

修士論文

KamLAND実験における
データストレージシステム及び
アナリシスフレームワークの開発研究

東北大学大学院理学研究科
物理学専攻

高木 大輔

平成10年

概要

本研究では、KamLAND 実験で得られるイベントデータを記録し、管理するためのシステムを開発した。

物理実験で得られたデータを記録するシステムは、単にデータが記録できればいいのではなく、データ解析時の利便性を考慮に入れたシステムでなければならない。

そのため、膨大な量のファイルをシステムが管理し、ユーザーが必要なイベントに素早くアクセスできるようにした。

また、物理実験では、測定器のアップデートにより、イベントデータの構造が変更される可能性がある。また、ユーザーによる解析結果も記録し、イベントデータと同様にアクセスできるようにするためには、柔軟なデータ構造を扱える必要がある。

そこで、今回作成したデータストレージシステムに、オブジェクト指向データベースの手法を用いることで、この問題を解決した。

次に、ユーザーの解析プログラムとデータストレージをつなぐフレームワークを開発した。解析を行なうユーザーは、フレームワークを通じてデータを取得し、必要な解析を行なう。また、それぞれの解析プログラムに共通のインターフェースを持たせることにより、解析プログラム同士を自由に組み合わせることができるようにした。

物理解析では、ある段階の解析処理が、その前段階の解析結果に依存することが良くある。そこで、各段階の解析処理を部品のように扱うことにより、他の部分に影響を与えずに特定の部分の解析手法を切替えることができる。

また、共通のインターフェースを用いることで、他のユーザーが開発した解析手法を自分の解析処理に簡単に組み込むことができる。また、自分が開発した解析手法を他のユーザーに簡単に利用してもらうことができる。

そして、このような解析部品が充実すれば、ユーザーは既存の部品を組み合わせるだけで解析を行なうことができる。もし、必要な部品が無いときは、その部分だけを実装すればよく、一度実装した部品は他のユーザーが簡単に利用することができる。

最後に、開発したデータストレージシステムのテストベンチでの性能評価を行なった。

目次

第1章 KamLAND 実験	1
1.1 物理的背景	1
1.1.1 原子炉反ニュートリノ	2
1.1.2 地球反ニュートリノ	8
1.1.3 太陽ニュートリノ	10
1.2 検出器	11
1.3 反ニュートリノ検出原理	14
1.3.1 遅延同時計数法	14
1.4 バックグラウンド	15
1.5 テストベンチ	16
1.5.1 目的	16
1.5.2 検出器	17
第2章 データストレージシステム	18
2.1 要請	18
2.2 設計方針	20
2.3 問題点	24
2.4 ROOT System	25
2.5 システムの概要	25
2.6 実装	27
2.7 インターフェース	27
2.8 ストレージ	28
2.8.1 セクションストレージ	28
2.8.2 ランストレージ	28
2.8.3 データストレージ	28
2.9 データベース	32
第3章 解析フレームワーク	33
3.1 概要	33
3.2 設計および実装	34
3.3 アナライザ	35
3.4 セレクタ	36
3.5 複合アナライザ	37
3.6 解析マネージャ	38

第4章 性能評価	39
4.0.1 結果	40
第5章 まとめ	41

目 次

1.1	日本国内に存在する原子力発電所の分布	4
1.2	KamLAND で検出される原子炉反ニュートリノのエネルギースペクトル	5
1.3	ニュートリノ振動パラメータの探索領域と、太陽ニュートリノ欠損、大気 ニュートリノ異常に対する振動解	7
1.4		9
1.5	標準太陽模型から予想されるニュートリノのスペクトル	10
1.6	KamLAND 実験施設の全体図	12
1.7	KamLAND の概要	13
1.8	反ニュートリノによる信号	14
1.9	テストベンチ	17
2.1	イベント構造	20
2.2	ディレクトリ構造	22
2.3	データストレージシステム	23
2.4	セマンティクスギャップ	24
2.5	システムの概要図	25
2.6	ラン - イベント - セクションによるインターフェース	27
2.7	ストレージ	29
2.8	セクションの書き込み	30
2.9	セクションの読み込み	31
2.10	データベース	32
3.1	アナライザの概念	33
3.2	アナライザの結合図	34
3.3	アナライザ	35
3.4	セクタ	36
3.5	アナライザやセクタの接続	36
3.6	複合アナライザ	37
3.7	解析マネージャ	38
4.1	テストベンチで用いるデータ構造	39

表 目 次

1.1	核分裂1回当たりに解放されるエネルギー	3
1.2	原子力発電所の熱出力と距離	3
1.3	原子炉反ニュートリノ検出器の性能	6
1.4	地球表面から 100km 以内に存在する主な放射性同位体からのフラックス	8
4.1	データストレージのデータ入出力性能	40

第1章 KamLAND実験

1.1 物理的背景

素粒子物理学、宇宙物理学の重要な研究課題の一つに、ニュートリノが質量を持つかどうかという問題がある。

ニュートリノの質量は実験的に非常に小さく、これまでゼロであると信じられてきた。しかし、光子のようなゲージ場の量子の場合と違って、質量がゼロでなければならない明確な理論的根拠はない。

近年、太陽ニュートリノ欠損や大気ニュートリノ異常など、検出されたニュートリノの数が、予想される数より有意に少ないという実験事実が報告された。

これらの現象を説明するものとして、ニュートリノ振動が考えられている。ニュートリノ振動とは、3種類のフレーバーニュートリノ (ν_e, ν_μ, ν_τ) が異なった質量を持つことによって、各種類のニュートリノの間に遷移が起きる現象である。

例えば、真空中で ν_e から ν_μ へ遷移する確率は、2つのニュートリノの質量の2乗差 ($\Delta m^2 = m_\mu^2 - m_e^2$) と、混合角 ($\sin^2\theta$) に依存し、

$$P(\nu_e \rightarrow \nu_\mu) = \sin^2 2\theta \sin^2 \left(\frac{1.27 \Delta m^2 L}{E} \right) \quad (1.1)$$

と表される。ここで、 L はニュートリノが生成されてから検出されるまでに走行した距離 (m)、 E はニュートリノのエネルギー (MeV) である。

ν_e のみに感度をもつ検出器では、 ν_μ, ν_τ を検出することができない。そのため、 ν_e が検出器に到達する前に他のニュートリノに遷移した場合、検出される ν_e ニュートリノの数は、遷移が起らないと予想した場合に比べて減少する。

もし、太陽ニュートリノ欠損や、大気ニュートリノ異常がニュートリノ振動によるものであるならば、ニュートリノが固有の質量を持つことになる。ニュートリノが固有の質量を持つかどうかは、素粒子物理学のみならず、宇宙物理学にも重要な影響を与える。そのため、ニュートリノの質量検出は各国の研究者が総力をあげて取り組んでいる重要な研究課題となっている。

超新星 1987A からのニュートリノバーストの検出成功や、太陽ニュートリノの検出成功など、素粒子物理学、宇宙物理学の研究発展に貢献してきた神岡実験は、1996年から稼働を開始したスーパー神岡実験に引き継がれ、その役目を終えることになった。そこで、神岡実験の検出器が設置されていた場所に、新たに大容量の液体シンチレーターを用いた検出器を建設し、超低エネルギーニュートリノの検出を目的とした実験を行うことになった。この実験を KamLAND (Kamioka Liquid scintillator Anti-Neutrino Detector) 実験と呼ぶ。

KamLAND 実験では数 100keV 領域までの低エネルギーニュートリノを検出することができる。また、液体シンチレータを用いることにより、1.8MeV 以上のエネルギー領域において、反ニュートリノ反応を識別することができる。

このエネルギー領域には、原子炉反ニュートリノ、地球反ニュートリノ、 ${}^7\text{B}$ 太陽ニュートリノなど、多くの興味深いニュートリノが存在している。

以下に KamLAND 実験で期待される物理について述べる。

1.1.1 原子炉反ニュートリノ

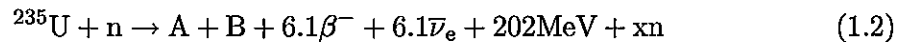
原子炉内では、燃料となるウラン、プルトニウムの核分裂により、反ニュートリノが生成される。この反応で生成される反ニュートリノは、ほぼ 100% の確立で電子型反ニュートリノとなる。また、原子炉の総熱出力量により、数% の精度で反ニュートリノのフラックスを知ることができる。

このように、他のフレーバーの混入のない反ニュートリノを定量的に放出するため、原子炉はニュートリノ振動を検証するためのすぐれた反ニュートリノ源となる。

そこで、原子炉からの反ニュートリノを検出することにより、世界最高感度でニュートリノ振動現象を検出する。

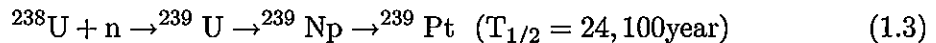
原子炉の燃料は、最初、2~4% の ${}^{235}\text{U}$ と残りを占める ${}^{238}\text{U}$ からなる。

燃料内の ${}^{235}\text{U}$ は、熱中性子を吸収して核分裂を起こし、核分裂によって生じた娘核がベータ崩壊を繰り返す際に、その際に平均 6.1 個の電子型反ニュートリノを生成する。

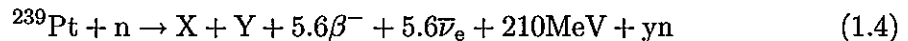


一方、 ${}^{238}\text{U}$ は、0.8MeV 以上の高速中性子に対して核分裂を起こすことが可能である。

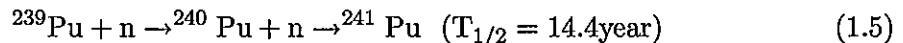
また、 ${}^{238}\text{U}$ は熱中性子を吸収後にベータ崩壊を 2 回起こし、核分裂核種 ${}^{239}\text{Pu}$ を生成する。



生成された ${}^{239}\text{Pu}$ は、 ${}^{235}\text{U}$ と同様に熱中性子により核分裂を起こし、平均 5.6 個の電子型反ニュートリノを生成する。



${}^{239}\text{Pu}$ は熱中性子を連続して吸収することにより、 ${}^{241}\text{Pu}$ を生成することも可能である。



${}^{241}\text{Pu}$ も、熱中性子を吸収して核分裂を起こすことが可能である。

核分裂1回当たりに解放されるエネルギーをまとめると表1.1のようになる。この結果、原子炉の熱出力から生成された反ニュートリノの数を見積もることができる。

Isotope	Released Energy [MeV/fission]
^{235}U	201.8
^{238}U	205.0
^{239}Pu	210.3
^{241}Pu	212.6

表 1.1: 核分裂1回当たりに解放されるエネルギー

図1.1に、日本国内に存在する原子力発電所の分布図を示す。これらの原子力発電所について、熱出力と距離についてまとめると、表1.2のようになる。

Reactor Site	Power [GW]	Distance [km]	Power Flux [GW/cm ²]
柏崎	24.5	160	7.7×10^{-15}
大飯	13.7	190	2.9×10^{-15}
浜岡	10.6	210	1.9×10^{-15}
高浜	10.2	210	1.8×10^{-15}
敦賀	4.5	150	1.5×10^{-15}
美浜	4.9	165	1.4×10^{-15}
志賀	1.9	97	1.3×10^{-15}
福島第一	14.2	357	8.9×10^{-16}
福島第二	13.2	351	8.5×10^{-16}
東海	3.3	306	2.8×10^{-16}

表 1.2: 原子力発電所の熱出力と距離

表の右端の熱出力フラックスとは、原子炉から見た KamLAND の立体角に熱出力をかけたもので、KamLAND までの距離を L とすると、 $\text{GW}/(4\pi L^2)$ で表される量である。

1995 年度の全国の原子炉の運転効率 77.21% を考慮すると、KamLAND での熱出力フラックスは、 $1.5 \times 10^{-14} [\text{Gw}/\text{cm}^2]$ となる。

したがって、KamLAND での全反ニュートリノフラックスは、

$$\frac{6 \times 10^9 \times 1.5 \times 10^{-14}}{1.6 \times 10^{-19} \times 200 \times 10^6} = 2.9 \times 10^6 \quad [/\text{cm}^2/\text{s}] \quad (1.6)$$

となる。

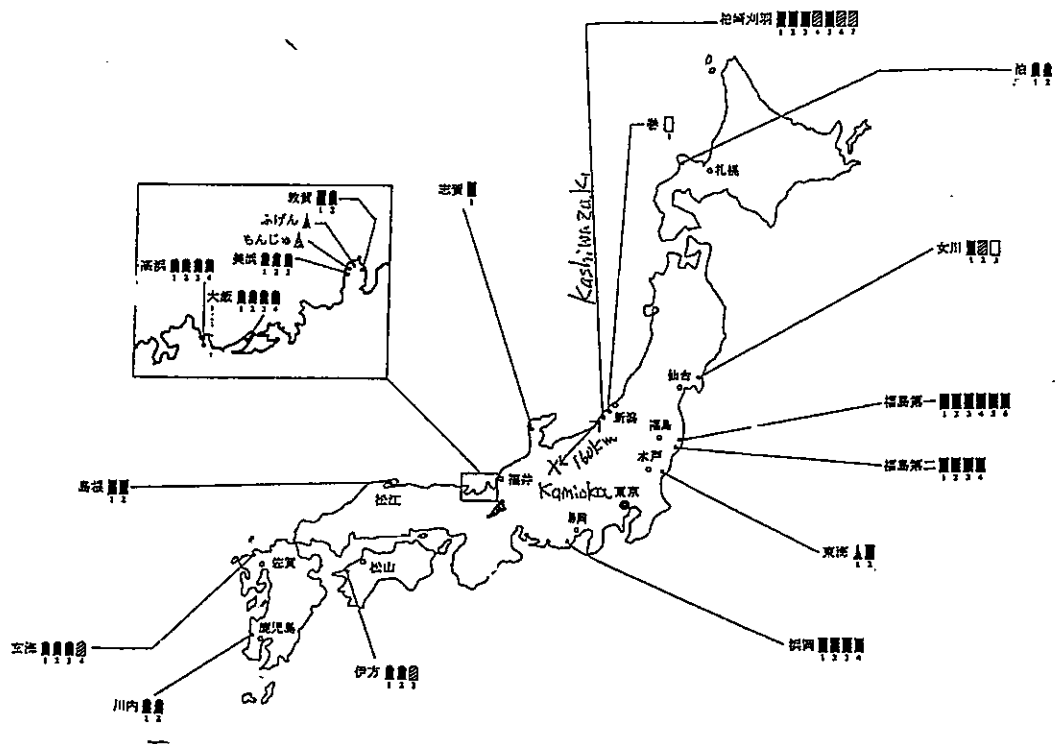


図 1.1: 日本国内に存在する原子力発電所の分布

この結果から、KamLANDで検出される事象数を見積もる。

KamLANDでは、液体シンチレータ内の陽子と反ニュートリノとの逆ベータ崩壊に起因する信号を観測して、反ニュートリノを検出する。

逆ベータ崩壊の反応断面積は、

$$\sigma = 9.6 \times 10^{-44} \times (E_\nu - 1.3) \sqrt{(E_\nu - 1.3)^2 - m_e^2} \text{ [cm}^2\text{]} \quad (1.7)$$

で表される。

反応の頻度は、反ニュートリノフラックス、検出器内の陽子数に比例する。

$$N = \sigma \times F_\nu \times N_p \quad (1.8)$$

また、検出器内の陽子数は液体シンチレータの水素 / 炭素比 ($r_{H/C}$) によって変化し、次の式で表される。

$$N_p/w = 6.02 \times 10^{23} / (1.00794 + 12.011/r_{H/C}) \quad (1.9)$$

ここで、 w は液体シンチレータの重さであり、 $r_{H/C} = 1.9$ のとき、 $N_p/w = 8.2 \times 10^{31} / \text{kt}$ になる。KamLANDでの有効質量を500トンとしたとき、KamLANDで検出される原子炉ニュートリノのエネルギースペクトルを図1.2に表す。これを積分すると、1年間で検出されるイベント数は、約390イベントとなる。

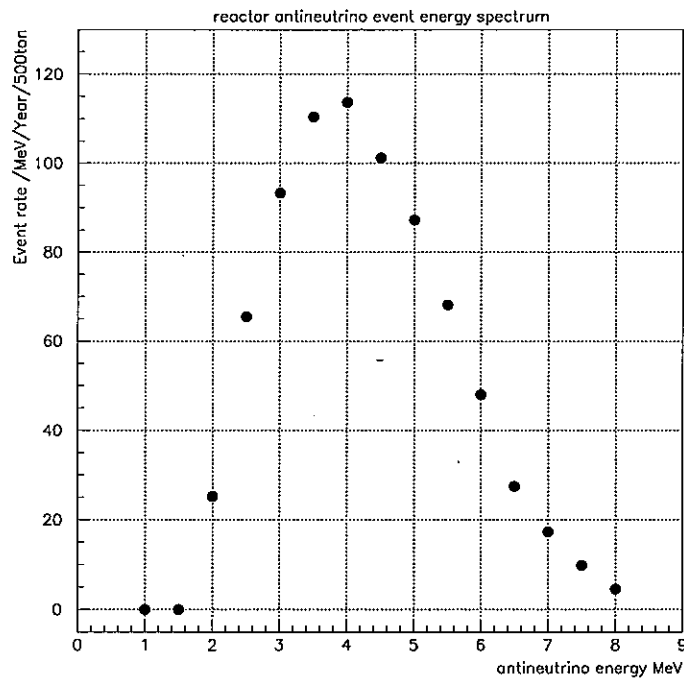


図 1.2: KamLANDで検出される原子炉反ニュートリノのエネルギースペクトル

原子炉反ニュートリノのニュートリノ振動検出による、 Δm^2 の感度を見積もる。式1.1によると、混合角 $\sin^2\theta$ が与えられた時、ニュートリノ振動は以下の時最大になる。

$$\Delta m^2 = \frac{1}{1.27} \times \frac{\pi}{2} \times \frac{E[\text{Mev}]}{L[\text{m}]} \quad (1.10)$$

よって、 Δm^2 の検出能は、ニュートリノの飛行距離とニュートリノのエネルギーによって制限される。より小さい Δm^2 を検出するには、ニュートリノの飛距離を長くするか、エネルギーの小さいニュートリノを検出しなければならない。

KamLAND 実験では、 $L \sim 150 \text{ km}$ 、 $E \sim \text{数 MeV}$ であるため、 Δm^2 の感度は、 $\sim 10^{-5}$ となる。表1.3に、世界の原子炉ニュートリノ検出器の性能を示す。

Experiment	Target Mass	Reactor Distance	$\Delta m^2(\text{eV}^2)$
Bugey	1,200 kg	15, 40, 95 m	2×10^{-2}
Chooze	4,800 kg	1025 m	1×10^{-1}
Palo Verde	12 ton	740, 850 m	$\sim 10^{-3}$
KamLAND	1,000 ton	$\sim 150 \text{ km}$	$\sim 10^{-5}$

表 1.3: 原子炉反ニュートリノ検出器の性能

KamLAND 実験は検出できる Δm^2 が 10^{-5} まであり、他の実験に比べ大幅に性能が向上している。この領域では、太陽ニュートリノ欠損問題の解の一つである MSW の大混合角解が検証できる。

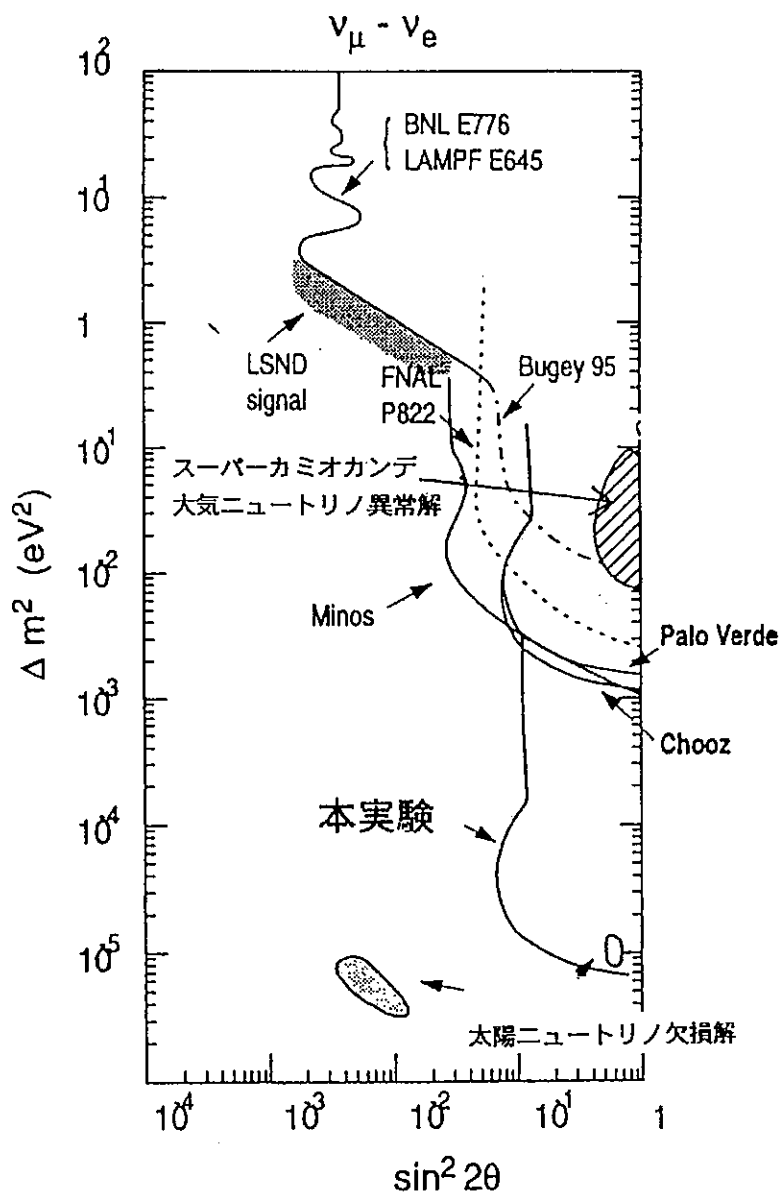
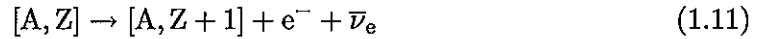


図 1.3: ニュートリノ振動パラメータの探索領域と、太陽ニュートリノ欠損、大気ニュートリノ異常に対する振動解

1.1.2 地球反ニュートリノ

地球は表面から 40TW の熱を放出している。そのうちの約 40% は放射化熱であり、放射化熱の 90% は、地球内部の地殻や、上部、下部マントルに含まれている放射線核種の崩壊エネルギーによるものと考えられている。地球の内部構造および発熱機構については様々な理論モデルが存在するが、それらの理論の検証のためには、地球内部の放射線核種の存在量を直接測定することが必要である。

地球内部の放射線核種はベータ崩壊する際、電子型反ニュートリノを放出する。



地球内部の放射線核種の崩壊にともなって放出される反ニュートリノを検出し、地球内部の放射線核種の存在量を見積もるという提案は、すでに、1966 年に G. Marx や G. Eders らによって行われていたが、あまりにも低エネルギーであり、これまで実現することができなかった。KamLAND 実験では、大容量の液体シンチレータを用いることにより、世界で初めて、この反ニュートリノを検出することができるものと期待されている。

検出される反ニュートリノに寄与する放射性核種は以下の表 1.4 に示すように 4 つあり、 ^{238}U と ^{232}Th の中には崩壊鎖も含める。表 1.4 は、地球表面に近いところに存在する放射性核種ごとの、地上に到達する反ニュートリノのフラックス量を示したものである。

Isotope	Flux [$/\text{cm}^2/\text{s}$]
^{40}K	1.1×10^7
^{87}Rb	4.6×10^6
^{232}Th	3.5×10^6
^{238}U	3.5×10^6

表 1.4: 地球表面から 100km 以内に存在する主な放射性同位体からのフラックス

原子炉反ニュートリノと地球反ニュートリノを合わせた、KamLANDで検出される1年間のエネルギースペクトルを図1.4に示す。破線が地球反ニュートリノ、点線が原子炉反ニュートリノのスペクトルを表し、実線が2つを合わせたスペクトルである。一点破線は $(\sin^2\theta, \Delta m^2) = (0.7, 10^{-5})$ のパラメータでニュートリノ振動が起こった時のものである。誤差は統計誤差のみを示す。

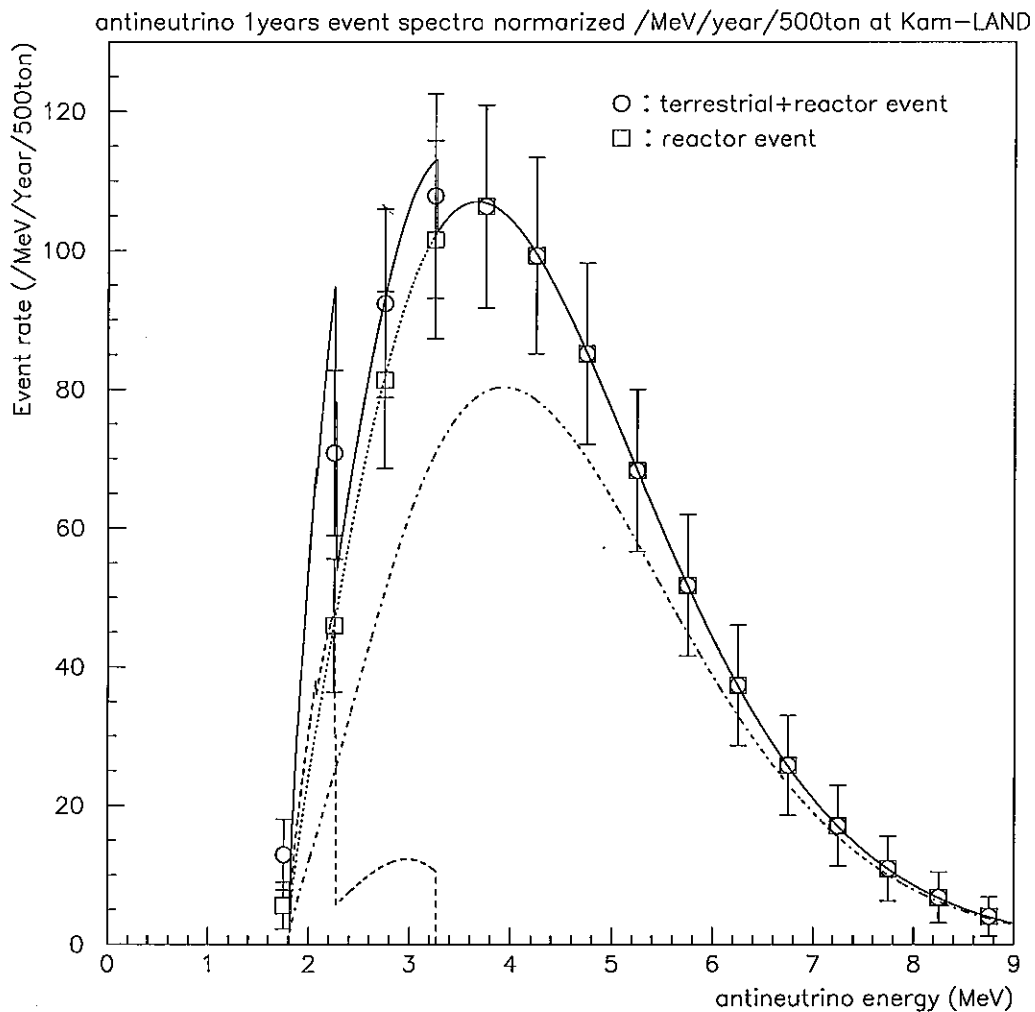


図 1.4:

1.1.3 太陽ニュートリノ

1970年、Homestake 実験により、初めて太陽ニュートリノが観測され以来、いくつかの実験で、太陽ニュートリノが観測されてきた。しかし、いずれの実験でも、観測されるニュートリノの数が、標準太陽模型から予想される数より有意に少ないという事実が明らかになった。これを太陽ニュートリノ問題という。

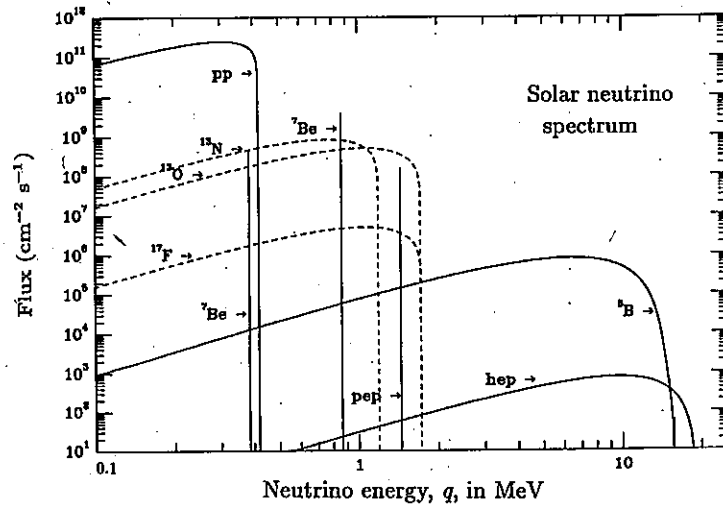


図 1.5: 標準太陽模型から予想されるニュートリノのスペクトル

図 1.5 に標準太陽模型から予想されるニュートリノのエネルギースペクトルを示す。

Homestake 実験は、615 トンの C_2Cl_4 を用いている。この C_2Cl_4 が太陽ニュートリノと以下の反応を起こす。



この反応で生じた放射性的 ^{37}Ar ガスを回収し、比例計数管により ^{37}Ar の崩壊数をカウントして太陽ニュートリノを観測した。この反応によるニュートリノ捕獲反応の閾値は 814keV である。このため、Homestake 実験では、主に 8B ニュートリノと 7Be ニュートリノを観測していることになる。観測されたニュートリノの数は標準太陽模型から予想される値の 29% であった。ただし、この実験では、 8B ニュートリノと 7Be ニュートリノのスペクトルを独立に解析することはできない。

スーパー神岡実験では、太陽ニュートリノと弾性散乱した電子によるチェレンコフ光を検出することにより、太陽ニュートリノを観測している。現在、検出できるニュートリノのエネルギー閾値は 5~6MeV 以上であるため、 8B ニュートリノのみを観測していることになる。検出されたニュートリノの数は、標準太陽模型から予想される値の 35% であったという結果を出している。

KamLAND 実験では、バックグラウンドを抑え、検出器の閾値を十分下げることにより、 8B ニュートリノだけでなく 7Be ニュートリノも検出できる。そして、高いエネルギー分解能を生かして、それぞれのスペクトルを独立に解析することにより、標準太陽模型の信頼性を確かめ、太陽ニュートリノ問題の解決を目指す。

1.2 検出器

図 1.7 に検出器の概要を示す。

検出器の中心には直径 13m のバルーンが設置され、1,000 トンの液体シンチレータで満たされている。光透過性、強度、低放射能などの要請を満たすため、バルーン膜には、ナイロン等の多層膜構造のフィルムを用いる予定である。

バルーンの外側には、直径 17m の球形のステンレスタンクが設置されている。ステンレスタンクの内壁にはバルーンを取り囲むように、約 1,300 本の 17 インチ光電子増倍管が配置されている。この光電子増倍管により、バルーン内部の液体シンチレータ内で発せられた信号を検出する。全体の表面積に対し、光電子増倍管がカバーする割合は約 25% である。将来的には、光電子増倍管を追加することが可能であり、さらにカバー率を上げることができる。

実験で用いる光電子増倍管には、神岡実験で用いられた 20 インチ光電子増倍管を改良し、時間分解能を向上させた物が使われる。外部の形状は 20 インチ光電子増倍管と同じであるが、光電子増倍管の有効光電面を中央部の 17 インチに限定している。これにより、いままでのベネチアン・ブラインド型のダイノードに代わり、ライン・フォーカス型のダイノードを使用することが可能になった。

ステンレス槽とバルーンの間は、バックグラウンドの原因となる、光電子増倍管中の放射性核種による γ 線や、検出器外部の岩盤からの放射線を遮蔽するためのバッファリーキッドで満たされている。バッファリーキッドには、バルーンにかかる浮力を抑えるため、液体シンチレータの主成分であり、液体シンチレータとの比重差が少ないイソパラフィンが用いられている。

ステンレス槽の外側には円筒形の超純水槽がある。岩盤にポリウレタン樹脂を塗布することによって防水タンクを形成し、超純水を保持している。この超純水層も検出器外部からの放射線を遮蔽している。超純水槽の内壁には光電子増倍管が取り付けられている。この光電子増倍管により、超純水槽内部で発生したチェレンコフ光を検出し、検出器外部から来る宇宙線 μ 粒子を識別する。宇宙線 μ 粒子を識別する際に、宇宙線 μ 粒子の経路を限定するため、超純水槽の内部は仕切り板によって分断されている。

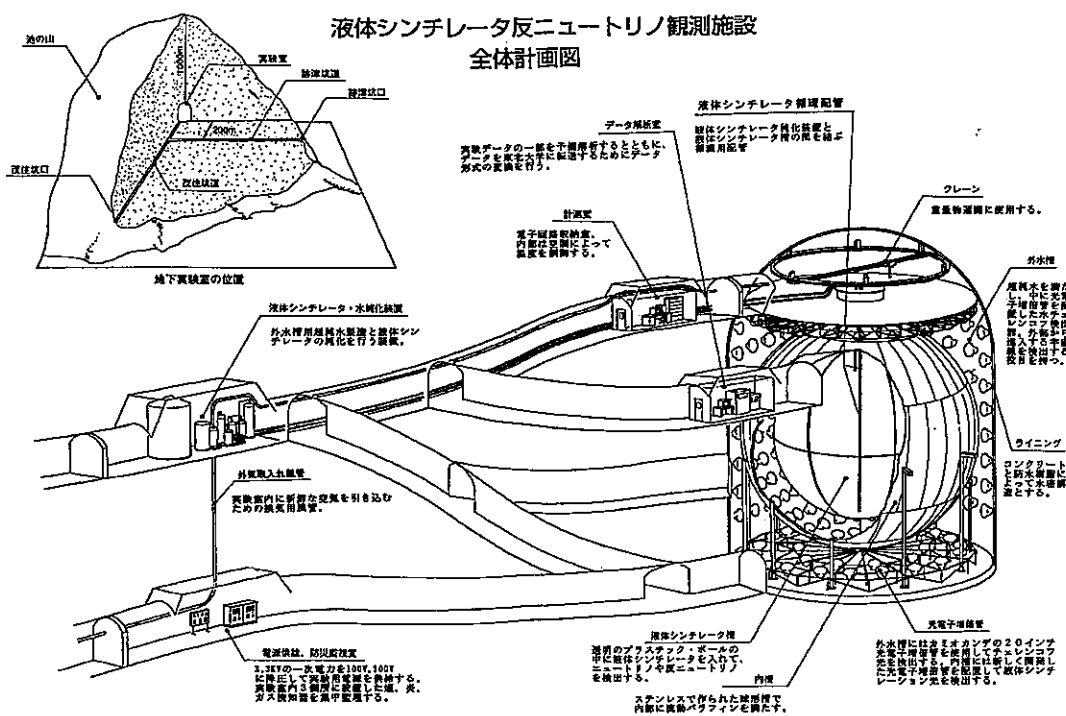


図 1.6: KamLAND 実験施設の全体図

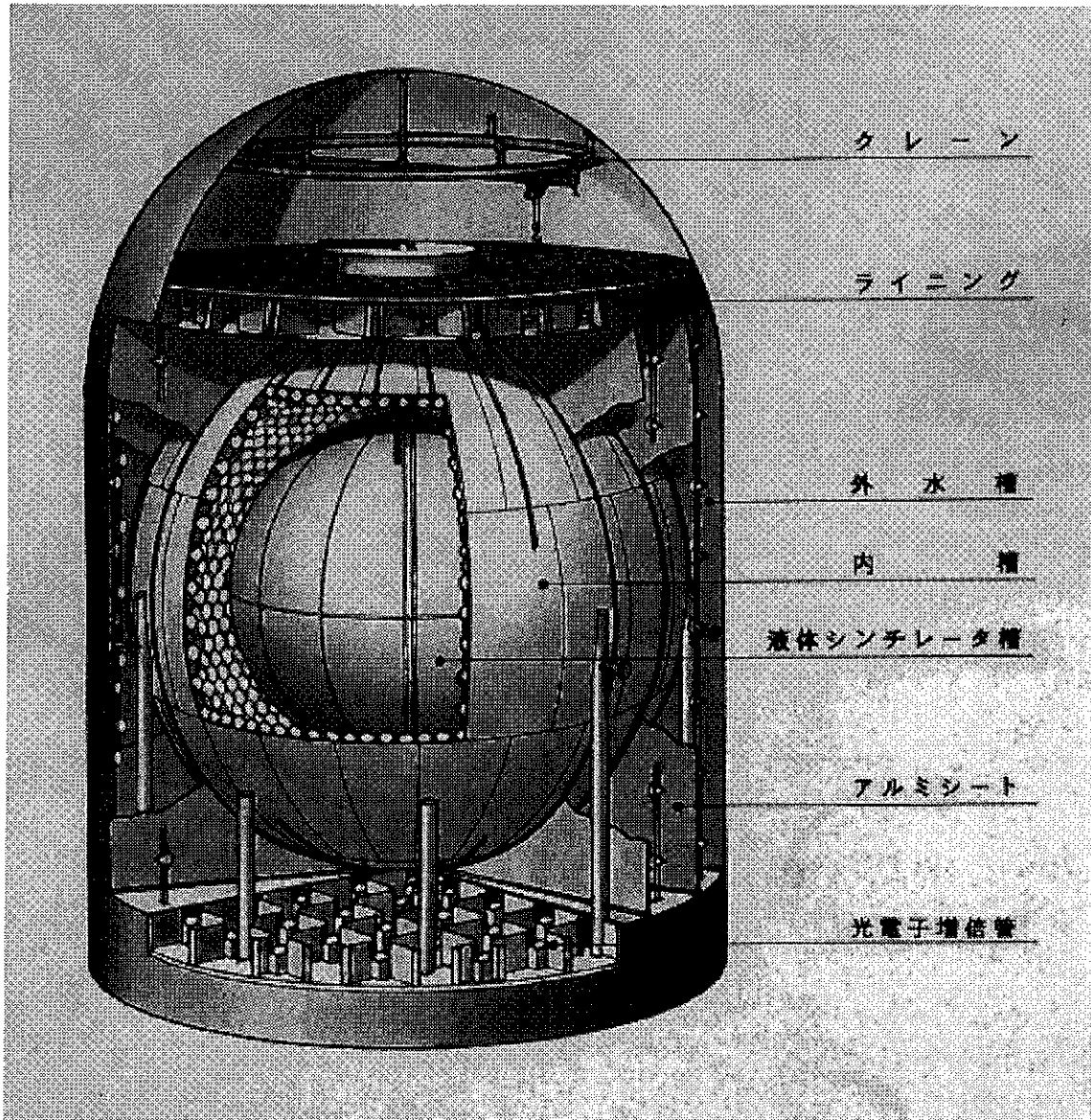


図 1.7: KamLAND の概要

1.3 反ニュートリノ検出原理

1.3.1 遅延同時計数法

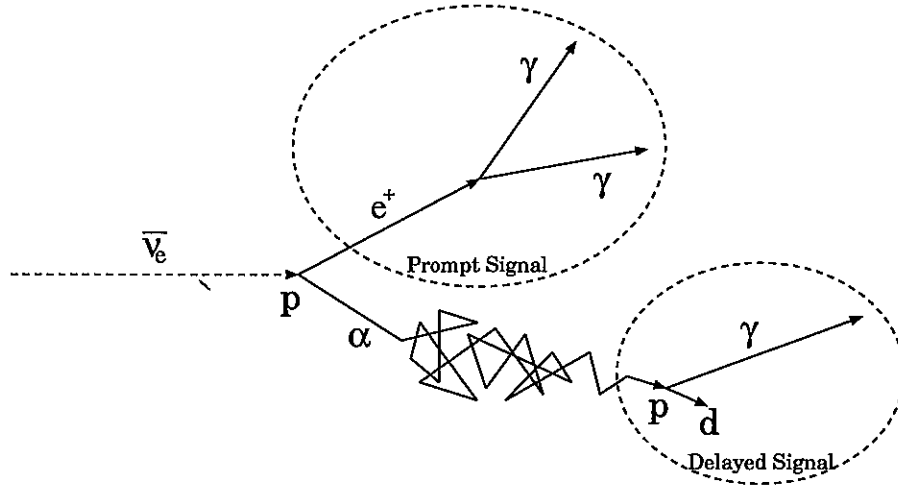


図 1.8: 反ニュートリノによる信号

ここで KamLAND 実験での反ニュートリノ検出方法について述べる。

検出器内に入射した電子型反ニュートリノ ($\bar{\nu}_e$) は、検出器内の水素原子核 (p) と衝突し、陽電子 (e^+) と中性子 (n) を生じさせる。ここで、陽電子の運動エネルギーは、電子型反ニュートリノのエネルギーから閾値である 1.8MeV を引いたものになる。

陽電子はエネルギーを失いながら、液体シンチレータ内を数 cm 飛行し、最後に液体シンチレータ内の電子と対消滅を起こして、0.5MeV の 2 本のガンマ線を生じさせる。これらのガンマ線は平均 10cm 飛行したあと、コンプトン効果と光電効果によりエネルギーを失う。

以上の過程により、液体シンチレータ内に、

$$(\text{入射した}\bar{\nu}_e\text{のエネルギー}) - 0.8\text{MeV} \quad (1.13)$$

のエネルギーが放出され、シンチレーション光となって検出される。この過程は数 ns 以内に行われる。この信号を Prompt Signal と呼ぶ。

一方、中性子は数 10keV 程度の反跳エネルギーを持ち、液体シンチレータ内の原子核と衝突を繰り返して熱中性子化し、平均 $170\mu\text{s}$ 後に、約 30cm 離れた所で陽子に吸収され、2.2MeV のガンマ線を出す。このガンマ線に起因する信号を Delayed Signal と呼ぶ。

これらの 2 つの信号を検出することにより、 $\bar{\nu}_e$ イベントを識別するとともに、バックグラウンドをカットすることができる。

1.4 バックグラウンド

KamLAND 実験では、超低エネルギーニュートリノの検出を行なうため、バックグラウンドをいかに減らすかが重要である。

KamLAND 実験においてバックグラウンドとなりうるのは、宇宙線 μ 粒子によるもの他に、液体シンチレータや、岩盤に含まれる放射性同位体によるものなどがある。

宇宙線 μ 粒子によるバックグラウンドは、神岡鉱山内の地下 1,000m に設置されている KamLAND では、地上の 10^{-5} 程度になる。このため、宇宙 μ 粒子の検出頻度は、およそ 0.47Hz となる。

宇宙線 μ 粒子と炭素との原子核粉砕反応により、 $Z=6$ 、 $A=6$ までの不安定核粒子が生成されることがある。生成された粒子が崩壊する際に放出される電子や γ 線、中性子などもバックグラウンドとなる。

純水槽に設置されたアンタイカウンターにより、宇宙線 μ 粒子を識別し、また、宇宙線 μ 粒子の飛跡にそって半径 1m 以内の範囲で発生したイベントを、3 秒間取り除くことにより、このバックグラウンドを取り除く。

液体シンチレータに U、Th、K などが混入していると、それらの崩壊系列から α 線、 β 線、 γ 線、中性子などが放出され、バックグラウンドとなる。そこで、液体シンチレータ内の放射性同位体を取り除くために、液体シンチレータの純化装置が設置される。

また、光電子増倍管のガラスに含まれる放射性物質もバックグラウンドとなる。これを防ぐために、光電子増倍管の表面を放射性物質が透過しにくいフィルムでおおう。

岩盤中に含まれる U、Th、K などの放射性同位体や、その崩壊系列から放出される放射線もバックグラウンドとなる。この放射線を防ぐため、液体シンチレータ槽は厚さ 2m のミネラルオイル層、厚さ 1m の純水層で岩盤から遮蔽される。

1.5 テストベンチ

1.5.1 目的

現在、KamLAND 実験の予備実験として、東北大学内にテストベンチが設置され、種類の性能を評価している。

テストベンチには、以下のような目的がある。

遅延同次計数法の評価

KamLAND 実験で用いる遅延同時計数法での信号取得が可能かを評価する。KamLAND 実験では電子型反ニュートリノによる信号を検出するが、テストベンチでは $^{241}\text{Am}/\text{Be}$ を線源として信号を検出する。この線源からは、中性子とガンマ線が放出される。放出された中性子は液体シンチレーター中の水素原子核と衝突し、弾き飛ばされた陽子による Prompt Signal と、衝突した中性子による Delayed Signal を発生させる。この2つの信号を同時遅延計数法で検出できるか確かめる。

液体シンチレーターの特性評価

テストベンチでは、液体シンチレータとしてバイクロン社製の BC519 と昭和シェル石油社製のパラオール 250 を混合したものを使用する。この液体シンチレータが、発光量、粒子識別などで期待する性能を発揮しているかどうかを検証する。また、液体シンチレータの粒子識別能は大容量になるほど悪くなるという傾向があるため、大容量液体シンチレータでの粒子識別能を検証するとともに、粒子識別に用いるパラメータの最適値を探る。

純化装置の評価

バックグラウンドの原因となる液体シンチレーター内の放射性不純物は、できるだけ取り除かなければならない。特に、U、Th は 10^{-14}g/g 以下、K は 10^{-11}g/g 以下、という目標がある。そこで、液体シンチレータの純化装置を開発し、実際にこの要求を満たすだけの性能があるか評価する。

液体シンチレータから U、Th、K を除去するには液々抽出法を用いる。金属塩や金属酸化物は、水と液体シンチレーターで可溶性が異なる。金属塩や金属酸化物は極性溶媒である水やメタノールに良く溶けるが、液体シンチレータなどの非極性有機液体には溶けにくい。この性質を利用し、液体シンチレーターを純水の中に通すことにより、液体シンチレーター内に混入している金属塩を純水中に溶かし、液体シンチレータから除去する。

1.5.2 検出器

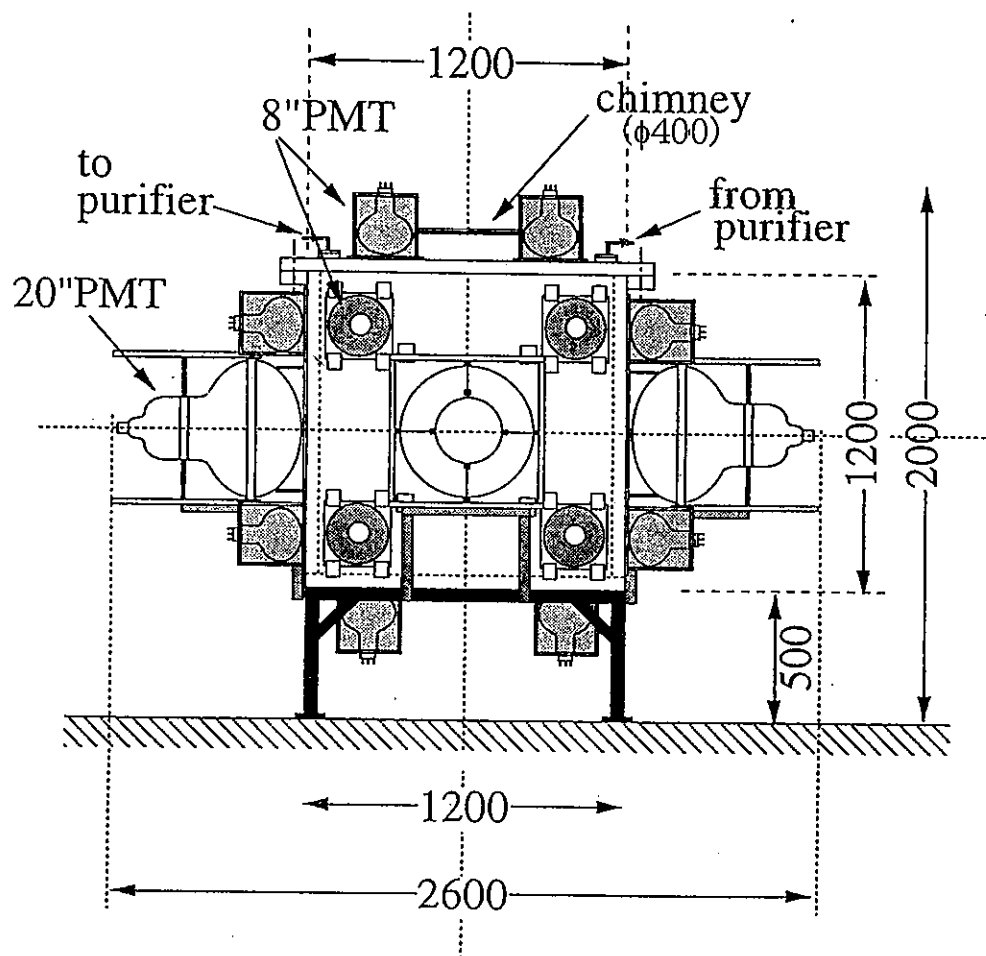


図 1.9: テストベンチ

図 1.9 にテストベンチの概略を示す。中心の亚克力槽は 1.5m の立方体である。側面には 20 インチの光電子増倍管が各 1 本ずつ計 4 本、さらに各面に 8 インチの光電子増倍管が各 4 本ずつ計 24 本設置されている。

亚克力槽の上部には直径 40cm の搬入口があり、ここから線源や液体シンチレータボックスを亚克力槽内部に入れることができる。

第2章 データストレージシステム

2.1 要請

物理実験で得られたデータを記録するシステムは、ただデータを記録できればいいのではなく、そのデータがどのように利用されるのか、つまり、データ解析での利便性も考慮に入れて設計しなければならない。データ解析まで含めた総合的なシステムとして考えた時、以下のような要請が考えられる。

イベントデータの管理

物理実験で記録されるデータは膨大な量になり、多数のファイルに記録される。その時、ユーザーが必要なイベントデータがどのファイルに記録されているのかを探すのは大変である。そこでシステムがファイルを管理し、ユーザーが効率的にイベントデータにアクセスできるような手段を提供する必要がある。

ユーザーが自由にデータを追加できること

イベントデータには、イベントの反応点や粒子のエネルギーなどの情報も含まれると思うが、これはそもそも、イベント再構成による解析の結果の値であり、測定値そのものではない。このように、ユーザー自身が解析した値もイベントデータと区別無く読み出せるようにすることにより、解析の利便性が向上すると思われる。

解析の結果イベントデータに追加されるデータの種類、構造はさまざまであるが、それらはシステム作成後に定義されるため、システム作成時にその構造を知ることはできない。このことは、システムがユーザーデータの記録に関して、サポートするのが難しいことを意味する。

従来、この問題を解決するために、システムがあらかじめデータ構造について制限し、その制限を満たすデータのみ記録することができた。

今回作成したシステムでは、解析プログラム中で利用するデータ構造をそのままの形で記録することができるようにした。このことにより、ユーザーはデータストレージでのデータ構造に制限されることなく、解析プログラムでの解析処理のみに集中できるようになった。

ユーザーによる解析結果を記録する際に、もう一つ重要なのは、どこに記録するかということである。

ユーザーの解析結果をイベントデータに追加する際、オリジナルの共有データに変更

を加えることは許されない。従来のシステムでは、オリジナルのデータのコピーとして、ユーザーの解析結果が追加されたファイルをまるごと作成していた。

今回作成したシステムでは、ユーザーの解析結果のみを別のファイルに記録し、イベントデータを読み込む時に、オリジナルのイベントデータと自動的に結合するようにした。

イベントデータの構造の変更が簡単にできること

KamLAND 実験は、2001 年より測定を開始する予定である。そのため、イベントデータの構造を現時点で決定することはできない。また、測定開始後に実験装置のアップグレードが行なわれることも考えられる。このため、イベントデータの構造の変更が簡単にできる必要がある。

もう一つ重要なのは、一時的なイベントデータ構造の変更をサポートすることである。キャリブレーションなどの特定の測定時には、通常のイベントデータの他に特別な情報を付け加えたいが、キャリブレーション終了時には元のデータ構造に戻したい。

これらの要請を満たすためには、イベントのデータ構造を動的に変更できる機能が必要である。

書き込み、読み込みのスピードが十分早いこと

イベントデータの記録、解析に用いるためには、イベントデータの読み込みにかかる時間が、解析時間に比べて無視できないようでは困る。また、データ収集システムからの要請から、平均のイベントレートより十分速いスピードで書き込めなければならない。1 台のマシンの能力を超えたイベントレートのデータを記録するためには、複数のマシンによる並列的な書き込みもサポートしている必要がある。

ランダムアクセス

物理解析を行なう際、ランダムアクセスが可能であると便利な場面がある。そこで、効率的なランダムアクセスを行なうためのインデックス管理を行なう。ただし、シーケンシャルなアクセスでの利用の方が多と思われるので、シーケンシャルアクセスに対する最適化を十分に行なう。

イベントデータの部分的な読み込み

物理解析では、イベントデータに含まれる全てのデータが必要なのではなく、ある特定の部分のデータのみ必要なことが多い。そこで、必要な部分のみ読み込むことにより、入出力にかかる時間を節約できる。

2.2 設計方針

イベントをセクションに分割

構造が動的に変更されるイベントデータに汎用的にアクセスするには、データに名前を付け、その名前の文字列によってアクセスするようなインターフェースでなければならない。ただし、個々の数値に名前でアクセスするのは、パフォーマンスの観点から勧められない。

実際の利用形態を考えると、イベントデータに含まれる数値はそれぞれ独立ではなく、あるグループごとに分けられる。例えば、イベントナンバーやイベント検出時刻やトリガーIDなどは、ヘッダーというグループにまとめられる。また、既存のイベントデータの構造を変更し、データを追加する時は、数値を1つだけ追加するのではなく、いくつかの数値をまとめて追加すると思われる。

そこで、イベントデータの内部構造として、セクションというものを考える。セクションはある意味を持った値の集まりを表し、データ構造はユーザーが自由に定義することができる。そして、セクションに名前を付け、その名前を指定することで、セクションにアクセスすることにする。

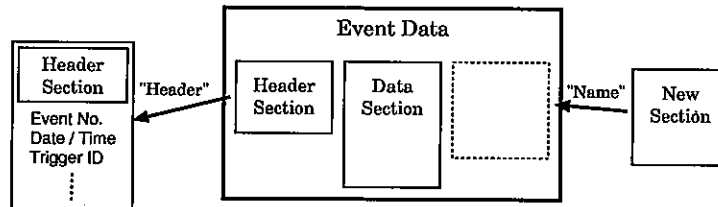


図 2.1: イベント構造

セクション毎にファイルを分割

イベントがセクションにより構成され、セクション単位でデータの取得、登録を行なうため、データファイルもセクション毎に分割する。このようにすることで、セクションを追加した際、他のデータファイルに変更を加える必要がなくなる。

ただし、1イベントのデータが複数のファイルに分割されることになるため、それらのファイルをシステムが管理し、ユーザーに複数のファイルの管理を意識させないようなインターフェースを用意する必要がある。

ディレクトリ階層

セクション毎にファイルが分割されるため、多数のファイルが生成されることになる。ファイルの管理はシステムによって自動的に行なわれるが、実際に、どのイベントがどのファイルに記録されているか直観的に分かる方が便利である。

そこで、ラン毎にディレクトリを作成し、そのランに含まれるイベントのセクションファイルは、すべてそのディレクトリ内に記録されるようにした。ディレクトリには、そのランを表す名前が付けられる。また、セクションファイル名は、セクション名+連番となる。

そして、ランのディレクトリは、1つのディレクトリ内にまとめて置かれる。そのディレクトリを頂点としたディレクトリツリーが1つのデータストレージを表す(図 2.2)。

このディレクトリツリーは複数の場所に作成することができ、それぞれのデータストレージのアクセス権を変更できる。例えば、パブリックなデータディスク上に共有なデータストレージのディレクトリツリーを構築し、その中に含まれるイベントデータは読み取り専用でメンバーに利用されることになる。また、個々のユーザーの解析結果は、そのユーザーのホームディレクトリに構築したデータストレージのディレクトリツリー内に記録する。そして、イベントデータを利用する際には、システムによって、それらの複数のデータストレージ内のデータが自動的に結合される(図 2.3)。ユーザーは利用するデータストレージのリストをシステムに通知すればよい。

データストレージのリストの内容は以下ようになる。

```
1: Read:
2:   /kamland/data1/
3:   /kamland/data2/
4:   $HOME/data
5:
6: Write:
7:   $HOME/data
```

Read:、Write:の各モードごとに、利用するデータストレージのディレクトリツリーのパスを列記する。このリストの順番に従ってシステムはデータを検索することになる。

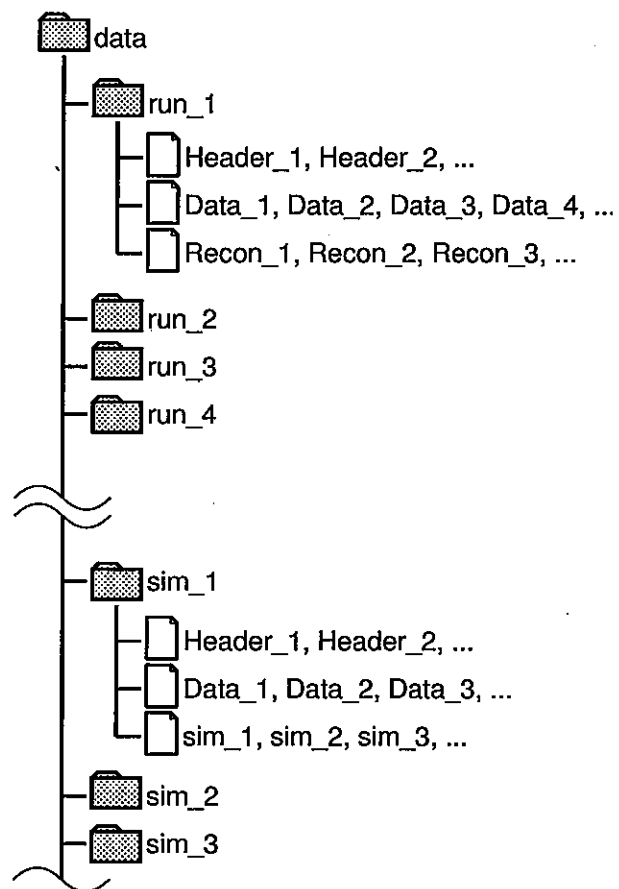


図 2.2: ディレクトリ構造

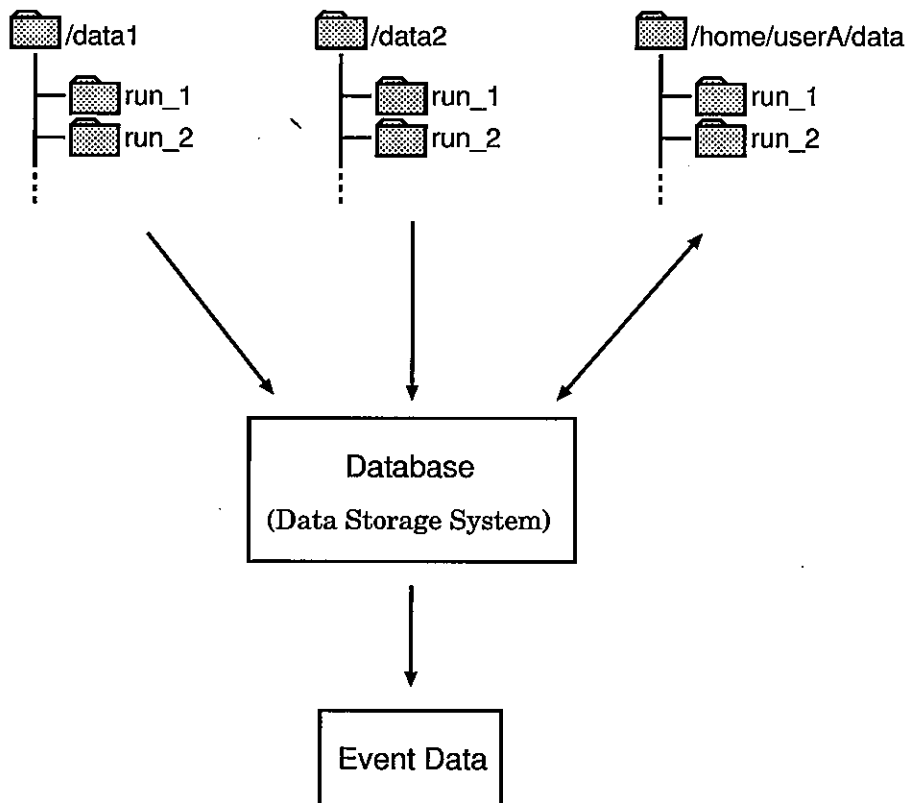


図 2.3: データストレージシステム

2.3 問題点

どのようにデータファイルを分割し、配置するか設計方針は決まったが、個々のファイルにどのようにデータを記録するかを決めなければならない。

ここでは、任意のデータ構造をファイルに記録しなければならない時に遭遇する問題点について述べる。

セマンティクスギャップ

ある構造を持ったデータをファイルへ記録する時、ファイルには一次元の構造を持ったバイナリデータとしてしか記録できないため、プログラム中で利用するデータ構造から一次元のデータ構造へ変換しなければならない。この構造の違いをセマンティクスギャップという。

一般に、この変換は表現力の大きなモデルから、表現力の小さなモデルへの変換となるため、一意に変換することができない。結局、データ構造ごとに固有の変換処理が必要になるため、個々のデータ構造に対して、ユーザーが変換処理を実装しなければならなかった。

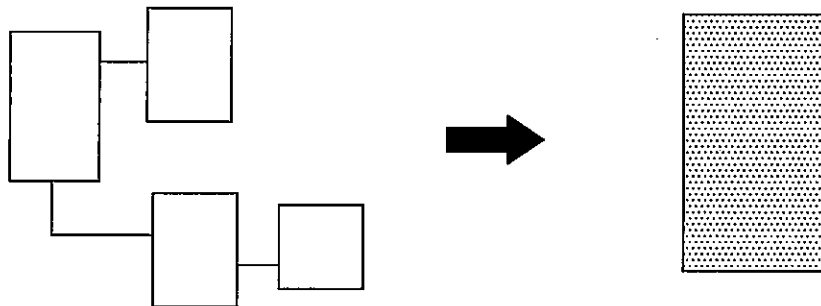


図 2.4: セマンティクスギャップ

オブジェクトの動的生成

ファイルに記録されたバイナリデータからデータを復元するには、そのデータ構造固有の(逆)変換処理を実行しなければならない。しかし、その変換処理はデータ自身によって実行されるので、記録されたデータを復元するためには、データ自身が復元されていなければならないというジレンマが生じる。

これを解決するためには、なんらかの手段で、記録されているデータと同じ型を持った空のデータを動的に生成することが必要になる。ここでいう動的とは、生成するデータの型をプログラム実行中に動的に指定するということである。

このように、データの型をプログラム中で情報として利用できる機能は、実行時型情報(RTTI)といい、任意のデータ構造を扱うようなシステムでは必須であるが、これまでのプログラム言語では、ほとんどサポートされていなかった。

2.4 ROOT System

このような問題点を解決するため、今回のシステムのベースとして、ROOT Systemを採用した。ROOT Systemとは、CERNにより提供されている物理解析のためのシステムである。

ROOT Systemには、ユーザーが定義したデータ構造に対応した変換処理を、自動的に生成するツールが含まれている。このツールで自動生成した変換処理を利用して、ユーザーが定義したデータをファイルへ記録できるようになる。

また、同じツールによって、そのデータ構造に関する情報を、プログラム中から利用し、動的に生成することが可能になった。

2.5 システムの概要

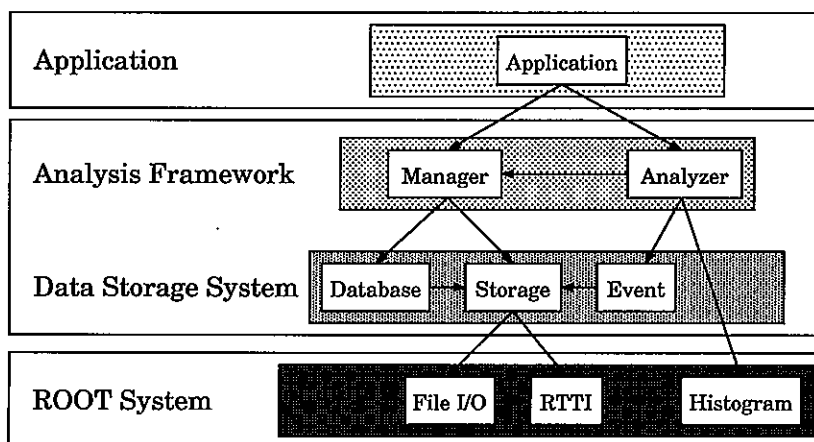


図 2.5: システムの概要図

図 2.5 に、システム全体の概要図を示す。図中の矢印は関連する機能を結びつけたものであり、矢印の向きが依存関係を示している。

まず、システムの最下層に ROOT System を利用している。実際の物理実験では、データ収集、解析において、さまざまなプラットフォームを利用することが考えられる。そこで、ファイル入出力などシステムに依存する機能は、ROOT System のインターフェースを通して利用することで、マルチプラットフォームを実現している。

ROOT System の機能を利用し、データストレージシステムが構築されている。データストレージシステムには、実際にデータの記録、読み出しを行うストレージ、ストレージをプログラム中から利用するためのインターフェースであるイベント、さらに、イベントデータの検索、管理を行うデータベースが含まれる。これらについて、この章のセクション 2.6 で説明する。

解析フレームワークには、イベントデータを用いて解析処理を行うアナライザ、データ

ストレージシステムからイベントデータを取得し、登録されたアナライザにより解析処理を行う解析マネージャがある。これらについては、第3章で説明する。

そして、これらの上に、実験ごとに必要なイベントデータ構造や解析処理を記述することによって、個々の実験のアプリケーションを構築する。

2.6 実装

2.7 インターフェース

インターフェースによって、ファイル管理や、データの書き込み、読み込みなどの処理をユーザーから隠蔽し、ユーザーはイベントデータを用いた解析処理のみに専念できる。

今回作成したデータストレージシステムでは、ラン - イベント - セクションのデータ階層によるインターフェースを用いて、イベントデータにアクセスしている。

まず、ユーザーはデータベースから希望するランを受けとる。次にそのランに含まれるイベントを取得し、セクションにアクセスする。ユーザーからのセクション要求があった時点で初めて、ストレージからデータが読み出される。

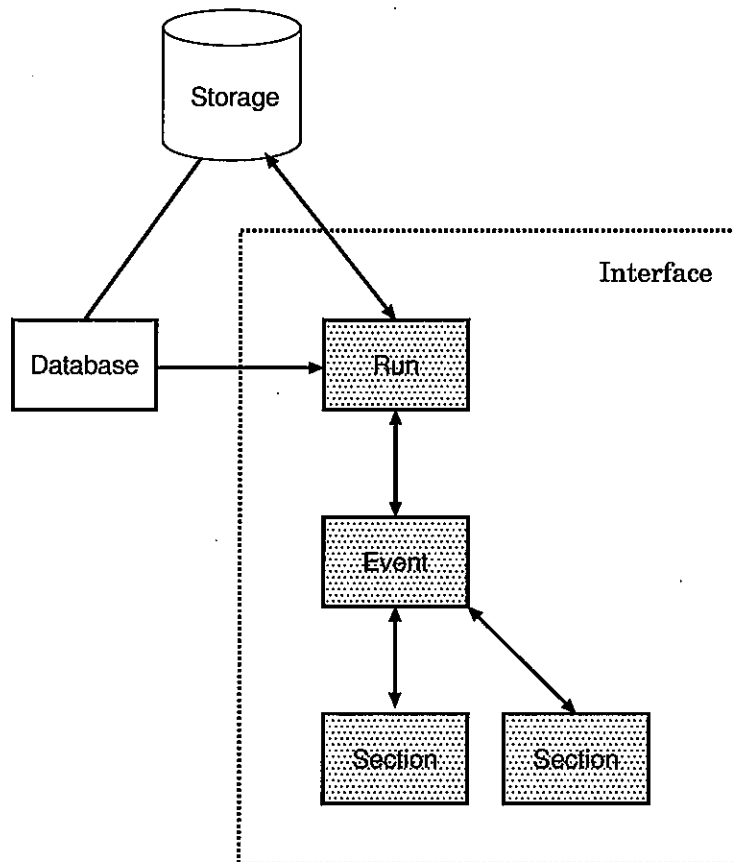


図 2.6: ラン - イベント - セクションによるインターフェース

2.8 ストレージ

実際にデータの記録、読み出しを行う機構がストレージである。ユーザーが直接にストレージにアクセスすることはない。ストレージは、図 2.7 のように、セクションストレージ、ランストレージ、データストレージの3つの階層に分かれている。以下にそれらを説明する。

2.8.1 セクションストレージ

ストレージの最下層であり、ファイルと直接データをやりとりする。セクションごとに、そのセクションを管理するセクションストレージが作成され、効率的な入出力を行なうためのインデックス管理を行なう。以下に、セクションストレージがどのようにセクションの書き込み、読み込みを行なうかを述べる。

2.8.2 ランストレージ

ランの中に含まれる全てのセクションストレージを管理する。インターフェースであるイベントにセクションが追加されると、対応するセクションストレージにセクションを渡して記録する。もし、該当するセクションストレージが存在しない時は、新たに作成する。

ランストレージごとにディレクトリが作成され、ランストレージが管理するセクションストレージのファイルは、すべてその中に記録される。

2.8.3 データストレージ

ストレージの最上層であり、データセット全体を管理する。ランストレージ、セクションストレージは、データストレージを頂点としたディレクトリ構造に作成される。

ストレージのパーミッションおよび、読み込み、書き込みなどのモードはデータストレージ単位に設定される。

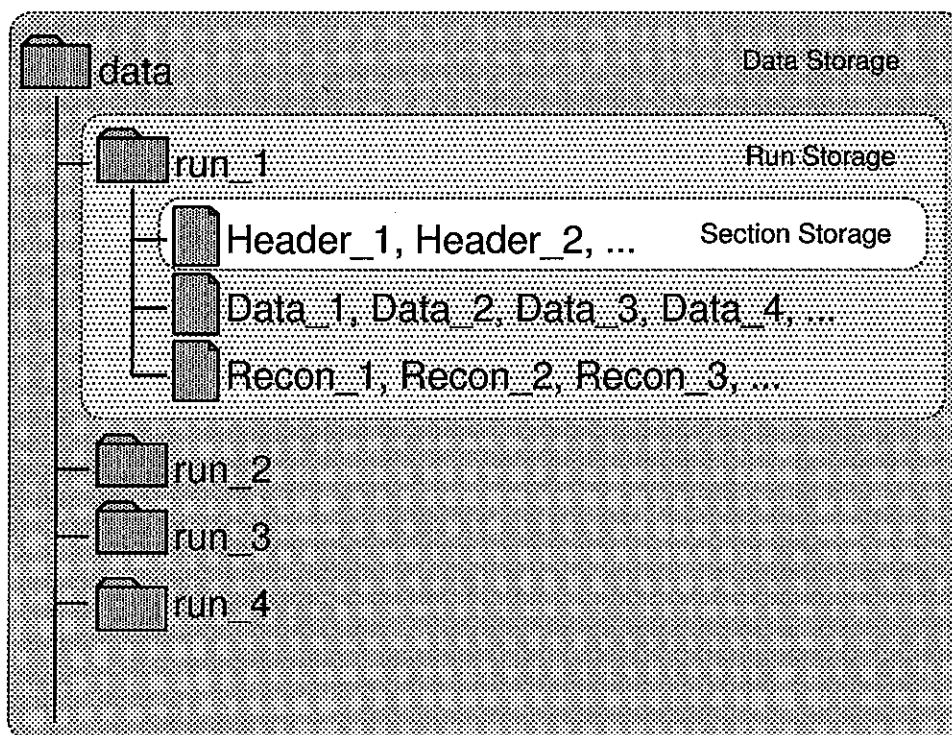


図 2.7: ストレージ

セクションの書き込み

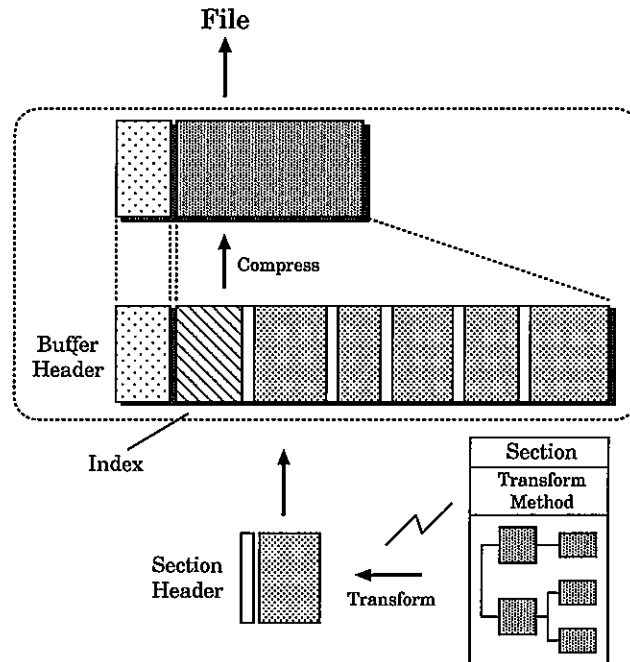


図 2.8: セクションの書き込み

セクションを記録する際、セクションストレージに渡されたセクションは、いったん、セクションストレージ内のバッファに保持される。このバッファをブロックと呼ぶ。

これは、ファイルへの入出力を行なう際、効率のため、1イベントずつ入出力を行なうのではなく、何イベントかのセクションをまとめて入出力を行なうためである。一度に入出力を行なうイベント数は、セクションのデータサイズにより、最適値が異なるため、セクションごとにユーザーがカスタマイズできるようになっている。

セクションはブロックに格納される際、セクションごとに実装された変換処理より1次元の構造へ変換される。この変換処理は、ROOTのツールを用いることにより自動生成される。

変換されたセクションデータには、読み込み時にセクション構造を復元するためのヘッダ情報が付加される。また、この時、セクションデータがブロック内のどの位置に記録されたかを管理するインデックスが更新される。

指定されたイベント数のセクションがブロックに記録されると、ブロック内のデータは圧縮されファイルに記録される。

セクションの読み込み

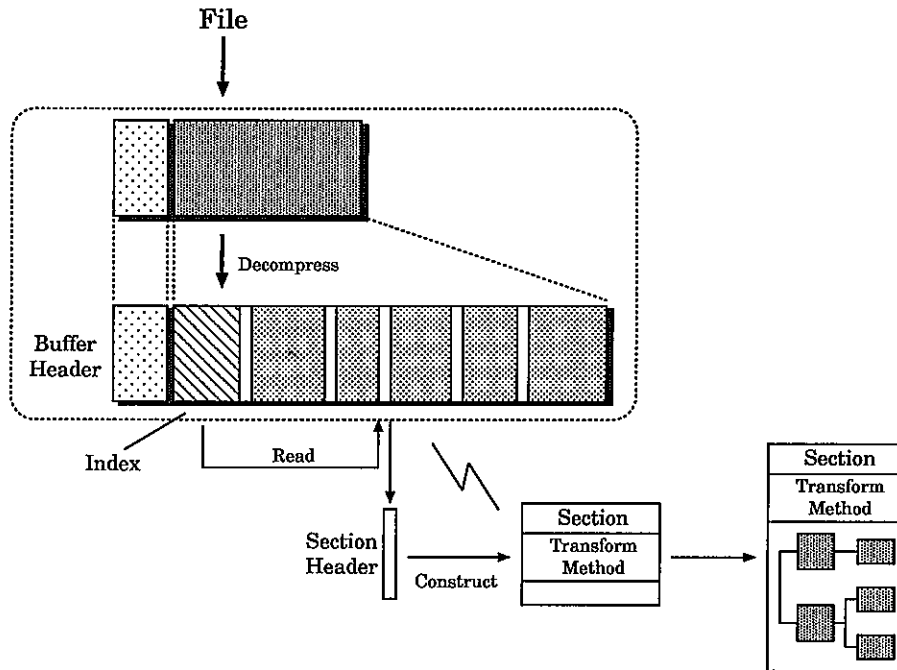


図 2.9: セクションの読み込み

セクションの読み込みは、書き込みに比べて少し複雑になる。それは、読み込む時には、まだセクションが構築されていないため、ファイルに記録されている1次元の構造を持ったデータから、セクションデータに変換するための変換処理を呼び出すことができないからである。

そこで最初に、セクションのヘッダ情報から、セクション自身を構築しなければならない。

まず、セクションストレージ内のインデックスを検索し、該当するセクションがどのブロックに記録されているか調べる。

セクションが記録されているブロックが、現在のブロックでなければ、該当するブロックをファイルから読み込む。ブロックは圧縮されてファイルに記録されているため、ここで復元する。復元されたデータサイズと圧縮前に記録したデータサイズとチェックし、データの整合性を確かめる。

復元されたブロックから該当するセクションのヘッダ情報を読み込み、空のセクションを構築する。さらに構築されたセクションの変換処理を用いて、セクションのデータをブロックに記録されているデータから再構築する。

2.9 データベース

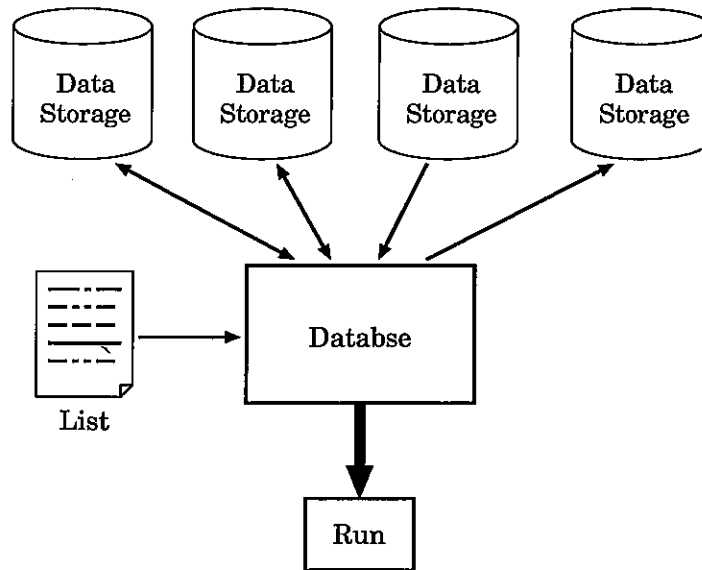


図 2.10: データベース

ユーザーから渡されたリストにしたがってストレージを検索し、ユーザーの指定した条件に一致するランをユーザーに受け渡す。

データストレージシステムを使用するユーザーは、最初にデータベースをオープンしなければならない。データベースをオープンする際に、データストレージの検索リストを受け渡す。オープンされたデータベースは、それ以降のすべてのデータ操作において、リストに書かれている順番にしたがってデータストレージを検索する。

ユーザーからラン取得の要求があると、リストに書かれているデータストレージから、対応するランストレージを受けとり、ランと結合させる。

ユーザーはストレージと結合されたランを受けとることによって、データにアクセスすることができる。

第3章 解析フレームワーク

3.1 概要

解析フレームワークとは、ユーザーの解析プログラムとデータストレージをつなぐ役割するものである。解析を行なうユーザーは、フレームワークを通じてデータを取得し、必要な解析を行なう。また、個々の解析プログラムにフレームワークにそった共通のインターフェースを持たせることにより、解析プログラム同士を部品の様に柔軟につなぐことができる。

物理解析では、ある段階の解析処理が、その前段階の解析結果に依存することがよくある。そこで、各段階の解析処理を部品の様に扱うことにより、他の部分に影響を与えずに、特定の部分の解析手法を簡単に切替えることができる。

共通のインターフェースを用いることで、他のメンバーが開発した解析手法を簡単に自分の解析に組み込むことができる。また、自分が開発した解析手法を他のメンバーに簡単に利用してもらうことができる。

そして、このような解析部品が充実すれば、ユーザーは部品を組み合わせるだけで解析を行うことができる。もし、必要な解析部品が無いときは、その部分だけを実装すればよく、一度実装した部品は、他のユーザーが簡単に利用することができる。

このように、作成した解析処理の再利用を促すことにより、解析処理の手間を減らすとともに、信頼性のある解析を行うことができる。

また、解析処理を共通のインターフェースで実装することにより、多くの解析処理に必要な共通の処理をくくり出し、システム側で提供できる。その結果、解析処理を実装するユーザーは解析処理の記述に専念できる。

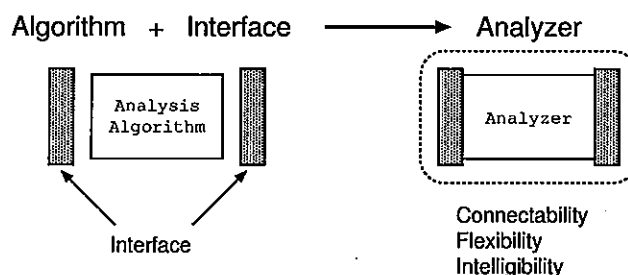


図 3.1: アナライザの概念

3.2 設計および実装

解析フレームワークは、多数のアナライザとそれを管理する解析マネージャで構成されている。

解析フレームワークで、解析を行なう部品としてのインターフェースがアナライザである。図3.2のように、アナライザ同士は共通のインターフェースにより自由につながることができる。また、ある部分のアナライザを別の手法を用いた解析を行なうアナライザに簡単に置き換えることができる。

ユーザーは自分の解析処理をアナライザとして実装し、そのアナライザを解析マネージャに登録することで解析を行なう。

解析マネージャは登録されたアナライザを管理し、その結合関係にしたがってアナライザの解析処理を実行する。解析マネージャはデータストレージと通信し、必要なイベントデータを取得し、アナライザに受け渡す。

以下に、アナライザ、解析マネージャについて述べる。

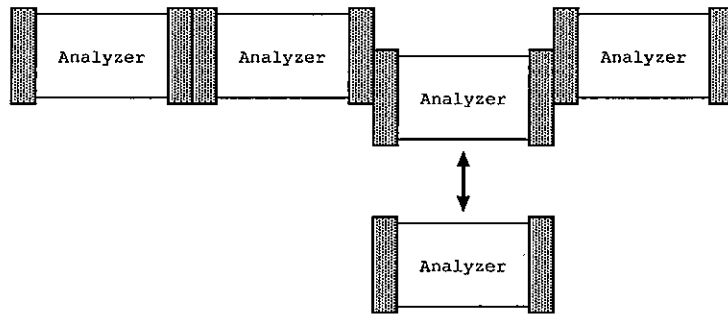


図 3.2: アナライザの結合図

3.3 アナライザ

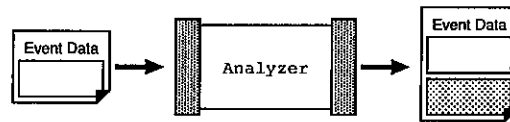


図 3.3: アナライザ

アナライザは解析マネージャからイベントデータを受けとり、何らかの処理を行なう。アナライザによる解析結果は、データストレージの機能を利用することにより、イベントデータに自由に追加することができる。その結果、このアナライザの下流に接続されたアナライザは、このアナライザによる解析結果が追加されたイベントデータを受けとることになる。

解析マネージャに登録されたアナライザは、初期化処理、解析処理、終了処理の3つの処理が実行される。これらの処理内容を実装することで、さまざまな処理を行なう。

初期化処理

解析マネージャに登録された全てのアナライザは、解析の最初に初期化処理が実行される。この初期化処理では、解析を通じて最初に一度だけ必要な処理を記述する。例えば、ヒストグラムの初期化、外部ファイルのオープン、一度だけ計算すればよい値の計算、などが考えられる。

解析処理

解析マネージャによって、イベントごとに実行される。解析マネージャは、登録されたアナライザの順序や結合関係に従って、アナライザの解析処理を実行する。解析処理を実行する際、解析マネージャによってイベントデータが渡される。アナライザはそのイベントデータを用いて、さまざまな処理を行なう。複数のイベントに関する解析は、途中結果をアナライザ内部に保持し、終了処理でデータストレージに記録することになる。

終了処理

解析マネージャによって、全てのイベントを解析した後に実行される。ここで、作成したヒストグラムの表示や、解析結果の表示や記録、ファイルのクローズなどを行なう。

3.4 セレクタ

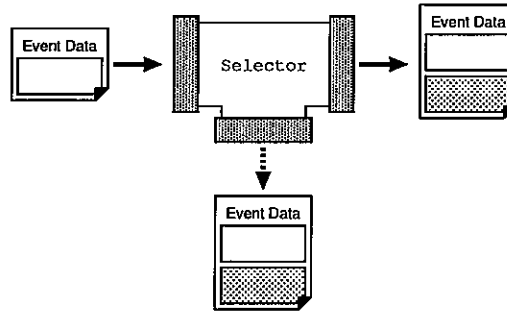


図 3.4: セレクタ

イベントデータを解析した結果により、その後の処理を振り分けたい時がある。そこで、アナライザの一種としてセレクタという部品を用いる。

セレクタには一つの入口と複数の出口があり、セレクタによる条件判断の結果により、次にどのアナライザの処理を実行するか変更できる。それぞれの出口にはアナライザおよびセレクタを接続することができる。セレクタの出口に別のアナライザやセレクタを接続しなければ、そのイベントに対する処理はそこで終了する。

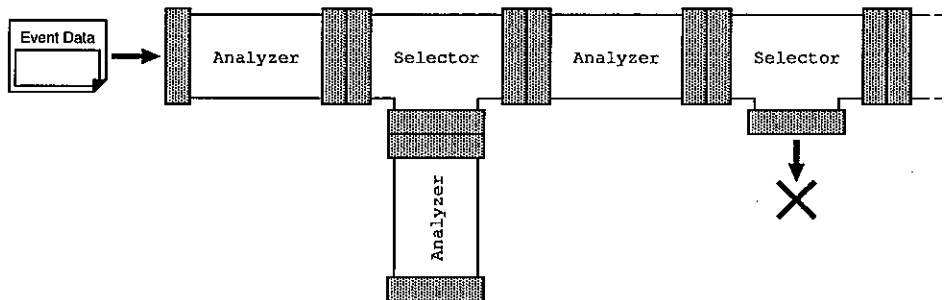


図 3.5: アナライザやセレクタの接続

アナライザとセレクタを次々と接続することにより、図 3.5 のように、アナライザ自体をロジックを持った構造に組み立てることができる。

3.5 複合アナライザ

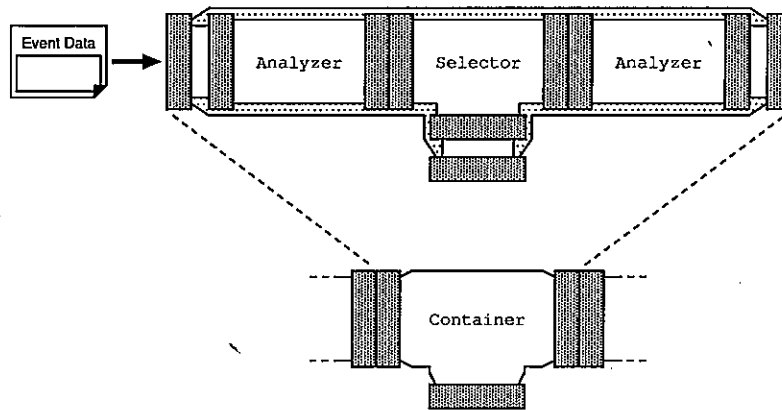


図 3.6: 複合アナライザ

アナライザを接続して解析処理を組み立てることにより、複数の手法を簡単に切替えることができるが、優れたアナライザの組合せが確立したら、そのアナライザの組合せをまとめて扱うことができる方が便利である。

そのためには、複数のアナライザを一つにまとめるような機構が必要になる。

複合アナライザは、その内部にアナライザ、セレクタを登録することができ、外見上は1つのアナライザのように振舞う。

3.6 解析マネージャ

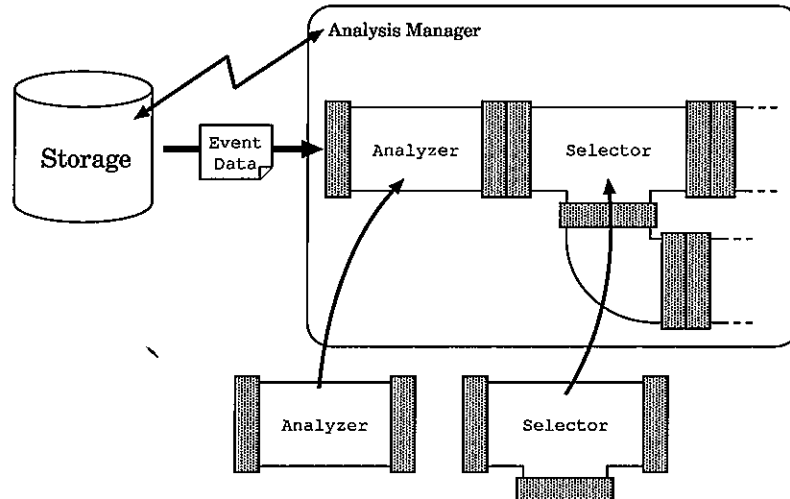


図 3.7: 解析マネージャ

アナライザの接続によって組み立てられた解析処理を登録し、データストレージとの受渡しをするのが解析マネージャである。

解析を行うには、まず、ユーザーは多数のアナライザ、セレクタの中から、自分の解析に必要なものを選択し、解析マネージャに登録する。このとき登録されたアナライザの構造によって、解析処理の内容が決定される。

次に、どのイベントを解析するかを解析マネージャに指定し、解析を開始する。解析が開始されると、解析マネージャは登録されたすべてのアナライザを走査し、その初期化処理を実行する。

すべてのアナライザの初期化処理が終了すると、指定されたイベントリストにしたがって、データストレージからイベントデータを取得し、最初のアナライザの解析処理を実行する。

登録されたアナライザ、セレクタの構造に従って解析処理を次々と実行し、実行するアナライザが無くなったところで、そのイベントに対する処理を終了し、イベントリストの次のイベントを取得する。

すべてのイベントの解析処理が終了すると、登録された全てのアナライザの終了処理を実行する。

第4章 性能評価

データストレージの書き込み、読み込みの性能を評価した。使用したマシンは、Sunのワークステーションで以下のようなスペックとなっている。

CPU	UltraSparc-II × 2
Clock	296 MHz
Memory	256MB

データ構造

テストベンチには、20インチの光電子増倍管4本と、8インチの高電子増倍管24本が設置されている。

軽粒子と重粒子による液体シンチレータの発光特性の違いを利用して粒子識別を行うため、それぞれの光電子増倍管ごとに Total Charge と Tail Charge の2つの値が記録される。さらに光の検出時刻が記録されるため、テストベンチで利用するデータ構造を図のように設計した。

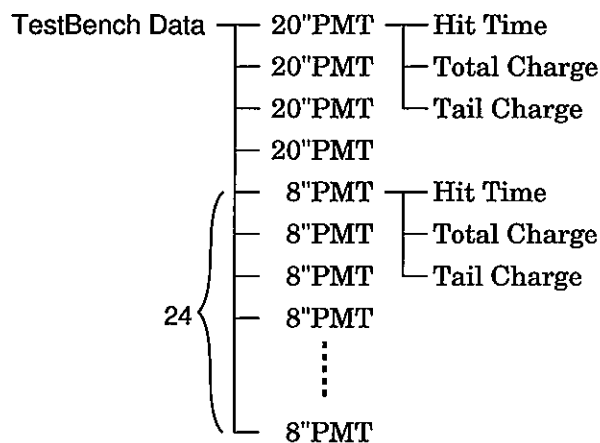


図 4.1: テストベンチで用いるデータ構造

4.0.1 結果

設計したデータ構造のデータを10,000イベント作成し、書き込み、読み込みに掛かった時間を計測した。今回作成したシステムでは、データの圧縮をサポートしているため、圧縮あり、圧縮なしの場合に分けて計測した。

図4.1に測定結果を示す。圧縮ありで記録した場合、圧縮なしの場合に比べて、ファイルサイズが38%に減少した。

	File Size	Write		Read	
		Real Time	CPU Time	Real Time	CPU Time
Uncompressed	11.76 Mbytes	13.5 sec	11.7 sec	13.8 sec	10.2 sec
Compressed	4.44 Mbytes	23.9 sec	23.1 sec	15.2 sec	12.0 sec

表、4.1: データストレージのデータ入出力性能

第5章 まとめ

KamLAND 実験のための、イベントデータの記録・管理を行なうシステムを開発した。

このシステムにより、以下のようなメリットが考えられる。

- データファイル管理を行ない、必要なイベントを自動的に取得
- イベント構造の柔軟な変更が可能
- 自由なデータ構造をそのまま記録できる
- イベントデータの特定の部分のみの読み込み
- ランダムアクセス

データストレージにアクセスし、必要なデータを取り出して解析を行なうためのフレームワークを開発した。

このシステムにより、以下のようなメリットが考えられる。

- 解析処理を部品化・再利用
- 特定の部分の解析手法を簡単に切替えることができる
- 部品を組み合わせるだけで解析を実行できる

参考文献

- [1] 石塚圭樹, オブジェクト指向データベース, アスキー出版局, 1996
- [2] ROOT System Home page, <http://root.cern.ch>
- [3] KONOE Home page, <http://jsdns1.kek.jp/~konoe>
- [4] E. Gamma, R. Helm, R. Jhonson, J. Vlissides, オブジェクト指向における再利用のためのデザインパターン, ソフトバンク, 1995