特　集「イノベーションとAI研究」

サイバーエージェントのデータ活用のための R&D体制と取組み

R&D at CyberAgent for Data Utilization

福田　一郎　株式会社サイバーエージェント
Ichiro Fukuda　CyberAgent, Inc.
fukuda_ichiro@cyberagent.co.jp, http://www.cyberagent.co.jp/corporate/labo/

鈴木　俊裕　（同　上）
Toshihiro Suzuki　suzuki_toshihiro@cyberagent.co.jp

善明　晃由　（同　上）
Teruyoshi Zenmyo　zenmyo_teruyoshi@cyberagent.co.jp

高野　雅典　（同　上）
Masanori Takano　takano_masanori@cyberagent.co.jp

藤坂　祐介　（同　上）
Yusuke Fujisaka　fujisaka_yusuke@cyberagent.co.jp

Keywords: Ameba, R&D, Hadoop, Hbase, graph database, filtering, datamining.

1. データ活用のためのR&Dとその体制

サイバーエージェント・秋葉原ラボは2011年に自社サービスの向上のために大量データを利活用することを主な目的として設立された研究開発部署である．本稿では秋葉原ラボにおけるデータ活用の取組みについて紹介する（図1）．

サイバーエージェントではAmebaを中心として多くのWebサービスを運営しており，日々サービス利用者（ユーザ）のアクセスログや行動ログが大量に蓄積されている．このような大量のデータは，推薦や検索などのデータをもとにしたユーザへの機能提供や，安全なサービス維持のための不正行為の検出，ユーザの傾向や趣味嗜好の発見など，Webサービスを発展させるために非常に重要である[Fan 13]．ところが，これらのログやデータは秋葉原ラボができるまでは十分に活用されておらず，ストレージ容量を圧迫することもあり，サービスポリシーの保存期間を過ぎたものは削除されていた．それらを有効活用するために，当社では2010年3月に秋葉原ラボの前身組織を設立し，Hadoopを利用したデータウェアハウス（DWH）および内製BIツールを構築した．それによって，Webサービスから生み出される大量のログやデータの利活用を本格的に開始した．

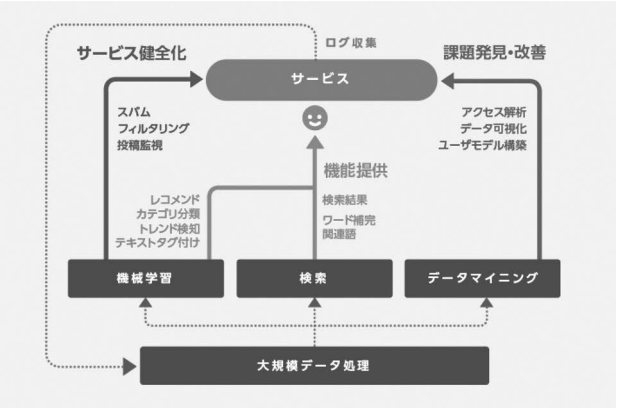


図1　データ活用に関する取組み概要

当時，大量のログデータを扱うことはデータ容量や計算リソースの面で困難であったが，Hadoopの出現により大量データの保存と分散計算による大量データの計算の両方が現実的なコストで解決できるようになり，我々のニーズと合致していた．また，当時Hadoopは普及期にさしかかったところであり，オープンソースソフトウェア（OSS）であったため，Hadoop本体やHadoopをベースとしたミドルフェア，周辺ツールなどのオープンな開発の場が非常に活発であった．そこでは，

Hadoop という新たなデータ処理方式という技術的な新たな種から，HBase や Hornet（2 章参照）といった今まで扱いにくかったデータ構造を扱いやすくする新たなデータストアが生まれ，また，Hadoop の普及に伴いデータ活用の生態系を扱うための新たなサービス，システム（例えば，Cloudera の CDH や Patriot（2 章参照））が生まれてきた．その意味で Hadoop は有望なシーズでもあったといえる．今もなお Hadoop およびその周辺では活発な開発が行われており，今後も大きな発展が期待できる．

秋葉原ラボは Hadoop の発展に沿う形で大量データの扱う範囲を広げてきた．Hadoop とその DWH である Hive は基本的にはバッチ処理向きのプロダクトであるため，データ集計・分析処理やレコメンド機能向けのデータ作成などに活用している．しかし，大量データ向けのデータストアをオンライン用途で利用したいというニーズも Web サービスをつくる際には存在する．そこで，Hadoop をバックエンドとしてオンライン用途でデータの書込み・読込みが可能な HBase の検証を行い，各種 Web サービスでの利用を進めている [鈴木 15]．さらにソーシャル系の Web サービスにおいて重要になる“ユーザ同士のつながり情報”を格納・利用するためのデータストアとして HBase を使ったグラフデータベース Hornet を開発し，ソーシャルネットワークサービスの基盤として利用している．

ログデータだけでなく，ユーザがつくり出すコンテンツ自体も大量データとなり得る．blog 記事などのテキストデータは検索できるように（転置）インデックスを構築する必要がある．秋葉原ラボではテキスト検索には Solr（Lucene）を利用しているが，Solr（Lucene）もまた Hadoop に関連するプロダクトである．

データを収集・処理する基盤ができたことで，それを効率的かつ迅速に活用することが可能となる．秋葉原ラボではサービス利便性向上のための活用としてレコメンド機能やスパム検知システムを構築している．さらに，スマートフォンゲームやコミュニティサービスのユーザ行動ログを分析し，ユーザの嗜好を把握することでサービス発展に寄与している．

2 章ではデータ活用のための処理基盤の事例として，Hadoop を利用したログ解析基盤「Patriot」と HBase をベースとしたグラフデータベース「Hornet」について，3 章ではサービス利便性向上のためのデータ活用の事例としてスパム検知システム「Orion」について，4 章ではサービス発展のためのユーザ行動データの分析事例をそれぞれ紹介していく．

2. データ活用のためのデータ処理基盤

2·1 ログ解析基盤 Patriot

Patriot は Hadoop を用いた独自のログ解析基盤で，

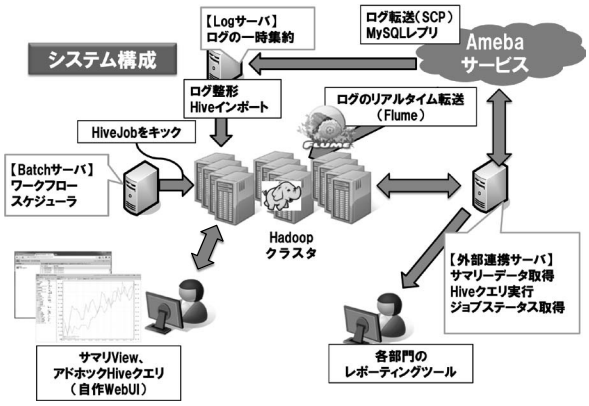


図 2　Patriot のシステム構成

アメーバ blog やアメーバピグなどさまざまなサービスのログを集約し，ユーザの行動分析やアクセスログの集計，レコメンドシステムへの応用などを行っている．Patriot のシステム構成を図 2 に示す．各サービスから収集されたログは HDFS に蓄積され，主に Hive を用いて処理した結果を HBase に格納している．HBase 上のデータは各部門向けのレポートやサービスでのレコメンド機能などに活用されている．Patriot のようなオンプレミスのログ解析基盤はコスト部門であり限られた人的，計算機リソースのもとで効果を最大化することが求められる．また，Web サービスのビジネスモデルの移り変わりは激しいため，内製化によって柔軟に複雑な要件に対応する必要がある．我々はこれらの課題に対して以下の取組みを行っている．

- ●独自のワークフロースケジューラの開発
- ●バッチ設定解析ツールの開発
- ●Github Enterprise（GHE）を用いた運用フローの整備

本章ではこれらの取組みについて説明する．

2015 年 1 月現在，Patriot では 1 日当たり約 7 000 個のジョブを管理している．これらのジョブは複数のサービスのデータを利用するものや長い期間の行動ログを参照するものなど複雑な依存関係でつながっている．このようなジョブを効率的に実行するためには，依存関係を細かく管理する必要がある．依存関係を荒く設定してしまうとログ取込みの失敗などによりそのログと関連のないジョブが遅延したり，むだな待ち時間が発生し Hadoop クラスタの利用効率を下げてしまう．

開発したワークフロースケジューラは，データフローに基づき依存関係を管理することで，複雑な依存関係を効率的に管理する．

依存関係の例を図 3 に示す．各ジョブは参照するプロダクトを生成するジョブがすべて終了した場合に実行可能になるものとする．例えば，サービス A の UU 集計はサービス A のログ取込みが終了した後に，全体 UU 集積はサービス A ログ取込みとサービス B ログ取込みの両方が終了した後に実行可能となる．図 3 では，サー

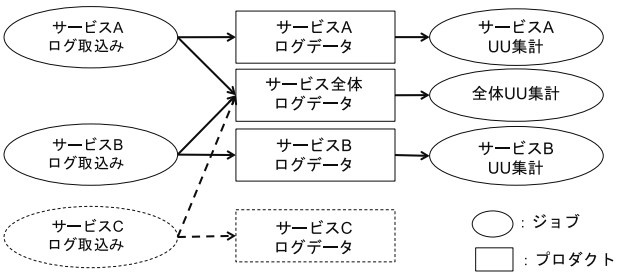


図3 依存関係のモデル

ビス全体ログデータという仮想的なプロダクトを設定している．これにより，例えば新たにサービスCが追加された場合も，サービスCのログ取込みジョブがサービス全体ログデータを生成するように設定することで，サービス全体の集計設定には影響を与えないようにできる．また，ジョブの設定は独自のDSLを用いて簡潔に記述できるようにしている [善明 13]．

多くのジョブを実行するうえではジョブの量だけでなく質も問題となる．例えば，非効率なジョブが一つでもあるとそれによりHadoopクラスタのリソースが占有され全体の処理を遅延させる可能性がある．Patriotでは，ルールベースのバッチ設定解析ツールを開発しこの問題に対応している．このツールはHiveクエリを効率的に記述するためのノウハウをルールとして設定し，非効率なクエリの検出や改善方法の提示を行う [善明 13]．バッチ設定の解析では，単一クエリの解析と複数クエリの解析を行う．単一のクエリの解析としては，入力範囲が十分にしぼられていないクエリの特定や非効率なJOINを行っている箇所（ONが指定されていないなど）の特定を行う．これにより，中間データの作成を検討したほうがよい部分やクエリのミスを検出できる．また，複数クエリの解析ではクエリの統合による最適化可能性の判定を行う．一般に同じデータを参照するクエリは，データの入力部分を共通化することで処理を効率化できる．例えばblogの記事投稿数と記事投稿ユーザ数は集約関数が異なるのみで同じデータを入力とするため，まとめて実行することで効率化できる．バッチ設定解析ツールはクエリを入力ごとにグループ化し，各グループに対して最適化ルールの適用可能性を判定する．これにより，似たようなクエリが量産されるのを回避しバッチ処理全体の効率化につなげることができる．また，重複した記述が減るため，クエリの統合はバッチの保守の観点からも有効である．

これにより，Hadoop/Hiveについて深い知識をもたない技術者でもある程度の品質のバッチ設定を記述することを可能としている．

Patriotでは，これらのスケージューラやバッチ設定検査ツールを用いて，他の部門でもPatriotに蓄積されたデータを柔軟に活用できる環境をGHE上に構築している．Patriotの運用体制を図4に示す．バッチ設定はGHEで管理され，Pull Requestによりさまざまな部門から追加，修正などが行えるようになっている．Pull Requestはバッチ設定効率化ツールにより自動的に検査され，効率化が可能な場合はその方法が提示される．また，バッチ設定は自動的にワークフロースケジューラに配置され随時実行される．

このように秋葉原ラボでは，ログ解析基盤に必要となる要素技術の開発からそれらを実際に運用に組み込むことまでを一貫して行っている．また，現在は，さらなる改善を目指してデータの標準化やKPI設計からログの生成までを一貫して行う手法の確立などに取り組んでいる．

2·2 Hornet

Hornetは当社で開発したHBaseをバックエンドとしたグラフデータベースである．グラフデータベースとは，グラフ構造を格納することに最適化されたデータベースのことである [Rodriguez 10]．

当社は，2012年6月に「デカグラフ [本宮 12]」と名付けた構想を掲げ，スマートフォンプラットフォームをリリースした．デカグラフ構想とは，各サービスのユーザ層を「ミニグラフ」と名付け，ユーザに共通のIDで自由に回遊してもらうことで，一つの巨大なスマートフォン向けサービスを構築するというものである．この構想を実現するためには，それらのグラフ構造を格納するためのデータベースが必要となり，Hornetの開発に着手した．

スマートフォンプラットフォームはオンラインサービスであるため，ユーザからのアクセスに対してインタラクティブにレスポンスを返す必要がある．また，プラットフォームがダウンすると，それを使用しているすべて

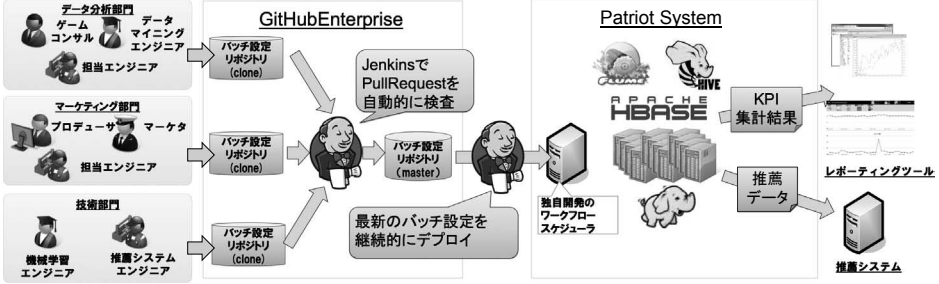


図4 Patriotの運用体制

のサービスがダウンしてしまう可能性がある．さらに，データは日々蓄積され，Web スケールのアクセスに対応することが必要である．そのため，Hornet は低レイテンシであること，高可用性をもつこと，スケール可能であることの三つの非機能要件が求められる．

当社では，このような要件に対して，複数の MySQL のプロセスを起動し，データを水平分割し分散配置することで対処することが慣例である．しかし，MySQL は分散処理を考慮した設計となっていないため，データの分割や分散配置に関するオペレーションやサーバがダウンした場合の処理などは，外部からツールなどを用いることにより解決しなければならない．これは，運用を煩雑にし，想定外の事象が起きた場合に場当たり的な対応にならざるを得ない．

そこで Hornet では，データの分割や分散配置を考慮した設計となっている HBase を採用することで，運用を煩雑にすることなく可用性やスケーラビリティに対する問題を解決した．これにより，大規模なグラフ構造を低運用コストで扱えるようになった．グラフ構造は現在の Web サービスの主流の一つであるソーシャルネットワークサービス（例えば Facebook, Twitter）やソーシャルゲームといったソーシャルグラフを扱う場合に必ず現れるデータ構造であり，そのような大規模なソーシャルグラフを扱いたい場合に，Hornet は有効な解決策であるといえる．本章では，Hornet が HBase を採用した経緯や，Hornet について説明する．

§1 HBase を採用した経緯

HBase は Google 社が開発した分散データベース「Bigtable」のオープンソースクローンであり，一般的に Hadoop の分散ファイルシステムである HDFS 上で動作する．いわゆるマスタ型と呼ばれるシステム構成となっており，Master と RegionServer というプロセスの 2 種類のプロセスが存在する．Master はメタデータの管理や RegionServer のコーディネーションなどを行い，RegionServer が実際にクライアントとデータのやり取りを行う．また，HBase は自動シャーディングと呼ばれる自動的にデータを水平分割する機構をもっており，各 RegionServer 上に分割されたデータが配置される．これにより，RegionServer を増設することで処理能力を増加させることが可能となっている．Master はホットスタンバイ構成を取ることができるので，ダウンした場合にはスタンバイがアクティブになり，サービスを継続することができる．RegionServer がダウンした場合も，自動フェイルオーバ機能により，自動的にデータの再配置が発生しサービスを継続することができる．

これらの特徴により，HBase を用いることで，Hornet はスケーラビリティや可用性を得ることができる．さらに，HBase はレンジスキャンや Read-Modify-Write 操作などグラフデータベースを構築するうえで必要な API をもっており，またすでにログ解析などで Hadoop を使っていた経緯もあり，Hornet で HBase を採用することに決めた．

ただし，Hornet の開発に着手した時点で HBase は安定しているとはいいがたい状況だった．そのため，Hornet の開発は，HBase のオープンソースコミュニティとやり取りをしながら進めていった．また，Hornet プロジェクトチームでは HBase のソースコードを解析し，バグなどがあった場合は修正パッチを当てて使用している．

§2 Hornet について

Hornet のシステム構成図を図 5 に示す．Hornet は，HBase の前方に Gateway を立てる構成となっている．Gateway は，クライアントからのグラフ構造に対するリクエストを受け，HBase にアクセスし，クライアントに結果を返す．また，メタデータの保存や Gateway 間のコーディネーションは Zookeeper を用いて行っている．

Hornet のデータモデルは，プロパティグラフ [Rodriguez 10] と呼ばれる構造になっている（図 6）．プロパティグラフとは，ノード（頂点），リレーションシップ（エッジ）が存在し，それぞれにキー・値型のプロパティ（属性）をもつことができる．リレーションシップはタイプと呼ばれるラベルを付けることができる．プロパティグラフは有向グラフなので，リレーションシップは，起点ノードから終点ノードへの方向をもつ．

Hornet のクライアント API を以下に示す．本 API は Java 言語で実装されている．この例では，ノードを二つ作成し，それらのリレーションシップを作成するものとなっている．また，ノードやリレーションシップを作

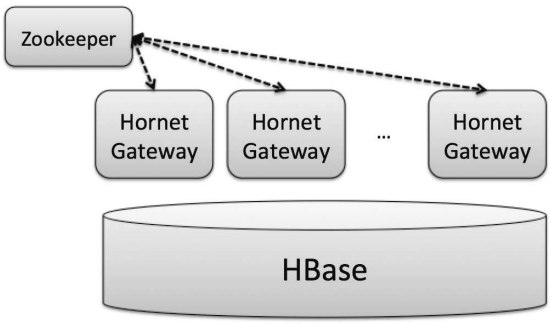


図 5 Hornet のシステム構成

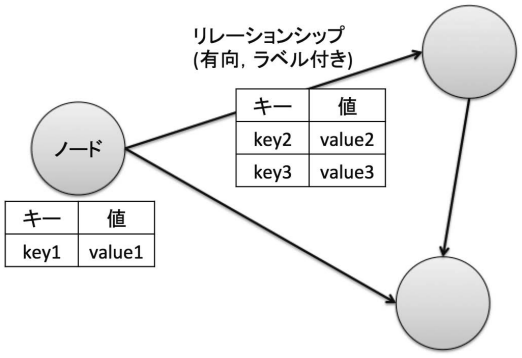


図 6 プロパティグラフ

成後に，プロパティを設定している．

```
Graph g = ...
Node node1 = g.addNode();
node1.setProperty("name", valueOf("node1"));
Node node2 = g.addNode();
node2.setProperty("name", valueOf("node2"));
Relationship rel =
node1.addRelationship("follow", node2);
rel.setProperty("name",
  valueOf("2015-02-19"));
```

2015 年 1 月現在 Hornet には，約 120 億のノードと約 650 億のリレーションシップが格納されている．また，約分間 600 万のアクセス数を受けており，レイテンシは平均 10 ms 以下で返している．

現在，Hornet は OSS 化に向けてソースコードやドキュメントの整備を進めている．OSS 化により，Hornet の開発の促進や，利用実績を公開することによる HBase オープンソースコミュニティへの貢献ができると考えている．

3. サービス利便性向上のためのデータ活用

Ameba サービスの健全な発展のための取組みとして，データを活用したユーザのカスタマエクスペリエンス向上に取り組んでいる．本章ではそれの例として，以下ではスパム行為*1 を行うユーザを検知するシステムについて述べる．

当社は「アメーバ blog」や「755」を始めとした多くのコミュニティサービスを提供しており，また中学生，高校生を始めとした若年層の利用者も多いため，それらをターゲットにした非健全な目的での個人情報の交換を行う温床となり得る危険性があった．また，アフィリエイトを目的とした大量の投稿により，サービスに対する印象が悪化する可能性が懸念されるようになった．

それらの問題を解決するために，以下の機能をもったスパム検知システムを開発・運用している．

- ●当社サービスに対して統一したスパム検知・削除サービスを提供できる
- ●国内最大規模の投稿量に対応し，かつ新規サービスに容易に導入できる
- ●サービスそれぞれの特性に合わせた検知条件を設定できる
- ●機械学習などを生かした高度な判別システムを提供できる

図 7 に，現行のスパム検知システム“Orion”のシステム構成を示す．同図のように，Orion は各サービスの投稿データやユーザからの通報データを受け取り，機械的なフィルタリングと有人監視を組み合わせてスパム行為を検知する機能を実現している．これによって素早く正確な対応を可能にしている．サービスやサービスの投稿箇所ごとに対象ユーザや投稿の性質が異なるため，フィルタの設定はサービスが定めた投稿箇所ごとにフィルタワードや条件などを指定したり，各サービスは事前の投稿チェックとしてフィルタのみを利用するなど柔軟な利用が可能である．フィルタはワード検出や連続投稿の検出に加え，過去数年に及ぶ blog 投稿と判別結果を学習データとした SVM（Support Vector Machine）判別器も提供しており，より効率的なスパム検出に貢献している [數見 14].

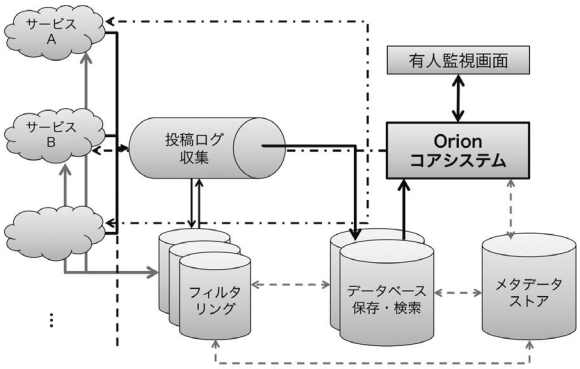


図 7　Orion のシステム構成

以上のシステムを軸として，より良いカスタマエクスペリエンスのために，以下の機能の実用化に取り組んでいる．

- ●有人監視の判別結果をオンラインでフィードバックする仕組みをつくる
- ●ユーザの行動をプロファイリングし，未然にスパム行為を抑止する
- ●機械学習によるスパム判別を画像や音声，動画データに拡大する

4. サービス発展のためのユーザ行動データの活用

本章ではスマートフォンゲーム「ボーイフレンド（仮）」におけるキャラクタカードの人気推定について紹介する [和田 14]．ボーイフレンド（仮）とは，当社で開発・運用されているスマートフォンゲームである．ボーイフレンド（仮）はガールフレンド（仮）*2 やアイドルマスターシンデレラガールズ*3 のように，1）同じキャラクタのカードが複数のレアリティ*4 モチーフでリリースされて

*1 本章におけるスパム行為とは，アダルトコンテンツの投稿，誹謗中傷，連絡先の交換などといった当社が定めた利用規約に反する行為を指す．

*2 http://vcard.ameba.jp
*3 http://cinderella.idolmaster.jp
*4 カードのグレード．対象のゲームではグレードの高い順に SSR，SR，HR，R，HN，N の 6 段階があり，グレードが高いほどゲームにおけるカードの能力値は高く，また，入手が困難になる．

おり，2）それぞれのキャラクタの個性やユーザとキャラクタの関係をゲームの設定として埋め込んでいる．そのため，ユーザもそういった部分に魅力を感じゲームをプレイしていると思われる．したがって，ゲームそのものの面白さやレベルデザインだけではなく，キャラクタやカードの人気を踏まえたカードの運用はユーザの満足度を高めるゲーム運営していくうえで非常に重要である．

キャラクタの人気度を測るための一般的な方法は，アンケート調査や AKB48 [松尾 15] に代表されるような人気投票を行うことなどが考えられる．しかし，これらの方法では次のような問題が存在する．調査対象のユーザにアンケートや人気投票に参加するユーザというバイアスが存在する．参加者は回答・投票意欲があることから比較的熱心にゲームをプレイしているユーザだと思われる．スマートフォンゲームではユーザ同士の社会的なやり取りが面白さの一つであり，幅広く多くのユーザにプレイしていただくことが重要である．したがって施策決定において，可能な限り幅広いユーザの趣味嗜好を考慮することが重要である．また，スマートフォンゲームは数日単位でさまざまなイベントが開催されているため日々状況が変わっていくが，アンケート調査や人気投票は実施は短期間に何度も実施することが難しく，人気度が把握できたときにはすでに状況が変わってしまうという問題も存在する．

そこで我々はユーザの行動ログからキャラクタの人気度を推定することを試みた．それによってゲームをプレイしている全ユーザに基づくキャラクタやカードの人気度をリアルタイムに把握することができる．また，2 章で述べたように分析基盤はすでに整備されているため，容易に素早く多くのユーザデータを分析することが可能である．

キャラクタやカードの人気度の指標として，“リーダカードの設置率”を採用する．リーダカードとはユーザが手持ちのカードから 1 枚選択したカードであり，そのユーザのプレイ画面やそのユーザが書き込んだ掲示板のメッセージの横などに表示される．どのカードをリーダカードにするかによるゲーム上のパラメータへの影響は少ない（分析対象期間の仕様）．したがってリーダカードはプレーヤの好み，または，他プレーヤにアピールしたいという欲求によって手持ちのカードから選択されていると考えることができる．また，各カードはリリースされた直後が最も人気があり，時間とともに減衰すると考えられるため，リリースされてからの経過時間も重要な要素である．そのため，我々はリーダカードが設置されるか否かは，キャラクタの人気とカードの人気，カードがリリースされてからの経過月数によって決まると考え，以下の一般化線形混合モデル [Gaecki 13, Littell 06] によりキャラクタとカードの人気を推定した．

$$n_{ij} \sim Poisson(\lambda_{ij}) \tag{1}$$

$$\ln\lambda_{ij} = \ln u_{ij} + \beta_0 + \beta_1 m + e_i + a_{ij} + (e'_i + a'_{ij})m \tag{2}$$

n_{ij} はキャラクタ i のカード j をリーダカードに設置したユーザ数である．それをカード j（キャラクタは i）を獲得したユーザ数 u_{ij} に比例するとし，カードリリースからの経過月数 m, キャラクタ i ごとの変量効果（リリース直後のキャラクタ固有の人気（人気初速）の影響：e_i, 経過月数によるキャラクタ固有の人気の落込みの影響：e'_i），キャラクタ i のカード j ごとの変量効果（リリース直後のカード固有の人気（人気初速）の影響：a_{ij}, 経過月数によるカード固有の人気の落込みの影響：a'_{ij}）に基づくポアソン分布で説明しようとするモデルである．カードは大まかにはレアリティによってゲーム上の性能と入手しやすさが異なるため，レアリティごとにモデルを作成している．分析に使用したデータを表 1 に示す．

ここではキャラクタ“周圭斗”のレアリティ“SSR”のカードの結果を一例として紹介する．SSR カードの平均的な傾向（固定効果）は $\beta_0 = -1.00$, $\beta_1 = -0.32$ であった．SSR のカードがリリースされているキャラクタごとの傾向（変量効果：e_i, e'_i）の抜粋を表 2 に示す．これは SSR カードの平均的な傾向に対する相対的な差異を示す．例えば“周圭斗”の場合，SSR カード全体の人気初速である −1.00 に比べて 0.32 高く，SSR カード全体の人気の落込みである −0.32 に比べて 0.01 低い（人気が

表 1 分析対象ゲームの基本情報

運営会社	株式会社サイバーエージェント
ゲーム名	ボーイフレンド（仮）
URL	http://bf.amebagames.com
分析対象期間	2013 年 12 月〜 2014 年 5 月

表 2 SSR カードのキャラクタごとの傾向

キャラクタ名	e_i	e'_i
鷹司正臣	0.46	− 0.02
芳屋直景	0.42	− 0.01
真山恭一郎	0.41	− 0.01
遊馬百汰	0.41	− 0.01
周圭斗	0.32	− 0.01
⋮	⋮	⋮
皇アラン	− 0.77	0.03

表 3 SSR の周圭人のカードごとの傾向

カード名	a_{ij}	a'_{ij}
[熱のせい] 周圭斗	0.54	0.18
[レバー] 周圭斗	0.24	0.12
[泣かないの ?] 周圭斗	0.22	0.11

長持ちしやすい）ことを示す．

次にカードごとの傾向の一例としてキャラクタ“周圭斗”について表3に示す．同表から“周圭斗”のカードの中では“[熱のせい] 周圭斗”というカードの人気初速 a_{ij} が最も高く，人気の減衰 a'_{ij} が最も遅いことがわかる．

このようにユーザの行動ログの分析をすることによって，ゲームをプレイしている全ユーザの嗜好をリアルタイムに把握でき，それによって日々状況が変わるスマートフォンゲームの品質向上のための施策に生かしている．このほかにもゲーム上の社会的環境向上のためにユーザ間の協力関係の分析なども実施している [高野15]．

5. ま と め

本稿では当社におけるデータ活用のための処理基盤とデータ活用の事例を紹介した．

Hadoopというシーズが大量データを活用したいというニーズにマッチしたことから始まったプロジェクトがデータが集まることで別のニーズを発掘し，そのニーズを満たすためにまた新たなシーズを適用していくというサイクルが生まれている．またWebサービス運用にOSSを活用し，そこでの問題を解決していくことで，経済的なメリットを生むだけでなく，OSSコミュニティに対して利用実績として還元することができ，ソフトウェア開発に間接的にではあるが貢献していると考えている．さらにはバグの報告やパッチを送ることで直接的な貢献もしている．

◇ 参 考 文 献 ◇

[Fan 13] Fan, W. and Bifet, A.: Mining big data: Current status and forecast to the future, *ACM SIGKDD Explorations Newsletter*（2013）

[Gaecki 13] Gaecki, A. and Burzykowski, T.: *Linear Mixed-Effects Models Using R — A Step-by-Step Approach,* Springer（2013）

[數見 14] 數見拓朗：アメーバブログにおけるスパムブログ検知：機械学習を用いたスパムフィルタの開発，第7回Webとデータベースに関するフォーラム（WebDB Forum 2014）（2014）

[Littell 06] Littell, R. C., Millike, G. A., Stroup, W. W., Wolfinger, R. D. and Schabenberger, O.: *SAS for Mixed Models, Second Edition,* SAS Institute（2006）

[松尾 15] 松尾 豊，吉田宏司，榊 剛史：AIの AKB48 論，人工知能，Vol. 30, No. 1, pp. 89-96（2015）

[本宮 12] 本宮 学：サービス連携で「大きく勝ちに」――新生「Ameba」が掲げる“デカグラフ”戦略，ITMedia（2012）

[Rodriguez 10] Rodriguez, M. A. and Neubauer, P.: Constructions from Dots and Lines, *Bulletin of Association for Information Science and Technology,* Vol. abs/1006.2361（2010）

[鈴木 15] 鈴木俊裕，梅田永介，柿島大貴：HBase 徹底入門 Hadoop クラスタによる高速データベースの実現, 翔泳社（2015）

[高野 15] 高野雅典，和田計也，福田一郎：ソーシャルゲームプレーヤの協調行動の分析，人工知能，Vol. 30, No. 1, pp. 74-82（2015）

[和田 14] 和田計也，高野雅典：スマートフォンゲーム「ボーイフレンド（仮）」におけるキャラクター & カードの人気度分析，株式会社サイバーエージェントテックレポート（https://www.cyberagent.co.jp/corporate/techreport/）（2014）

[善明 13] 善明晃由：Amebaにおけるログ解析基盤の変遷，第6回Webとデータベースに関するフォーラム（WebDB Forum 2013）（2013）

2015年3月9日 受理

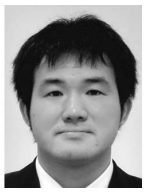
著 者 紹 介

福田 一郎

2006年東京大学工学部システム創成学科卒業．2008年同大学院工学系研究科精密機械工学専攻修士課程修了．同年より，株式会社サイバーエージェント勤務．現在，同社メディア事業担当 最高技術責任者．大規模データ解析基盤の研究開発に従事．

鈴木 俊裕

2006年茨城大学工学部情報工学科卒業．2008年同大学院理工学研究科情報工学専攻博士前期課程修了．同年より，株式会社サイバーエージェント勤務．自社基盤システムの開発・運用に携わった後，大規模データ解析基盤や分散データベースの研究開発に従事．著書に「HBase 徹底入門 Hadoop クラスタによる高速データベースの実現」（翔泳社，2015）がある．



善明 晃由

2004年東京工業大学大学院情報理工学研究科計算工学専攻修士課程修了．総合電機メーカを経て，現在は株式会社サイバーエージェントにて大規模データ解析基盤の研究開発に従事．分散コンピューティング，ソフトウェア工学などに興味をもつ．博士（工学）．

高野 雅典（正会員）

2009年名古屋大学大学院情報科学研究科博士課程修了．博士（情報科学）．大手 SIer を経て，現在は株式会社サイバーエージェントに勤務．自社サービスの開発・運用に携わった後，現在は自社のソーシャルサービスのデータマイニングに従事．複雑系，社会性の進化，統計モデリング，データビジュアライゼーションなどに興味をもつ．

藤坂 祐介

2012年東京工業大学大学院情報理工学研究科計算工学専攻修士課程修了．大学院在学中より，株式会社サイバーエージェントにてインターンとして勤務．現在はフィルタリング基盤の研究開発や社内システムの運用に従事．自律分散システム，GPUコンピューティングなどに興味をもつ．