Webサイトのアクセス解析

㈱ ふくおかフィナンシャルグループ 営業企画部
須永 真昼

SASインスティチュートジャパン プロフェッショナルサービス本部
村上 廉史

Web-Site Analytics

Fukuoka Financial Group, Inc.

Mahiru Sunaga

SAS Institute Japan Ltd.
Kiyoshi Murakami

SAS ユーザー総会 アカデミア/テクノロジー&ソリューション セッション 2010

要旨:

結合ログ形式(Combind log format)のApacheのログファイルを入力ファイルに、具体的なWebサイトのアクセス解析を実施する方法を、サンプルプログラムを交えてご紹介します。

キーワードを続けてキーワードを記載

Web-site analytics apache combined log format Base SAS

Webアクセスログとは・・・

ホームページへの外部からのアクセス履歴を、テキスト形式で保存したもの。

＜例＞

XXX.XXX.XXX.XXX - - [02/May/2010:12:52:55 +0900] "GET /url/url2/page2.htm HTTP/1.1" 200 - "http://www.domain.co.jp/url/url2/page1.htm" "(compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0; YTB720; GTB6.4; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; .NET CLR 3.5.30729; .NET CLR 3.0.30618)"

Webアクセスログから分かること・・・

アクセスログに記載される主な情報

- 接続元IPアドレス
- リクエストの日時
- リクエストファイル（＝閲覧されたページ）
- リクエスト結果
- データサイズ
- リファラー（前のページのURLと検索キーワード）
- エージェント（接続環境）

だれが（IPアドレスベース）
いつ
どのページを
見た／見ようとした
どこから
どんな環境で

3

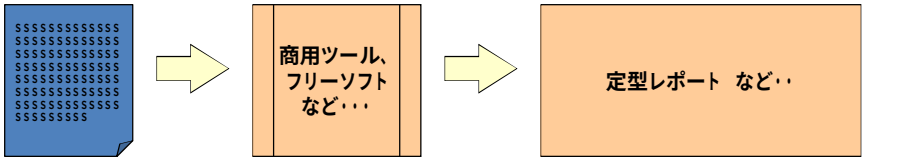
大量のアクセスログを処理することで分かること

- よく見られるページ（人気のあるページ）がどこか
- どの時間帯／曜日のアクセスが多いか
  ⇒ 時間帯／曜日で人気に差異はあるか
- どのページがから、HPに来ているか
- どのページで、自社のHPから出て行ってしまっているか
- ページに来た人は、目的とするページに到達しているか
- どのような環境からインターネットに接続しているか

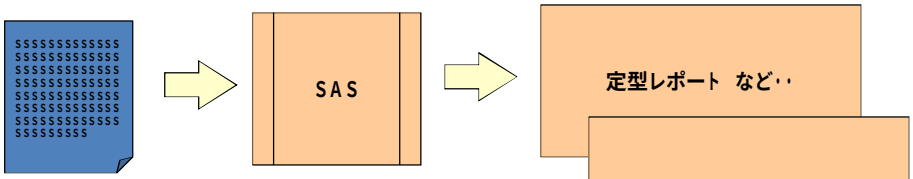
- 顧客のニーズは何か（注目されている商品は？　商品の人気は上昇／下降）
- 自社のHPで、訴求できていない部分（不満足なページ）がわかる
  ⇒ コンテンツ内容、デザイン等を見直す
- ケータイからのアクセスはどれくらいか？
- ユーザーの利用している、OS、ブラウザは
  ⇒ ケータイのページ必要性や、新しい機種への対応の必要性

4

一般的なデータ取り込みの流れ



SAS® を使用して・・・



アクセスログを取り込む上での課題

- **分析に不要なデータが大半**
  ＜理由＞
  アクセスログは、全てが記載されているため、Webページ内の画像ファイル等の取り込みについても全てが記載されている。

  ＜対応策＞
  ログファイルの中から・・・ gif、jpg、bmp 等の画像ファイルを削除する。

- **いわゆるクローラーと呼ばれる、ロボットによるアクセスが大量に存在する**
  ＜理由＞
  検索サイト等が、世界中のWebサイトを巡回し、ページの内容を取り込みINDEX化を実施しているため、定期的に自社サイトにもアクセスしてきている。
  （自社のHPで設置している、サイト内検索ツール等も巡回している。）

  ＜対応策＞
  クローラーと判断できる（できそうな）アクセスについて削除する。
  （一部の大手サイトは、自社のクローラーの情報を公開しているが、非公開のクローラーも多く、データを見たうえで判断していくしかないものも多い。）

  多くは、"bot" や "crawl" といったキーワードをagentに埋め込んである。

- **Webサイトへのアタックによるログが残る**
  ＜理由＞
  国内・海外を問わず、Webサイト内の情報を取得するために、多様なアタックの形跡がある

  ＜対応策＞
  Webページに公開していないURLで、拡張子がzipやgz、mdbなどをリクエストしてくるケースを削除する。

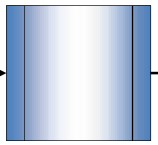
- **検索キーワードの文字コードは様々であり、エンコードしないと読めない。**
  また、検索キーワードの抽出は一筋縄では上手くいかない。
  ＜理由＞
  検索エンジン等によっては、多様なブラウジング環境へ対応するため、ユーザーが文字コードを指定して利用することができる。
  検索キーワードのアクセスログへの埋め込み方は、多様であり個別の対応がどうしても必要。

  ＜対応策＞
  検索サイトからの流入してきたログについては、検索エンジンの種類、ログ内のリファラー情報の記載内容に応じた　判断処理を実装しなければならない。
  (現在：99%程度は、対応できているが・・・)

SASによるデータ取り込むプログラム

処理フロー概要

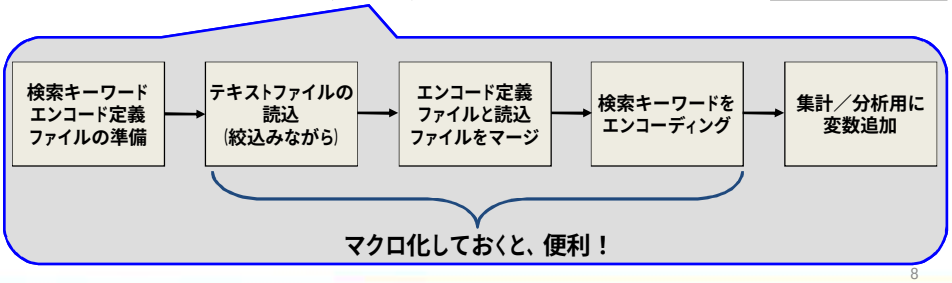
ZIP形式で圧縮された
テキストファイル



DATAステップ
ファイルの読み込み

SAS　DATASET

集計／分析用の
サブデータセット



詳細プログラム

まず、検索エンジンのエンコードタイプの指定をしたデータセットを作成しておきます。(固定で、データセットとして、保持していても問題なし)

```
/*検索エンジン設定ファイル*/
data sengine;
  infile cards;
  input domain:$40. key:$15. key2:$15. enc:$5.;
  keylen=length(key);

  cards;
  www.google.co.jp        ?q=       &q=     UTF-8
  search.yahoo.co.jp      ?p=       &p=     UTF-8
  search.goo.ne.jp        ?MT=      &MT=    UTF-8
  /****more****/
  ;
run;
```

ここでは、3パターンを記載していますが、実際は、40パターン程度を設定していします。

Key key2 を用意することで、検索ワードの表示位置の差異に対応しています。


```
%macro getlog(logdir=,zipfile=,logfile=,out_DS=);  /*マクロとして定義*/

/* ZIPファイルのまま読む込むための、filenameステートメント saszipam */
/* 注：SAS社の正式サポート構文ではない */
filename INPUTZIP saszipam "&logdir.¥&zipfile";

data _blog;
  /* アクセスログのデリミタは、半角スペース */
  infile INPUTZIP(&logfile) dlm=" " MISSOVER DSD lrecl=32767;
  format ipadd $15. _cident $10. _uid $10. dtime datetime19. date yymmdd10.
       time time8. _gmt $6. _req $550. method $7. url $512. http $3.
       _status $3. status 3. _size $12. size best12. referer $1000. agent $256.;
  informat ipadd $15. _dtime $21. dtime 8. _gmt $6. _req $200.
        _status $3. status 3. size best12. referer $1000. agent $256.;
  input  ipadd _cident _uid _dtime _gmt _req _status _size referer agent;

  /* gif,jpg等を削除 */
  if prxmatch('/.(gif|jpg|js|css|png|bmp|zip|gz|mdb)/',_req) then delete;
  else do;

/*************** NEXT PAGE ***************/
```

```
/* 監視アクセス（ロードバランサー）を削除 */
    if ( ipadd='10.10.10.10' or ipadd='10.10.10.11' ) then delete;
    else do;
     /* クローラー等の削除 */
     _agent=lowcase(agent);
     if ( index(_agent,'bot') or
          index(_agent,'craw') or
          index(_agent,'y!j') or


              /****more****/


          ipadd='XXX. XXX. XXX. XXX'
             /* 関係者や監視系については、IPアドレスで削除 */
     ) then delete;


/**************** NEXT PAGE ***************/
```

Webの世界を巡回している、クローラーと呼ばれるアクセスを削除していく。
公開されているものもあれば、非公開のものもある。（キレイにするのは、かなりの手間）

社内からのアクセス、制作会社のアクセス、サーバー監視関連のアクセスについては、agentでは分からないものも多いので、IPが特定できる場合はIPアドレスを使って、削除。


```
    else do;
      dtime=input(substr(_dtime,2),datetime20.);
      date=datepart(dtime);
      time=timepart(dtime);

      method=scan(_req,1," ");

      if _status="-" then _status=".";
      status=input(_status,3.);

      if _size="-" then _size=".";

      size=input(_size,best12.);
      url=urldecode(scan(_req,2," "));
      http=scan(scan(_req,3," "),2,"/");
      sengine=scan(referer,2,"/");
    end;
  end;
 end;
 drop _:;
run;

/**************** NEXT PAGE ***************/
```

生のテキストデータの値から、SAS日付値・SAS時間値を生成する。

SASで扱い易いように、欠損値の置換処理をおこなったり、文字列⇒数値への型変換を実施するなど、SASで取扱いやすくするための処理を実施

```
options compress=YES;/*ファイルサイズを小さくするためのオプション*/
data WLOG.&out_ds;
  format ipadd dtime date time method url http status
         size referer keywords agent;
  set _blog;
 /*  マージをしていく部分・・・  データが大きいので  merge  を使わず、
     SET  ＋  SETで対応していく  */
  if missing(sengine)=0 then
  do;
    do i=1 to na;
      set wlog.sengine point=i nobs=na;
      if sengine=domain then leave;
    end;

/***************  NEXT  PAGE  ***************/
```

_blog

<VAR>	<VAR>	sengin	<VAR>
		yahoo	
		google	
		yahoo	
		google	
		yahoo	
		・	
		・	
		google	

sengin

domain			
yahoo			
google			
・・・etc			

「sengin = domain」を、順番に見て行って、一致するobsを横結合する。


```
    _pos=kindex(referer,key);   /*  ?ではじまるパターン */
    _pos2=kindex(referer,key2); /*  &ではじまるパターン */

    /* リファラー内でエンコーディングが指定されているもの */
    if kindex(referer,"ei=UTF-8")     > 0 then enc="UTF-8";
    if kindex(referer,"ei=Shift_JIS") > 0 then enc="SJIS";
    if kindex(referer,"ei=EUC-JP")    > 0 then enc="EUC";
          /****more****/

    if _pos > 0 then
      keywords=kcvt(urldecode(scan(substr(referer,_pos+keylen),1,'&')),enc,"SJIS");
    else if _pos2 > 0 then
      keywords=kcvt(urldecode(scan(substr(referer,_pos2+keylen),1,'&')),enc,"SJIS");
  end;
  output wlog.&out_ds.;
  drop sengine enc key key2 domain keylen _pos _pos2;
run;
%mend;  /*マクロ定義終了*/

/*マクロの実行処理例*/
%getlog(logdir=E:¥SUGI_LOG,zipfile=apache_testlog.zip,
          logfile=apache_testlog.log,out_DS=all);
```

検索キーワードの入り方を、確認する。

リファラー内でエンコードが指定されているケースもある。

1）文字列の切り出し
scanとsubstr
2）文字列のSJISへの変換
KCVTとurldecode

例）%93%FA%96%7B%8C%EA
⇒ 日本語

プログラム　集計／加工用の処理（例）

```
/* 媒体、ブラウザ、OSを判定 */
data wlog.all2;
  set wlog.all;
  format MOB $15.;
  format Browser $15.;
  format OS $15.;
/*媒体*/
  select;
    when (index(agent,'DoCoMo') > 0 ) MOB = 'docomo';
    when (index(agent,'BlackBerry') > 0 ) MOB = 'BlackBerry';
    when (index(agent,'SO-01B') > 0 ) MOB = 'Xperia';
    when (index(agent,'KDDI') > 0 ) MOB = 'Au';
    when (index(agent,'SoftBank') > 0 ) MOB = 'SoftBank';
    when (index(agent,'iPhone;') > 0 ) MOB = 'iPhone';
    when (index(agent,'Nintendo Wii') > 0) MOB = 'Wii';
    when (index(agent,'Windows') > 0 ) MOB ='WIN' ;
    when (index(agent,'Mac') > 0 ) MOB ='Mac' ;
          /****more****/
    otherwise MOB = 'other';
  end;

/***************  NEXT  PAGE  ***************/
```

ログ内のエージェントの項目を利用して、接続媒体やブラウザ、ＯＳの判定を実施する

15


```
  /* ブラウザ */
  select;
    when (index(agent,'Opera') > 0 ) Browser = 'Opera';
    when (index(agent,'Safari') > 0 ) Browser = 'Safari';
    when (index(agent,'Chrome') > 0 ) Browser = 'Chrome';
   when (index(agent,'Firefox/3.6') > 0 ) Browser = 'Firefox/3.6';
   when (index(agent,'MSIE 5.5') > 0 ) Browser = 'IE5.5';
    when (index(agent,'MSIE 6.0') > 0 ) Browser = 'IE6';
    when (index(agent,'MSIE 7.0') > 0 ) Browser = 'IE7';
    when (index(agent,'MSIE 8.0') > 0 ) Browser = 'IE8';
          /****more****/
    otherwise Browser = 'OTHER';
  end;
  /* OS */
  select;
    when (index(agent,'Windows NT 5.1') > 0 ) OS = 'WIN_XP';
    when (index(agent,'Windows NT 6.0') > 0 ) OS = 'WIN_VISTA';
    when (index(agent,'Windows NT 6.1') > 0 ) OS = 'WIN_7';
    when (index(agent,'Windows 98') > 0 ) OS = 'WIN 98';
          /****more****/
    otherwise OS = 'OTHER';
  end;
run;
```

16

パフォーマンスに関して

SAS化前	一部SAS化	SAS化後
そもそも、処理不可能	ZIPファイル⇒テキストファイル化 テキストファイルを、フリーのソフトで読み込み⇒CSV形式ファイル CSV形式ファイルを、SASに取り込み	現在　約2時間程度でデータセット化終了
あきらめ・・・	3日～4日の手作業処理	

SASデータセット化したアクセスログの活用

SASデータセット化してしまえば、後は比較的自由に活用できる・・・

（例）
「Webサイトへの訪問者の動きをトレースしたい」
⇒ 同一セッションを特定しておく必要がある。
IPアドレスは、同じIPアドレスが利用されるので、同一IPアドレスからの接続について、
30分以上開いたら、別セッションとする。（1つの考え方）

```
/* セッションをまとめる */
proc sort data=wlog.ALL out=_sort;
  by ipadd dtime;
run;

data _session;
  set _sort;
  by ipadd;
  retain session 1 _dt2 view_no 1;
  /* IPアドレス毎の初期値を指定 */
  if first.ipadd then do;
    _dt2 = dtime;   view_no=0;   session=1;
  end;

  /* dtime+30minを_dt2に設定する   9.1.3 SP4 */
  _dt2=TINTNX('min30',_dt2,1,'SAME');
  /* 30分以上のケース  新セッション */
  if _dt2 < dtime then do;
    SESSION+1;  View_NO = 1;  _dt2=dtime;
    output;
  end;

/*****************  NEXT  PAGE  ****************/
```

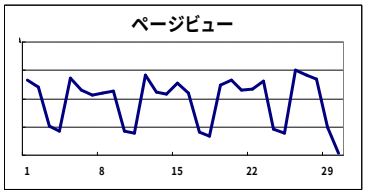
1度の接続をある程度特定したい。
(来訪回数や、平均閲覧ページ数、
退出されやすいページを見たい)

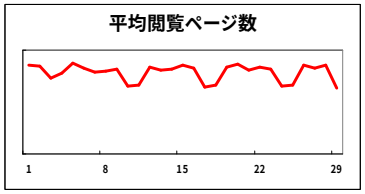
(・・・いたしかたないので、ソーティング)
※データサイズが大きい場合には、sortプロシージャーを変更することで、パフォーマンスの改善が可能。

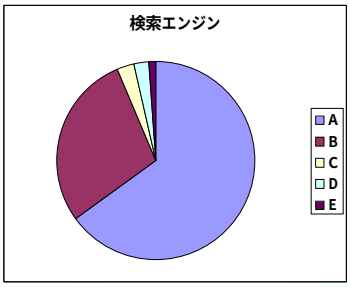
同一IPで、30分以上間隔が空いたら、別セッションと見なす例

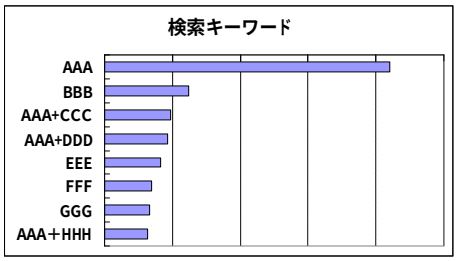

```
/* 30分以内同一セッション */
  else if _dt2 >= dtime then do;
    VIEW_NO = VIEW_NO + 1;
    output;
    /* 前OBSの値を格納 */
    _dt2=dtime;
  end;
  drop _:;
run;

/* 流入ページ、流出ページ */
proc sort data=_session out=_sessort;
  by ipadd session;
run;

data wlog.inout;
  set _sessort;
  length in_p out_p $3.;
  by ipadd session;
  if first.session then  in_p="IN";
  if last.session then  out_p="OUT";
run;
```

各種集計等を行えば・・・

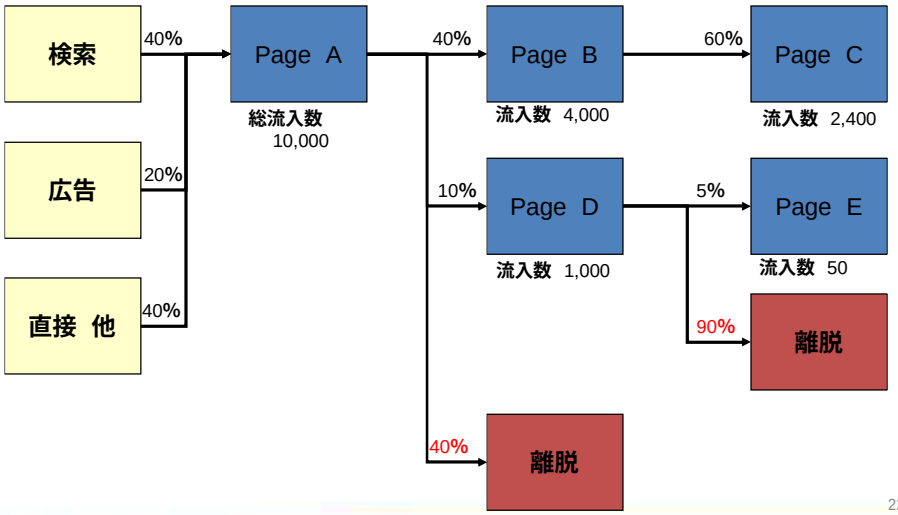








更に、加工をしていくなら・・・（専用ツールなら、ボタン1つですが・・・）



サンプルプログラムのハイライト(1/2)

■ SASZIPAMアクセスメソッド
Windows版SASでのみ利用可能で、サンプルプログラム中でZIPファイルの解凍に使用
SASZIPAMは、SASのインストールプロセスにて使用されるツール
インストール時以外の使用は、サポート対象外

Usage Note 31244: SASZIPAM access method - use at your own risk.
http://support.sas.com/kb/31/244.html

■URLENCODE, URLDECODE関数
URLENCODE : URLに含むことでできない文字を符号化する
URLDECODE : 復元

```
DATA _null_;
  a=URLENCODE('English%日本語');
  b=URLDECODE(a);
  PUT a=; PUT b=;
RUN;
```

a=English%25%93%FA%96%7B%8C%EA
b=English%日本語

サンプルプログラムのハイライト(2/2)

■ KCVT関数
別タイプのエンコーディングデータに変換する関数

```
DATA a;
  a="E697A5E69CACE8AA9E"x; /* UTF-8の16進数にて日本語を指定 */
  b=KCVT(a,"UTF-8","SJIS");
  PUT b=;
RUN;
```

*** 注意点 ****
文字列を出力するK関数のデフォルトの変数長は、200バイトとなる
LENGTHステートメントにて変数長を事前に定義することを、お勧めします

変数と属性の昇順リスト

#	変数	タイプ	長さ
1	a	文字	9
2	b	文字	200

＜参考サイト＞

■ Yahoo! JAPANの検索エンジン（クローラー）について
http://help.yahoo.co.jp/help/jp/search/indexing/indexing-15.html
■ モバイル版Yahoo!検索の検索エンジン（クローラー）について
http://help.yahoo.co.jp/help/jp/search/indexing/indexing-27.html
■ Yahoo!検索（ウェブ検索）の検索パラメータ仕様
http://developer.yahoo.co.jp/other/query_parameters/search/websearch.html
■ Googlebot の確認
http://www.google.com/support/webmasters/bin/answer.py?answer=80553
■ Apacheのログファイルについて
http://httpd.apache.org/docs/2.0/ja/logs.html

SAS ユーザー総会 アカデミア/テクノロジー&ソリューション セッション 2010

ご清聴ありがとうございました。