資 料 3−2

Belle データ損失調査委員会 最終報告書

山中 卓 (大阪大学大学院理学研究科)
坂本 宏 (東京大学素粒子物理国際研究センター)
堺井義秀 (KEK 素粒子原子核研究所)
佐々木 節 (KEK 共通基盤研究施設 計算科学センター)

2012 年 11 月 6 日

目 次

1 Belle データ損失調査委員会 **3**
1.1 はじめに 3
1.2 Belle データ損失調査委員会 3
1.2.1 委員 3
1.2.2 調査委員会 3
1.2.3 担当 SE へのインタビュー 4

2 データ損失の事実関係 **4**
2.1 B 計算機の大容量ストーレジシステム 4
2.2 Belle のデータ 5
2.2.1 データの種類 5
2.2.2 データの在処 6
2.3 B 計算機システムの移行の方針と契約 6
2.3.1 データの移行の方針 6
2.3.2 データ移行の契約 6
2.4 Belle のデータの移動作業 7
2.4.1 データの移動作業の概略 7
2.4.2 データの移動作業手順 8
2.5 Belle のデータ損失 9
2.5.1 ファイルが失われた段階 9
2.5.2 コピーされなかったファイルの比率 10
2.5.3 コピーされなかったファイルのパターン 10
2.6 データの復旧と最終的な消失 12

3 データ損失の原因とその背景の分析 **14**
3.1 技術的な面 15
3.2 データ移行の体制 16
3.3 問題を引き起こした背景 17

4 今後の対策の指針 **18**

5 最後に **19**

A コピーすべきファイルのリストを作る Python スクリプトの例 **20**

1 Belle データ損失調査委員会

1.1 はじめに

KEK Belle 実験がデータ解析に使用していた B ファクトリー計算機システム (以下 B 計算機) は 2012 年 2 月に運用が終了した。それに伴い、B 計算機に付属していた大容量ストーレジシステム (Hierarchical Storage Management; HSM) に蓄えられていた大量のデータを、2012 年 4 月から運用の始まった KEK 中央計算機システム (以下 新中央計算機) に移行する作業が行われたが、その際に大量のデータが失われた。

このデータ損失の事実関係および原因を明らかにし、今後の対策の指針を示すために KEK 外部の委員を入れた「Belle データ損失調査委員会」が作られた。

1.2 Belle データ損失調査委員会

1.2.1 委員

Belle データ損失調査委員会の委員は次の 4 名である。

- 山中 卓：大阪大学大学院理学研究科・教授（委員長）
- 坂本 宏：東京大学素粒子物理国際研究センター・教授
- 堺井義秀：KEK 素粒子原子核研究所・教授
- 佐々木 節：KEK 共通基盤研究施設 計算科学センター・教授

1.2.2 調査委員会

下記の調査委員会を開き、聞き取り調査を行った。

- **日時：**
  2012 年 9 月 26 日 (水) 13:20 ～ 16:50
- **場所：**
  KEK 2 号館 1 階会議室（大）
- **参加者：**
  調査委員会委員、峠暢一理事、[redacted] (KEK 素核研・Belle)、[redacted] (KEK 計算科学センター)、Belle 関係者など傍聴者約十名
- **議事：**
  - 委員打合せ (closed session)
  - 挨拶　– 峠理事
  - Belle 担当者報告（経過、損失データ等）　– KEK/Belle [redacted] 氏

- 計算科学センター担当者報告（業者・契約等）– KEK/計算科学センター[redacted]氏
- 今後の対策の議論 – [redacted]氏 と委員
- Closed session

- **配布された資料：**
  「参考資料」の [1] ～ [8]、他多数。

1.2.3 担当 SE へのインタビュー

また後日、作業を担当した SE の人にインタビューを行い、より詳細な技術的な手法などを聞いた。

- **日時：**
  2012 年 10 月 30 日 (水) 15:00 ～ 16:15
- **場所：**
  [redacted]本社
- **参加者：**
  山中、堺井、[redacted]の SE 1 名
- **質問内容：**
  - SE の役割分担と作業環境
  - データ移行作業の技術的な詳細
  - 行ったチェック
  - 作業に際して困ったこと
  - 将来への提言

2 データ損失の事実関係

2.1 B 計算機の大容量ストーレジシステム

B 計算機の大容量ストレージシステム (HSM) は、bhsm01, bhsm02, ··· bhsm11 と bhsm21 と名付けられた 12 台のファイルサーバーシステムからなる [2]。各ファイルサーバーシステムは 6 個の partition からなり、各 partition には磁気ディスクがある。そのディスクに書かれたファイルのうち 8 MBytes 以上のものは自動的に磁気テープに移動される。また、読み出しの要求のあったファイルがディスクの上に残っていない場合は、そのファイルが自動的に磁気テープからディスクにコピーされる。従って、各 partition は巨大なディスクのようにユーザーからは見える。例えば bhsm01 サーバーの partition は`/bhsm/h011`, `/bhsm/h012`, `.../bhsm/h016` という directory として見える。

2.2 Belle のデータ

2.2.1 データの種類

Belle が扱うデータには、図 1 に示すような種類がある。

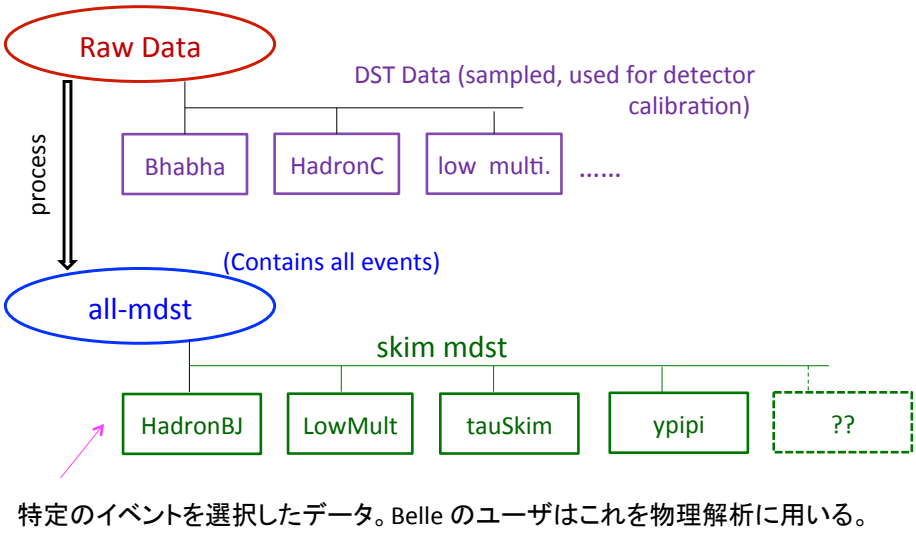

図 1: Belle の生の実験データ (raw data) は選択、処理され、目的に応じて様々な種類のファイルに分けられている。

実験で収集したデータには次のようなものがある。

- raw data: 実験で読みだした生データ
- all-mdst: raw data を reconstruction し、解析に 必要な情報のみ残したもの
- skim mdst: all-mdst から解析カテゴリーに応じて必要なイベントのみを選別したもの
- dst: キャリブレーション用に選別したデータ

この他に、以下のようなモンテカルロ (MC) シミュレーション関係のデータがある。

- MC generator: 粒子の崩壊過程のみのモンテカルロシミュレーション
- background: ランダムなトリガーで取ったデータ (raw data の一部を選別したもの)

- MC mdst: 上記の MC generator と background ファイルから検出器のシミュレーションと reconstruction を行い、generator を含む必要な情報を残したもの (データの mdst に相当する)

その他に、

- users: ユーザーのデータやファイル

がある。

物理解析に使用するのは、skim mdst と MC mdst である。

2.2.2 データの在処

上記のファイルの多くが HSM に保存されていた。それらのうち、TauSkim (タウ粒子の解析に使用) と ypipi skim(後述) の一部を除く skim mdst と 全ての MC mdst は磁気ディスク上にもあり、磁気ディスク上のファイルは全て別にコピーされた。また、MC mdst および background ファイルのすべて、および ypipi skim の一部以外の全ての skim mdst ファイルは、名古屋大学や PNNL の計算機にもコピーされていた。

2.3 B 計算機システムの移行の方針と契約

2.3.1 データの移行の方針

B 計算機のリース契約は 2006 年 3 月より 2012 年 2 月までであり、リース終了後に B 計算機の大容量記憶装置にあるデータを次の計算機に移行することになっていた。2009 年末ごろより関係者によるデータ移行のについての議論が始められ、当初案では、その移行には約 10ヶ月の期間を要し、1 年間 B 計算機のリースを延長して新中央計算機にコピーするのに 9000 万円の費用がかかると見積もられた (図 2「当初案」)[1]。2011 年 6 月に、B 計算機と旧共通計算機を統合した新中央計算機の仕様を決定する必要があったが、その数か月前ごろに計算科学センター長より、素粒子原子核研究所の意向で Belle II 実験のためにできるだけ予算を圧縮する様にとの指示があった。そこで図 2「検討案」に示すようにデータ移行開始を前倒し、B 計算機のリース延長期間を短縮することが検討されたが、B 計算機のリース会社である [redacted] の意向によりリース契約は 1 年単位でしか延長できないことが明らかになり、断念した。そこで、2011 年 6 月から 2012 年 1 月までのリース期間中に、B 計算機にあるデータを一旦 [redacted] の旧共通計算機に移動し、旧共通計算機のリース契約を数ヶ月延長して新中央計算機と平行して運用し、その間にデータを旧共通計算機から新中央計算機に移動することになった。これにより、B 計算機のリース延長の費用を削減することができ、必要なソフトウエアの開発のみに圧縮した (次節を参照)。

2.3.2 データ移行の契約

今回問題が起きたのは、B 計算機から旧共通計算機へのデータの移行である。B 計算機のレンタル契約は、リース会社の [redacted]、システムインテグレータの [redacted]

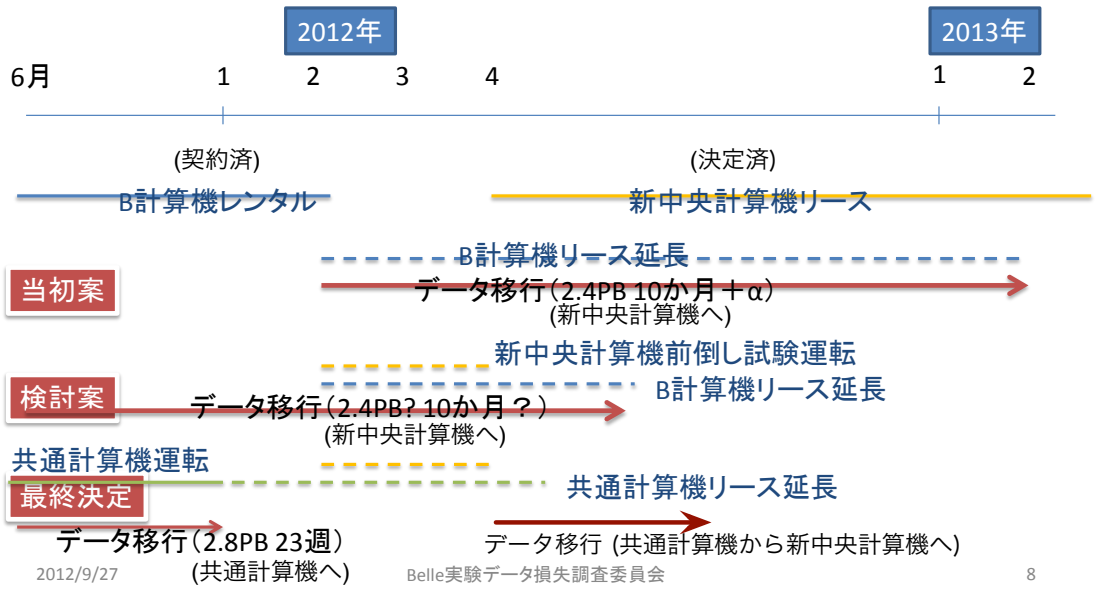

図 2: データ移行計画案の変遷。(2012 年 9 月 27 日の調査委員会で[redacted]氏が示したスライドに補足。)

[redacted]、KEK の 3 者の間で結ばれている。B 計算機の仕様書 [3] の 2.9 には、

> 又、本システムから次期導入のシステムへの移行に協力すること。大容量ストレージシステムのテープライブラリ装置に保存されるデータの後継システムへの移行に協力すること。

と書かれている。これは、移行するデータの総量や移行する先の仕様が決まらないと作業量が見積もれないため、仕様書の段階ではこのような記述にならざるを得ない。従って契約上は、B 計算機から旧共通計算機へのデータ移行作業は高エネルギー加速器研究機構が[redacted]と[redacted]の協力の下に行われた [1]。

なお、選ばれたファイルをテープライブラリから順序よく磁気ディスクに読み出し、旧共通計算機に転送し、転送が正常に行われたことを検証した後に磁気ディスク上のファイルのみを消去するソフトウェアが必要であった。この作業は常駐 SE の能力だけではできなかったため、そのソフトウェア (dmove) の開発の契約を KEK と[redacted]の間で結んだ [1]。次に述べるように、この dmove 自身は正しく働き、今回のデータ損失には関係なかった。

2.4 Belle のデータの移動作業

2.4.1 データの移動作業の概略

HSM からのデータコピーは、随時打ち合わせをしながら行われた。打合せの参加者は、主に計算科学センター側の作業の担当責任者の[redacted]氏、Belle 側の担当者[redacted]氏、作業を行った[redacted]([redacted])の SE 二名と、旧共通計算機の SE である。

B 計算機のリース期間内にデータをまず旧共通計算機に移動するために、HSM 内の全てのデータを移動するのではなく、磁気ディスク上にも存在する skim mdst や MC mdst などを除いた、必要な directory 内のファイルのみを移動することになった。上記の移行データ量の縮小と実データを用いて計測した転送速度に基いて計算した結果、当初推定の 10ヶ月の移行期間を 23 週に短縮できることがわかった。移動すべき directory の一覧は Belle 実験の[redacted]氏が用意し、[redacted] に渡した。User ファイルや一部のカテゴリーについては、subdirectories の中のファイルも全てコピーするように指示された。それ以外のほとんどのファイルについてはコピーすべき directory が明示的に与えられ、指定した directory の中の階層のファイルのみをコピーし、subdirectories は無視するように指示された。

[redacted] の下で HSM の導入と保守を行っていたのは[redacted] ([redacted]) であり、[redacted] の名刺を持つ[redacted]の SE ２名が実際にデータの移動に携わった。[redacted] ([redacted]) は受け取った directory のリスト [4] と、HSM システム上の全ファイルのリスト [5] を元に、partition ごとにコピーすべきファイルのリスト [6] を作成した [7]。このリストを元に、テープごとに 8 MBytes 以上のファイルをディスクに戻し、旧共通計算機にコピーした。旧共通計算機側では [redacted] の SE がファイルの損傷やサイズをチェックした。8 MBytes 以下のファイルはディスク上に保管されていたため、別途コピーされた。

データは 2012 年 4 月から 7 月にかけて、旧共通計算機から新中央計算機に移動された。Belle のユーザーが新中央計算機でデータを読め始められようになり一部のデータ損失が見つかったのは 2012 年 5 月以降であり、raw data の消失が判明したのは移行が完了した 7 月である。

2.4.2 データの移動作業手順

以下、今回のデータ損失に関連する作業を、[redacted] の報告書 [7] と SE の人へのインタビュー [13] をもとに再構築する。ただし、スクリプト自身は残っていないので、あくまで SE の人の記憶に頼った推定である。

1. HSM システム、およびスクリプトの開発と実行が行われた計算機は異なるが、いずれも OS は SunOS 5.9 であった [13]。また、スクリプトは全て shell script で書かれた。以下に示すコマンドは、Belle 実験が持っている HSM のシステムのノード (SunOS 5.9) の上で動きを確かめている。

2. HSM システムの partition ごとに全てのファイルのリスト（以下、`filelist` と呼ぶ）を作る。これは、`20121027_HSM_filelist` で与えられたファイルと同じである。
例：

```
drwxr-xr-x   2 21285    21285       8192 Sep 30 17:11 /h031/dstprod/dat/e000055/Rawdata/0000/energy/02
-rw-r--r--   1 21285    21285       3640 Oct  6  2006 /h031/dstprod/dat/e000055/Rawdata/0000/energy/02/md5.out
-rw-r--r--   1 21285    21285    2048000000 Oct  5  2006 /h031/dstprod/dat/e000055/Rawdata/0000/energy/02/e000055r000270.raw
-rw-r--r--   1 21285    21285    2048000000 Oct  5  2006 /h031/dstprod/dat/e000055/Rawdata/0000/energy/02/e000055r000270.raw-001
```

[redacted] の報告書によれば、これには `ls -lR` を用いたとのことであるが、このコマンドでは directory ごとに区切られ、かつ full path は表示されない。しかし、たとえば `find /h031 | xargs ls -ld` あるいは `ls -ld $(find /h031)` を用いれば、上に示したような出力は得られる。

3. この filelist には directory も含まれるため、grep -v ^d を用いて行の先頭が “d” でない行のみを選んで directory を排除し、awk を用いてファイル名の部分のみを書き出したファイル (以下、shortFilelist と呼ぶ) を作った。

4. 2011 年 6 月にコピー作業を開始した HSM サーバーについては、コピーすべき directory のリスト (以下 directoryList と呼ぶ) から 1 行ずつ sed -n ${num}p directoryList で$num を一つずつ増やしながらコピーすべき directory を選び、その directory の文字列が入ったファイルを shortFilelist から抽出する。例えば次のような方法を用いたと考えられる。

```
dir=‘sed -n ${num}p directoryList‘
egrep "^$dir" h031.shortfilelist >x.list
```

ただし、これでは探した directory の下の subdirectory の中のファイルも抽出されてしまった。従って、手作業によってそれらの余分なファイルはリストから外した [13]。

5. 2011 年 9 月にコピー作業を開始した HSM サーバーについては、7 月半ばに改良したスクリプトを用いて、コピーすべきファイルを抽出した。新しいスクリプトでは、shortFilelist からも 1 行ずつ sed でファイル名を取り出しては dirname でその directory の部分を切り出し、それを探している directory と比較して完全に一致していればそのファイル名を書き出した。例えば次のような方法を用いていたと考えられる。

```
fullname=‘sed -n ${fileno}p shortFileList‘
filepath=‘dirname ${fullname}‘
if [ $filepath = $dir ] ; then
     print $fullname’\n’
fi
```

6. 上の 4 あるいは 5 の方法で選び出した、コピーすべきファイルのリストは 20120210_HSM_filelist に入っている。

7. その後コピーすべきファイルのリストをを HSM のテープごとに切り分け、テープ内の順に並べなおして、順次それらのファイルをハードディスクに呼び出しては旧共通計算機にコピーを行った。

2.5 Belle のデータ損失

2.5.1 ファイルが失われた段階

上記の 4 あるいは 5 で選び出されたファイルは、全て正しく旧共通計算機にコピーされた [7]。しかし、[redacted]氏の指定した directory 内のファイルであるにもかかわらず、コピーすべきファイルのリストに入っていないファイルがあった。従って、2.4.2 節に示した 3 から 5 のまでのどこかで間違いが発生した。

2.5.2 コピーされなかったファイルの比率

コピーすべき directory のリスト [4]、全てのファイルのリスト [5]、コピーするように選ばれたファイルのリスト [6] を山中が解析して調べた結果を表 1 に示す。指定された 8 MBytes 以上のファイルのうち、容量で 29%、ファイルの個数で 62%がコピーされなかった。容量で見ると raw data では 18%がコピーされておらず、これは[redacted]氏の報告書 [2] の、積分ルミノシティで 18%失われた、という記述と一致する。また、low multi. と呼ばれる較正用のデータは容量で 40%、その他の dst, all-mdst 及び skim mdst ファイル (表 1 の「DST」) は 33%がコピーされていない。MC generator、background は全てのファイルがコピーされていない。

表 1: コピーされなかったファイルの容量と個数でみた比率。users と subdirs は subdirectories 内のファイルも全てコピーするように指定された。

種類	コピーされなかった ファイルの容量比	コピーされなかった ファイルの個数比
全て	530 TB/1820 TB = 29%	203 万/330 万 = 62%
raw data	180 TB/1010 TB = 18%	7.7 万/24.7 万 = 31%
low multi. (dst のうち、low multi のみ)	8.11 TB/20.4 TB = 40%	0.92 万/2.30 万 = 40%
DST (all-mdst, skim-mdst, 上の low multi 以外の dst)	98.8 TB/301 TB = 33%	26 万/62 万 = 42%
MC generator	22.6 TB/22.6 TB = 100%	5.0 万/5.0 万 = 100%
background	0.85 TB/0.85 TB = 100%	1.2 万/1.2 万 = 100%
users	206 TB/448TB = 46%	151 万/220 万 = 68%
subdirs	11.4 TB/13.4 TB = 85%	11 万/15 万 = 76%

2.5.3 コピーされなかったファイルのパターン

Belle の raw data は、実験 43 以前は 1 つの run のデータが 1 つのファイルとして保存されていたが、実験 45 以降は、2 GBytes のファイルに分割されて保存された。分割された最初のファイルには `.raw` という拡張子が付き、それ以降は `.raw-001`, `.raw-002`, ... というように枝番がつけられた。その他の `.dst`, `.mdst` 等の拡張子についても同じように枝番がつけられた。

コピーされなかったファイルには次のようにいくつかパターンがある。

- **枝番付きファイル**
  先頭の
  `/h031/dstprod/dat/e000055/Rawdata/0000/energy/02/e000055r000270.raw`
  はコピーされているが、

/h031/dstprod/dat/e000055/Rawdata/0000/energy/02/e000055r000270.raw-001
のように、枝番がついたファイルがコピーされていない場合。([redacted]氏の報告書 [2] のパターン B)

- **先頭と枝番付きファイル**
  /h016/dstprod/dat/e000055/Rawdata/0000/energy/00/e000055r000023.raw
  /h016/dstprod/dat/e000055/Rawdata/0000/energy/00/e000055r000023.raw-001
  のように、先頭のファイルには "–" が入っていないのに、先頭のファイルもそれに続く枝番のついたファイルもコピーも共にコピーされていない場合。([redacted]氏の報告書のパターン A)

- **特殊記号付きファイル**
  /h081/subdetectors/svd/harat/SbtEvtGen/gsimonly-jpsiks1_c10000.bbs や
  /h081/subdetectors/trg/nkzw/gdltiming/DAS/#das_e31r301.log#
  のように、"–" や "#" の記号が名前に入ったファイル。

- **一貫性はないが、"–" 記号のついたファイル**
  /h052/dstprod/dat/e000073/LowMult/0127/5S_scan/09/LowMult-e000073r000913-b20090127_0910.dst
  のように、"–" のついたファイル。しかし同じ partition 内の同様の名前の
  /h052/dstprod/dat/e000073/LowMult/0127/5S_scan/07/LowMult-e000073r000796-b20090127_0910.mdst
  /h052/dstprod/dat/e000073/LowMult/0127/5S_scan/07/LowMult-e000073r000796-b20090127_0910.mdst-001
  はコピーされている。

- **理由不明なファイル**
  /h031/dstprod/dat/e000043/Rawdata/e000043r000512
  のように、"–" 記号も入っていないファイル。しかし、同じ partition 内の同様の名前の
  /h031/dstprod/dat/e000043/Rawdata/e000043r000501
  はコピーされている。

- **partition 抜け**
  h046，h086，h096 の 3 つの partition については、コピーするファイルのリストさえもない。これは、コピーを指定された directory に入っていた全てのファイルの名前に "–" 記号が入っていたためである。
  (例: /h046/dstprod/dat/.../0529/Background-e000049r000400-b20060529_2127.bbs)

図 3 に、partition ごとのコピーされた／されなかったファイル名のパターンを示す。partition の順序は、表 2 に示した、サーバーごとのコピー開始日順である。2011 年 6 月に移行が始まった bhsm01, 02, 05, 06, 10, 11 の 6 つのサーバーについては raw data はほぼコピーされているが、Low Mult. のファイルの一部はコピーされていない。また、名前に "–" 記号のついたファイルもほぼコピーされている。ただし、パターンは一貫しておらず、partition によっても異なる。

それに対し、2011 年 9 月以降に移行が始まった partition では、"–001" などの枝番のついたファイルや、それ以外にも "–" 記号のついたファイルは一切コピーされていない。この

表 2: HSM サーバーごとの、コピーの開始された日 (2011 年)。

コピー開始日	HSM サーバー
6/3	bhsm02
6/6 頃	hhsm01, 05, 06, 10, 11
9/2	bhsm08
9/30	bhsm03, 07
10/7	bhsm09
10/25	bhsm21
10/28	bhsm04

ために、これらの partition に入っていた枝番付きの raw data などは全てコピーされていない。しかし 9 月以降に移行が始まった partition に含まれていた raw data の容量は 0.28 PBytes で、これは全 raw data の 28%と低い。このために、半数の partitons のコピーが完全に失敗しているにも関わらず、raw data の損失は 18%に抑えられた。

2.6 データの復旧と最終的な消失

まず、HSM に使われた磁気テープは B 計算機の運用停止後、4 月 20 日までに全て消去された [8] ため、磁気テープからのデータ復旧は不可能である。

以下、2.2.1 に示したカテゴリーごとに、データの復旧と消失の見通しをまとめる。

- **raw data と all-mdst**
  all-mdst ファイルは、raw data から reconstruction プログラムで処理することにより作れる。また物理解析で使う skim dst は all-mdst から作れるので、raw data が消失した場合には raw data と同等の役割を果す。

  Belle の実験番号 45~53, 71, 73 の raw data は Belle グループ所有の別の HSM にも保存されており、それらのデータがほぼ復旧された [9]。この復旧により、raw data の損失は積分ルミノシティで 12%に下がる。

  all-mdst も raw data も失われたのは積分ルミノシティで 7%であったが、Belle 所有の HSM から raw data が復旧されたため 5%となる [2]。

- **skim mdst と MC mdst**
  物理解析で使用する skim dst および MC mdst は、磁気ディスク上にあり別途コピーされた。一部コピーされなかったものがあるが、名古屋大学や PNNL の計算機にコピーされていたため、次の例外を除いて損失はなかった。

  例外は、ypipi skim と呼ばれる特殊な物理解析に用いられるファイルである。この一部が、他の機関の計算機にもコピーがなっかたため失われた。さらにスキャンデータの 25 のエネルギー点のうちの 6 点および $\Upsilon(5S)$ でのデータについては対応する元の raw data も all-mdst も消失した部分があり、復旧できずに損失した。損失し

part	Lost .raw	-001	Lost LM	-001	Copied .raw	-001	Copied LM	-001	Lost -	Copied -
h021	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h022	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h023	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h024	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h025	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h026	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
h011	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h012	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h013	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h014	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h015	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h016	TRUE	TRUE			TRUE	TRUE			TRUE	TRUE
h051	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h052	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h053	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h054	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h055	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h056	FALSE	FALSE			TRUE	TRUE			FALSE	TRUE
h061	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
h062	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h063	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h064	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h065	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h066	FALSE	FALSE			TRUE	TRUE			FALSE	TRUE
h101	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h102	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h103	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h104	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
h105	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h106	FALSE	FALSE			TRUE	TRUE			FALSE	TRUE
h111	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
h112	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h113	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
h114	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
h115	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
h116	FALSE	FALSE			TRUE	TRUE			FALSE	TRUE
h081									TRUE	FALSE
h082									TRUE	FALSE
h083									TRUE	FALSE
h084									TRUE	FALSE
h085									TRUE	FALSE
h086									TRUE	FALSE
h031	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h032	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h033	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h034	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h035	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h036	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h071	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h072	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h073	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h074	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h075	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h076	FALSE	TRUE			TRUE	FALSE			TRUE	FALSE
h091	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h092	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h093	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h094	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h095	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h096									TRUE	FALSE
h211	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h212	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h213	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h214	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h215	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
h216	FALSE				TRUE				TRUE	FALSE
h041	FALSE				FALSE				TRUE	FALSE
h042									TRUE	FALSE
h043									TRUE	FALSE
h044									TRUE	FALSE
h045									TRUE	FALSE
h046									TRUE	FALSE

図 3: Partition ごとの、コピーされなかった／されたファイル名のパターン。青色は正常 (コピーされるべきものがコピーされ、失なわれていない)、赤色は失敗 (コピーされるべきものがコピーされていない) の状態を表す。白色は、調べた対象のファイルが元々無かったために正常か失敗かの判別がつかなかったことを示す。Lost .raw は raw data の先頭のファイルが失われたか (True は失われた、False は 1 個も失われていない)、その横の-001 は raw data の枝番付きのファイルで失われたものがあるかを示す。その右の Lost LM は LowMult というカテゴリーのファイルの失われ方を示す。Copied .raw は raw data で先頭のファイルがコピーされたか (True はコピーされた、False は 1 個もコピーされていない) を示す。以下同様。

たのは、6 点のそれぞれで平均 30%ずつ、$\Upsilon(5S)$ のデータの 2.5%である。統計精度が約 4%落ちるが、ypipi skim のスキャンデータを使った解析で結果を出すことはできる。

- **dst**
  dst ファイルはキャリブレーション用のものであり元々 raw data を間引きして作られたものである。物理解析には使用しないが、必要に応じてチェックのため使用する。消失した分は、必要に応じて raw data から作ることができる。

- **MC generator**
  MC genarator ファイルは、MC mdst を作り直す必要が生じた場合に使うため保存してあったが、その情報は MC mdst に含まれており、必要な時に作り直すことが可能である。

- **background**
  background ファイルは、他の機関の計算機にコピーがあり、損失はなかった。

- **8 MBytes 以下のファイル**
  8 MBytes 以下のファイルは別途、ディスクから圧縮されて移行された。しかし、`/bhsm/h041` の中のユーザー用以外の 8 MBytes 以下のファイルは全く移行されていない。ここに Belle のデータのプロダクションのスクリプトとログファイルがあり、それらは全滅した。今後の all-mdst の復旧のプロダクションで前のログファイルと直接比較確認できないなど、手間と作業時間に対する影響は大きい [2][1]。

- **users**
  ユーザーのデータやファイルも大量に失われたが、この復旧の可能性や、実質的な損失は各ユーザーに依るため、わからない。

dst, MC generator, background に関してはコピーによる消失はあったが、データとしての実際の損失はない。但し、復旧の手間と時間が必要である。

表 3 にデータの実質的な損失についてまとめる。今回のデータ損失が Belle の解析全体に与える影響は比較的小さい [11]。

3 データ損失の原因とその背景の分析

データ損失の直接の原因は [redacted] ([redacted]) がコピーする directory のリストから作成したファイルのリストに間違いがあり、抜けおちたファイルがあったためである。それに気付かずにコピーを行ったため、最終的にファイルを消失する事態となった。この過程に十分なチェックが行なわれるような責任体制をとっていれば防ぐことができたであろうが、Belle グループ、素核研および計算科学センターのマネージメントのデータ移行プロジェクトに対する体制・意識などの問題により、十分な処置がされず事故に至った。

[1] 前のログファイルがない場合は、最初のプロダクションの時に行ったすべての確認作業を行うことになり、大きな手間と作業時間が必要となる。Belle では、今後に備えてデータのコピーを他の機関の計算機に置くため、all-mdst の復旧を計画している。

表 3: データの実質的な損失のまとめ。括弧内は Belle 所有の別の HSM から raw data を復旧する前。

種類	損失量	
raw data	12%	(18%)
raw data と all-mdst 両方	5%	(7%)
skim mdst (ypipi 以外)	0%	
ypipi mdst (scan; 25 点中 6 点で)	30%	
ypipi mdst ($\Upsilon(5S)$ データ)	2.5%	

以下にその詳細を、データ損失の原因を技術的な面とデータ移行の体制の面から分析し、またそれらの問題を起こした背景を考察する。

3.1 技術的な面

1. SE の人へのインタビューによれば、コピーすべきファイルを選ぶ段階で複雑な正規表現などは使っていないようである。`awk` を用いたファイル名の切り出しも、空白で区切られた文字列を選んでいるだけなので、文字列の中に "–" が入っているかどうかは関係ない。また、"–" が名前に入ったファイルが一律コピーされていないのは、9 月以降にコピーを開始した HSM のサーバーのファイルである。これらは 7 月半ばに改良したスクリプトを用いてファイルを選びだしており、つけ加えられた機能は、`dirname` を用いてファイルの置かれた directory を切り出して探している directory と比較している部分である。従って、`dirname` コマンドが正しく動いていなかった可能性がある。`dirname` が誤った文字列を返すと、これは探している directory と異なると判断され、そのファイルはコピーしないことになる。

   `dirname` に関して文献 [14] には、次のような記述がある。

   > *dirname* and *basename* have a bug in many System V implementations. They don't recognize the second argument as a file name suffix to strip. Here is a good test:
   > `% basename 0.foo .foo`
   > If the answer is `0`, your *basename* implementation is good. If the answer is `0.foo`, the implementation is bad. If *basename* doesn't work, *dirname* won't, either.

   Belle 実験が持つ HSM システムの Solaris SunOS 5.9 の上で上の例を試すと、`0.foo` が返され、`basename` が正しく動いていない。ただし、データ移行されなかったファイル名について `dirname` を実行しても、結果は正しく返される。

   同じ OS のバージョンではあるが、[redacted] がつくば営業所でスクリプトを走らせていた計算機の `dirname` にバグがあった可能性は残る。

2. 6 月からコピーした partition の間違い方は一貫しておらず、これは最初に作った (subdirectory 入りの) ファイルのリストを手作業で直したという話と符合する。それに対して、9 月以降にコピーした partition の間違い方は一貫しており、6 月からコピーした partition の間違い方とも異なる。これは、7 月半ばに固定した、改良されたスクリプトを用いたという話と合う。

   ただしそれならば、新たなスクリプトの出力結果の確認がおろそかであった。多くのファイル名に “–001, –002, . . . ” のような枝番が付いているのに対して、スクリプトの出力結果にそれらの枝番がついていないのは、出力ファイルを少し見れば明らかである。出力ファイルは[redacted] を通してメーリングリストでも配布されたとのことであるが、誰も注意してそれらのファイルをチェックしていなかったことになる。

3. コピーすべきファイルを選び出す作業は、中間ファイルを作りながら shell script で行った [7, 13] とのことであるが、shell script で行うには複雑な作業である。より高度な Python などのスクリプトを用いれば、もっと簡単に、かつ確実に作業できる。例えば Appendix A に示した Python のスクリプトなら、実質約 30 行で全ファイルリスト [5] から指定された directory[4] 内のファイルを抽出できる。このようなより高度なスクリプト言語を用いていれば、今回のデータ損失が起きる確率を下げられたと考える。

4. HSM 内の全ファイルを移行しなかったことも、問題を起こしやすい要因であった。このためにファイルを選ぶ作業が複雑になり、さらにコピーすべきファイルの数を partition 内のファイル数と比較するなどの簡単なチェックも行いにくくなった。

   ただし、2011 年 11 月 4 日のデータ転送ミーティングで、[redacted]氏が「総転送容量が 1.27 PB となっているが、テープライブラリの総容量の半分以下である。本当に間違っていないか？」と打合せの資料 [10] で疑問を呈している。ここでより深くチェックされていれば、問題を発見できた可能性がある。

5. 実験 45 以降、データファイルを 2 GBytes ごとに区切ることになったが、各ファイルが何個に区切られたのかが Belle のデータベースに入っていなかった事も、問題の発見を遅らせる要因であった。何個のファイルがコピーされるべきか、Belle 側も認識していなかった。

   ただし、多くの場合このようなファイルは約数十個に区切られており、ファイルの数のオーダーの概算はできたはずである。

3.2 データ移行の体制

1. 約 2 PBytes の重要なデータを移行するプロジェクトとしては、人数が少ない。Belle 側で必要な directory を選んだのは[redacted]氏 1 人であり、「KEK が[redacted] と [redacted] の協力の下に行った」[1] とは言え、実質は[redacted] の (実際には[redacted]の)SE 2 名と[redacted] 側の SE によって移行作業が行われた。

2. コピーすべきリストに入ったファイルについては B 計算機と旧共通計算機側の間でチェックが行われて不具合も直されていたが、指定した directory の中のファイルの

リストを作る段階についてクロスチェックされていなかった。単純な作業でありプロに任せておけば間違いはないと考え、人間は間違いを起こすことを前提としていなかった。

3. 責任体制が曖昧であった。B 計算機の契約には「後継システムへの移行に協力すること」と書かれているだけであるため、今回の誤りに関して、実際の作業を行った会社の責任を追求できない。また、コピーすべきファイルのリストが正しいことを誰が確かめるべきだったのか (計算科学センターなのか、Belle なのか) も明らかではない。

3.3 問題を引き起こした背景

上で述べた問題が引き起こされた裏には、次のような背景がある。

1. データ移行にかけられた予算が不足していた。Belle II 実験の建設のために、データ移行にかけられる予算が圧縮された。このことは、次の二つの問題を引き起こした。
   (a) 次の 2 に示すように、B 計算機と新中央計算機の運用期間をオーバーラップさせることができなかった。
   (b) データ移行の作業を、改めて正式に契約することができず、業者に「協力」という形でしか作業を依頼できなかった。このために、責任体制が不明確になった。
2. 新中央計算機の運用と B 計算機の運用期間にオーバーラップがなかった。これは、リース会社が月単位の契約の延長に同意しなかったことも一因である。このことは、次の二つの問題の引き起こした。
   (a) データを移行している間に、ユーザーがすぐにそのデータを用いて解析を行い、調べることができなかった。(一時的にデータを移した旧共通計算機では Belle の解析の環境が整備されておらず、その上のデータをユーザーがわざわざ使うことがなかった。) このために、問題の発見が遅れた。
   (b) 問題があることがわかった時点では B 計算機のデータは既に消去されており、コピーしなおすことができなかった。
3. データ移行に与えられた期間が約 7 ヶ月と短かった。このことは、次の問題を引き起こした。
   (a) 当初、現場では B 計算機が停止してから 1 年かけてデータを移行する計画でいたが、急に短期間でデータを移動することになり、そのための開発や試験を行う時間が足りなかった。
   (b) 限られた期間内にデータを移動するために、必要な directory のみを選んでコピーすることになった。このために作業が複雑になり、今回のようなスクリプトの間違いが入り込む余地を残した。また、コピーすべき容量やファイルの数の合計などの簡単なチェックが行えなかった。

(c) 期間内にコピーを終えることが最重点課題となり、突き放した視点からのチェックがおろそかとなった。

4. 計算機関係の人不足。今回のテープデータ移行に主に携わったのが、Belle 側から 1 人、計算科学センターから 1 人のみ[2]であり、作業量と作業の大変さは認識されていたが、十分な人数を割り当てる事は難しかった。さらに担当者は他の責務もあり、十分に移行作業に目を行き届かせる事ができなかった。

4 今後の対策の指針

今回のデータ損失から我々は教訓を得て、将来同じような過ちを犯さないようにしなければならない。そのために、次のような指針を示す。

1. **重要な作業に関しては、当事者以外のチェックが必要である。**
   少し離れた視点から実験屋のセンスでチェックするだけでも、今回のような事故は防げたと考えられる。こうしたチェックは、作業の間ずっとする必要もないし、スタッフ以外の人が行ってもよい。ただし、決して「確認者の欄に印を押す」というような形式的な物にしてはいけない。

2. **プロジェクトの遂行に十分な人員と時間を充てる。**
   上とも関連するが、一人ではなく複数による作業や確認ができる体制を作ることが望ましい。また、時間に余裕を持たせてプロジェクトを計画するべきである。

3. **作業の責任を明確にすべきである。**
   多くの作業は会社との信頼関係に基づいて行われるが、そうした場合でも、どこからどこまでの作業は誰の責任で行うのか明確にすべきである。責任体制に穴のない契約を結ぶことが望ましい。ただし、責任の線引きを明確にすることによってその内側しか気をつけないようになっては逆効果である。自分の責任範囲外の事に対しても情報を交換して互いにチェックしあうことが重要である。

4. **データの保護の重要性を認識すべきである。**
   最近の考え方ではデータは人類共有の資産であり、研究所は未来に渡ってその保持に責任を負う [12]。従って、現実的な予算の範囲内で人災や天災に対してもその資産を守る手立てを Belle, 素核研、計算科学センター, ひいては機構も考慮すべきである。

5. **計算機やソフトウェアの重要性を認識し、資源を確保すべきである。**
   実験で得られたデータの保存、解析用計算機の管理、解析のためのソフトウェアやデータベースを開発などは、加速器や測定器を作って動かす事と同等に重要なことである。したがって、計算機やソフトウェアのための人材と予算を素核研と機構は責任を持って充実させるべきである。

[2]毎週の打合せにはその他に 3 名が出席し、うち一人が旧共通計算機側の作業に関わっていた。

5　最後に

今回のデータ損失は意図的な要素は全くなく純粋な事故であるが、これに似た事故は、用語を入れ替えれば他の現場でも起き得ることである。人間はどれだけ気をつけていても誤りを犯す。そのことを前提にして、少しでも誤りを防ぎ、それでいて機動性と柔軟性を失わないようにプロジェクトを進めていける体制を Belle、素核研、計算科学センター、ひいては機構も作っていくべきである。決して今回の教訓を「気をつけて作業しよう」というような標語作りに終わらせてはならない。

A　コピーすべきファイルのリストを作る Python スクリプトの例

全ファイルのリストから、指定された directory に入っているファイルのみを抜き出して書きだす Python script の例を下に示す。2.6 GHz Intel core i7 で core を 1 個だけ使っても、6 分強で 4800 万個のファイル名をスキャンして 95 万個のファイルを選び出せる。directory ではなくファイルであることは、`ls -l` で得られる`-rw-r--r--`などの mode の一文字目でチェックしている。ファイルが指定された directory に入っているかどうかは、索引の早い dictionary という Python の型を用いている。このような、より高度なスクリプト言語を用いれば、問題の発生の確率は下げられる。

```
#!/usr/bin/env python
# Usage:
#   $ bin/makelist.py h     : all partitions (with h*)
#   $ bin/makelist.py h01   : all partitions with h01*
# 2012-10-08 Taku Yamanaka

import os
import sys
import glob
import re

directoryList = {}  # directories to copy
for directoryFile in glob.glob('data_transfer/data_transfer_110812/*.list'):
    for dir in open(directoryFile):
        dir = dir[:-1]
        if dir[:5] == '/bhsm':
            dir = dir[5:]
        directoryList[dir] = 0
        # Looking up a directory is much faster than searching a list.

reSplit = re.compile('(\S+)')
kPermission = 0
kSize = 4
kName = 8

kFilesizeThreshold = 8 * 1024 * 1024

for filelist in glob.glob('20120127_HSM_filelist/' + sys.argv[1] + '*.log'):
    for fileInfo in open(filelist):
        fileItems = re.findall(reSplit, fileInfo)
        fullname = fileItems[kName]
        filesize = int(fileItems[kSize])

        if fileItems[kPermission][:1] == '-' and \
           fullname[-4:] != 'md5S' and \
           fullname[-7:] != 'md5.out' and \
           filesize >= kFilesizeThreshold:
            dir = os.path.dirname(fullname)
            if dir in directoryList:
                print fullname
```

参考資料

[1] [redacted]「B 計算機の HSM 上のファイルの移行の失敗によるデータ消去について– 契約関係を中心に –」(2012.09.23)。

[2] [redacted]「B 計算機の HSM 上のファイルの移行の失敗によるデータ消失について」(2012.09.27 改訂)。

[3] B 計算機仕様書。

[4] `data_transfer` 以下のファイル：[redacted]氏の渡した、移行すべき directory のリスト。

[5] HSM システム上の全ファイルのリストとして、`20120127_HSM_filelist` 以下のファイルが残っている。

[6] [redacted] ([redacted]) が作成した、移行の対象となったファイルのリストとして、`20120210_HSM_filelist` 以下のファイルが残っている。

[7] [redacted]「B ファクトリー実験データの損失に関する調査報告書」(2012.09.24)。

[8] [redacted]「B ファクトリー実験データの消去について」(2012.09.20)。

[9] [redacted]、2012 年 11 月 2 日 メールでの報告。

[10] [redacted]「B 計算機 → 旧共通計算機 → 新中央計算機データ転送打合せ」資料 (2011.11.04)

[11] 堺井義秀「Belle データ損失の物理解析への影響について」(2012)。

[12] DPHEP Study Group (`http://www.dphep.org`) "Status Report of the DPHEP Study Group: Towards a Global Effort for Sustainable Data Preservation in High Energy Physics", arXiv:1205.4667 (2012).

[13] 2012 年 10 月 30 日に行った SE の人へのインタビュー。

[14] J. Peek, T. O'Reilly, and M. Loukides, "Unix Power Tools", O'Reilly Associates, Inc., p.914 (1993).