 測定における結果ファイル一覧	30
1773	3.3 ToT 測定における結果ファイル一覧	30
1774	3.4 ノイズ占有率測定における結果ファイル一覧	30
1775	3.5 ピクセル解析の評価基準一覧	32
1776	4.1 品質試験に用いる主なコレクション	39
1777	4.2 中央データベースにおける組み立て工程と付随するテスト項目	43
1778	4.3 ローカルデータベースユーザ権限及び使用機能一覧	46
1779	5.1 温度読み出しシステムに使用した装置一覧	60
1780	5.2 読み出しに使用した PC の性能	61
1781	6.1 測定に使用したノート PC の性能	78
1782	6.2 検索機能処理時間測定の詳細	78
1783	6.3 処理 3,5 における型変換処理 6.5 の割合	82
1784	7.1 各ローカルデータベースサーバーの性能一覧	86
1785	7.2 同期ツールの中で使用する中央データベースの主な API 一覧	87
1786	7.3 中央データベース API 実行時の処理時間測定結果。	88
1787	7.4 ダウンロード機能を用いて保存する情報一覧。	90
1788	7.5 登録したモジュールのダウンロード処理時間測定結果	94
1789	7.6 ダウンロード機能における詳細処理にかかる時間測定	95
1790	7.7 中央データベースにおける読み出し試験結果に関する情報一覧	98

1791	7.8	アップロード処理における結果、設定値ファイル添付実行結果と処理時間	101
1792	7.8	アップロード処理における結果、設定値ファイル添付実行結果と処理時間	102
1793	7.9	アップロード処理改善後における結果、設定値ファイル添付実行結果と処理時間	103