**FIT5148 – Distributed and Big Data Processing, Semester 1, 2016**

**Big Data Report**

Luyang Ye 26610000

Zhengxing Xia 26557355

Content

[**Part 1 3**](#_Toc452036633)

[**Task 1 4**](#_Toc452036634)

[**Task 2 18**](#_Toc452036635)

[**Task 3 45**](#_Toc452036636)

[**Task 4 69**](#_Toc452036637)

[**Task 5 85**](#_Toc452036638)

[**Part 2 97**](#_Toc452036639)

[**1. Introduction 97**](#_Toc452036640)

[**2. Current state of the art 97**](#_Toc452036641)

[**2.1 Challenges 97**](#_Toc452036642)

[**2.2 The proposed approaches 98**](#_Toc452036643)

[**2.3 Findings and results 99**](#_Toc452036644)

[**2.4 Compare 100**](#_Toc452036645)

[**3. Conclusion 101**](#_Toc452036646)

[**4. References 101**](#_Toc452036647)

# **Part 1**

|  |
| --- |
| System’s detail  I use lab’s computer so I only can check the information below. |

|  |
| --- |
|  |

## **Task 1**

|  |
| --- |
| Upload all the files |

|  |
| --- |
| Table temp\_awardscoaches |

|  |
| --- |
| Table temp\_awardsplayers |
| Table temp\_coaches |
| Table temp\_masters |

|  |
| --- |
| Table temp\_scoring |

|  |
| --- |
| Table temp\_teams |

## **Task 2**

|  |
| --- |
| a)  **Assumption:**  Join table scoring and table team together with same teamID.  Join table scoring and table master together with same playerID.  Set gwg equals to the maximum value of gwg  **Pre-processing:**  create table t2\_a\_scoring(playerid STRING,year int,pos STRING,tmID STRING,lgID STRING,GWG int);  insert overwrite table t2\_a\_scoring SELECT  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){1}', 1) playerid,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){2}', 1) year,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){6}', 1) pos,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){4}', 1) tmid,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){5}', 1) lgid,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){17}', 1) gwg  from temp\_scoring;  -------------------------------------------  create table t2\_a\_master(playerid STRING,firstName STRING,lastName STRING,birthCountry STRING,birthYear int,birthMon int,birthDay int);  insert overwrite table t2\_a\_master SELECT  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){1}', 1) playerid,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){4}', 1) firstName,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){5}', 1) lastName,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){23}', 1) birthCountry,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){20}', 1) birthYear,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){21}', 1) birthMon,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){22}', 1) birthDay  from temp\_master;create table t2\_a\_teams(tmID STRING,tname STRING);  --------------------------------------------------  create table t2\_a\_teams(tmID STRING,tname STRING);  insert overwrite table t2\_a\_teams SELECT  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){3}', 1) tmid,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){19}', 1) tname  from temp\_teams; |

|  |
| --- |
| HIVE:  select b.gwg,b.year,b.pos,b.lgid,d.tname,a.firstname,a.lastname,a.birthcountry,a.birthyear,a.birthmon,a.birthday  from t2\_a\_scoring b  join  (  select tname,tmid from t2\_a\_teams  ) d  on (b.tmid = d.tmid)  join  (  select firstname,lastname,birthcountry,birthyear,birthmon,birthday,playerid  from t2\_a\_master  ) a  on (b.playerid = a.playerid)  where b.gwg in  (  select max(gwg)  from t2\_a\_scoring  )  group by b.gwg,b.playerid,b.year,b.pos,b.lgid,d.tname,a.firstname,a.lastname,a.birthcountry,a.birthyear,a.birthmon,a.birthday  ;  **Log:**      **Result**      **Time cost (89 sec)** |

|  |
| --- |
| **Assumption:**  Find max value of gwg first and join each table with same playerID or same teamID.  **PIG:**  scoring\_1 = load '/tmp/as2/Scoring.csv' using PigStorage(',');  scoring\_raw = FOREACH scoring\_1 GENERATE $0 as playerid, $1 as year, $5 as pos, $3 as tmid,$4 as lgid,$16 as gwg;  scoring\_2 = FILTER scoring\_raw BY playerid != 'playerID';  group\_scoring = GROUP scoring\_2 ALL;  max\_gwg1 = FOREACH group\_scoring GENERATE MAX(scoring\_2.gwg) as max;  max\_gwg2 = FILTER scoring\_2 BY gwg == max\_gwg1.$0;  player\_1 = load '/tmp/as2/Master.csv' using PigStorage(',');  player\_raw = FOREACH player\_1 GENERATE $0 as playerid, $3 as firstname, $4 as lastname, $22 as birthcountry,  $19 as birthyear,$20 as birthmon,$21 as birthday;  player\_2 = FILTER player\_raw BY playerid != 'playerID';  join\_playergwg = JOIN max\_gwg2 BY (playerid),player\_2 BY (playerid);  team\_1 = load '/tmp/as2/Teams.csv' using PigStorage(',');  team\_raw = FOREACH team\_1 GENERATE $2 as tmid, $18 as tname;  team\_2 = FILTER team\_raw BY tmid != 'tmID';  join\_playergwgteam = JOIN join\_playergwg BY (tmid),team\_2 BY (tmid);  join\_playergwgteam2 = DISTINCT join\_playergwgteam;  final = FOREACH join\_playergwgteam GENERATE $1 as year,$2 as pos,$4 as lgid,$5 as gwg,$7 as firstname  ,$8 as lastname,$9 as birthcountry,$10 as birthyear,$11 as birthmon,$12 as birthday,$14 as tname;  final2 = DISTINCT final;  dump final2;  **Result:**    **Log:**      **Time cost (1min 35 sec)** |

|  |
| --- |
| b)  **Assumption:**  Using the same query in part a) and get the max number of awards time. Find out this player and join with table master to get his whole name.  **Pre-processing:**  create table t2\_awardsPlayers (playerID STRING,year int);  insert overwrite table t2\_awardsplayers SELECT  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){1}', 1) playerID,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){3}', 1) year  from temp\_awardsplayers;    **HIVE:**  select a.firstname,a.lastname,c.countNum  from t2\_a\_scoring b  join  (  select firstname,lastname,playerid  from t2\_a\_master  ) a  on (b.playerid = a.playerid)  join  (  select playerid,count(playerid) as countNum  from t2\_awardsplayers  group by playerid  ) c  on (c.playerid = b.playerid)  join(  select max(d.countNum) as countNum from (  select a.firstname,a.lastname,c.countNum as countNum  from t2\_a\_scoring b  join  (  select firstname,lastname,playerid  from t2\_a\_master  ) a  on (b.playerid = a.playerid)  join  (  select playerid,count(playerid) as countNum  from t2\_awardsplayers  group by playerid  ) c  on (c.playerid = b.playerid)  where b.gwg in  (  select max(gwg)  from t2\_a\_scoring  )  group by a.firstname,a.lastname,c.countNum) d  ) e  on(c.countNum = e.countNum)  where b.gwg in  (  select max(gwg)  from t2\_a\_scoring  )  group by a.firstname,a.lastname,c.countNum;  **Result:**    **Log:**      **Time cost (221sec)** |

|  |
| --- |
| **Assumption:**  Using same script in part a) to get the playerID and get their awards time. Find out which get more times.  **PIG:**  scoring\_1 = load '/tmp/as2/Scoring.csv' using PigStorage(',');  scoring\_raw = FOREACH scoring\_1 GENERATE $0 as playerid,$16 as gwg;  scoring\_2 = FILTER scoring\_raw BY playerid != 'playerID';  group\_scoring = GROUP scoring\_2 ALL;  max\_gwg1 = FOREACH group\_scoring GENERATE MAX(scoring\_2.gwg) as max;  --16  max\_gwg2 = FILTER scoring\_2 BY gwg == max\_gwg1.$0;  --(esposph01,16)  player\_1 = load '/tmp/as2/Master.csv' using PigStorage(',');  player\_raw = FOREACH player\_1 GENERATE $0 as playerid,$3 as firstname, $4 as lastname;  player\_2 = FILTER player\_raw BY playerid != 'playerID';  join\_playergwg = JOIN max\_gwg2 BY (playerid),player\_2 BY (playerid);  award\_1 = load '/tmp/as2/AwardsPlayers.csv' using PigStorage(',');  award\_raw = FOREACH award\_1 GENERATE $0 as playerid;  award\_2 = FILTER award\_raw BY playerid != 'playerID';  group\_award = GROUP award\_2 BY playerid;  count\_award = FOREACH group\_award GENERATE $0 as playerid,  COUNT(award\_2.playerid) as award\_num;  join\_playergwgaward = JOIN join\_playergwg BY ($0),count\_award BY ($0);  --(esposph01,16,esposph01,Phil,Esposito,esposph01,17)  join\_playergwgaward1 = FOREACH join\_playergwgaward GENERATE $3 as firstname,$4 as lastname,$6 as countnum;  group\_final = GROUP join\_playergwgaward1 ALL;  max\_final = FOREACH group\_final GENERATE MAX(join\_playergwgaward1.countnum) as maxcount;    join\_playergwgaward3 = FILTER join\_playergwgaward1 BY countnum == max\_final.$0;  final = DISTINCT join\_playergwgaward3;  DUMP final;  **Result:**    **Log**:      **Time cost 2min 1sec** |

|  |
| --- |
| **2.c)**  **Assumption:**  Combine b) and a) and join with table scoring on same year he won awards to find out his records in these year.  **Pre-processing:**  create table t2\_c\_awardsPlayers (playerID STRING,year int,awardName STRING);  insert overwrite table t2\_c\_awardsplayers SELECT  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){1}', 1) playerID,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){3}', 1) year,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){2}', 1) awardName  from temp\_awardsplayers;    create table t2\_c\_scoring(playerid STRING,year int,pos STRING,tmID STRING,lgID STRING,GWG int,Pts int);  insert overwrite table t2\_c\_scoring SELECT  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){1}', 1) playerid,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){2}', 1) year,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){6}', 1) pos,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){4}', 1) tmid,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){5}', 1) lgid,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){17}', 1) gwg,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){10}', 1) pts  from temp\_scoring;    **HIVE:**  select b.awardname,b.year,c.pts from  (select a.firstname,a.lastname,c.countNum,a.playerid as playerid  from t2\_c\_scoring b  join  (  select firstname,lastname,playerid  from t2\_a\_master  ) a  on (b.playerid = a.playerid)  join  (  select playerid,count(playerid) as countNum  from t2\_c\_awardsplayers  group by playerid  ) c  on (c.playerid = b.playerid)  join(  select max(d.countNum) as countNum from (  select a.firstname,a.lastname,c.countNum as countNum  from t2\_c\_scoring b  join  (  select firstname,lastname,playerid  from t2\_a\_master  ) a  on (b.playerid = a.playerid)  join  (  select playerid,count(playerid) as countNum  from t2\_c\_awardsplayers  group by playerid  ) c  on (c.playerid = b.playerid)  where b.gwg in  (  select max(gwg)  from t2\_c\_scoring  )  group by a.firstname,a.lastname,c.countNum) d  ) e  on(c.countNum = e.countNum)      where b.gwg in  (  select max(gwg)  from t2\_c\_scoring  )  group by a.firstname,a.lastname,c.countNum,a.playerid) a  join (select playerid,year,awardname  from t2\_c\_awardsplayers) b  on (a.playerid = b.playerid)  join (select playerid,year,pts  from t2\_c\_scoring  ) c  on (a.playerid = c.playerid and c.year = b.year)  order by year  ;  **Result:**    **Log:**      **Time cost (258sec)** |

|  |
| --- |
| **PIG:**  scoring\_1 = load '/tmp/as2/Scoring.csv' using PigStorage(',');  scoring\_raw = FOREACH scoring\_1 GENERATE $0 as playerid,$16 as gwg,$1 as year,$9 as pts;  scoring\_2 = FILTER scoring\_raw BY playerid != 'playerID';  group\_scoring = GROUP scoring\_2 ALL;  max\_gwg1 = FOREACH group\_scoring GENERATE MAX(scoring\_2.gwg) as max;  --16  max\_gwg2 = FILTER scoring\_2 BY gwg == max\_gwg1.$0;  --(esposph01,16)  award\_1 = load '/tmp/as2/AwardsPlayers.csv' using PigStorage(',');  award\_raw = FOREACH award\_1 GENERATE $0 as playerid,$2 as year,$1 as awardname;  award\_2 = FILTER award\_raw BY playerid != 'playerID';  group\_award = GROUP award\_2 BY playerid;  count\_award = FOREACH group\_award GENERATE $0 as playerid,  COUNT(award\_2.playerid) as award\_num;  join\_playergwgaward = JOIN max\_gwg2 BY ($0),count\_award BY ($0);  --(esposph01,16,1970,152,esposph01,17)  join\_playergwgaward1 = FOREACH join\_playergwgaward GENERATE$0 as playerid,$5 as countnum;  group\_final = GROUP join\_playergwgaward1 ALL;  max\_final = FOREACH group\_final GENERATE MAX(join\_playergwgaward1.countnum) as maxcount;  join\_playergwgaward3 = FILTER join\_playergwgaward1 BY countnum == max\_final.$0;  final = DISTINCT join\_playergwgaward3;  scoring\_3 = FILTER scoring\_2 BY playerid == final.$0;  award\_3 = FILTER award\_2 BY playerid == final.$0;  join\_scoreaward = JOIN scoring\_3 BY (year),award\_3 BY (year);  --(esposph01,8,1974,127,esposph01,1974,Second Team All-Star)  final\_result = FOREACH join\_scoreaward GENERATE $2 as year,$3 as pts,$6 as awardname;  DUMP final\_result;  **Result:**    **Log:**      **Time cost 2min 24sec** |

## **Task 3**

|  |
| --- |
| **Pre-processing:**  create table t3\_coaches (coachID STRING,year int,lgID STRING,g int, w int, l int,t int);  insert overwrite table t3\_coaches SELECT  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){1}', 1) coachID,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){2}', 1) year,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){4}', 1) lgID,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){7}', 1) g,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){8}', 1) w,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){9}', 1) l,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){10}', 1) t  from temp\_coaches;        create table t3\_awardscoaches (coachID STRING);  insert overwrite table t3\_awardscoaches SELECT  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){1}', 1) coachID  from temp\_awardscoaches;        create table t3\_master (playerID STRING, coachID STRING,firstName STRING,lastName STRING,pos STRING,birthyear int, birthmon int,birthday int,birthcountry STRING);  insert overwrite table t3\_master SELECT  regexp\_extract(con\_value, '^(?:([^,]\*)\,?){1}', 1) playerID,  regexp\_extract(con\_value, '^(?:([^,]\*)\,?){2}', 1) couchID,  regexp\_extract(con\_value, '^(?:([^,]\*)\,?){4}', 1) firstName,  regexp\_extract(con\_value, '^(?:([^,]\*)\,?){5}', 1) lastName,  regexp\_extract(con\_value, '^(?:([^,]\*)\,?){19}', 1) pos,  regexp\_extract(con\_value, '^(?:([^,]\*)\,?){20}', 1) birthyear,  regexp\_extract(con\_value, '^(?:([^,]\*)\,?){21}', 1) birthmon,  regexp\_extract(con\_value, '^(?:([^,]\*)\,?){22}', 1) birthday,  regexp\_extract(con\_value, '^(?:([^,]\*)\,?){23}', 1) birthcountry  from temp\_master; |

|  |
| --- |
| a)  **Assumption:**  Join table coaches and table master with same coachID. And set where coachID who has highest number of awards.  **Hive:**  SELECT m.firstName,m.lastName,m.birthYear,m.birthMon,m.birthDay,m.birthCountry,a.awards  FROM t3\_master m  join  (  select coachid ,count(coachid) as awards  from t3\_awardscoaches  group by coachid  ) a  on (m.coachid = a.coachid)  where a.awards in  (  select max(b.countCoach)  from (  select count(coachid) as countCoach  from t3\_awardscoaches  group by coachid) b  )  ;  **Result:**    **Log:**      Time cost |

|  |
| --- |
| **Assumption:**  Count the number of coachID and find out who has the most. Join with the table master to get his name.  **PIG:**  t3\_awardscoaches = load '/tmp/as2/AwardsCoaches.csv' using PigStorage(',');  for\_awardscoaches = FOREACH t3\_awardscoaches GENERATE $0 as coachID;  awardscoaches = FILTER for\_awardscoaches BY coachID != 'coachID';  group\_awardscoaches = GROUP awardscoaches BY coachID;  count\_awardscoaches = FOREACH group\_awardscoaches GENERATE $0 as coachID,  COUNT(awardscoaches.coachID) as awardsNum;  records\_group = GROUP count\_awardscoaches ALL;  with\_max = FOREACH records\_group GENERATE MAX(count\_awardscoaches.$1);  max\_awardscoaches = FILTER count\_awardscoaches BY awardsNum == with\_max.$0;  t3\_master = load '/tmp/as2/Master.csv' using PigStorage(',');  for\_master = FOREACH t3\_master GENERATE $0 as playerID, $1 as coachID, $3 as firstname,  $4 as lastname, $18 as pos,$19 as birthYear, $20 as birthMon,$21 as birthDay,$22 as birthCountry;  master = FILTER for\_master BY playerID != 'playerID';  join\_max = JOIN max\_awardscoaches by $0,master by $1;  t3a\_final = FOREACH join\_max GENERATE $4 as firstname,$5 as lastname, $7 as birthYear,  $8 as birthMon,$9 as birthDay,$10 as birthCountry,$1 as awardsNum;  DUMP t3a\_final;  **Result:**    **Log:**      **Time cost: 1min 9sec**  **3.b)**  **Assumption:**  Join table master and table coach with same coachID.  Select win number as the highest number.  **HIVE:**  SELECT m.firstName,m.lastName,c.year,c.games,c.win,c.lose,c.tier  FROM t3\_master m  join  (  select coachid ,year,g as games,w as win,l as lose,t as tier  from t3\_coaches  ) c  on (m.coachid = c.coachid)  where c.win in  (  select max(w)  from t3\_coaches  )  ;  **Result:**    **Log:**      **Time cost 53sec** |

|  |
| --- |
| **Assumption:**  Find the maximum number of win for coachID. Join with master table to get the name.  **PIG:**  t3\_coaches = load '/tmp/as2/Coaches.csv' using PigStorage(',');  coaches\_raw = FILTER t3\_coaches BY $1 > 0;  coaches = FOREACH coaches\_raw GENERATE $0 as coachID, $1 as year, $3 as lgID,  $6 as g, $7 as w, $8 as l,$9 as t;  coach\_group = GROUP coaches ALL;  with\_max = FOREACH coach\_group GENERATE MAX(coaches.w);  max\_coaches = FILTER coaches BY w == with\_max.$0;  t3\_master = load '/tmp/as2/Master.csv' using PigStorage(',');  for\_master = FOREACH t3\_master GENERATE $1 as coachID, $3 as firstname, $4 as lastname;  master = FILTER for\_master BY coachID != 'coachID';  join\_coachmaster = JOIN max\_coaches BY $0, master BY $0;  join\_coachmaster2 = FOREACH join\_coachmaster GENERATE $8 as firstname, $9 as lastname,$1 as year,  $3 as g,$4 as w,$5 as l,$6 as t;  DUMP join\_coachmaster2;  **Result:**    **Log:** |

|  |
| --- |
| **3.c)**  **Assumption:**  Join table master and table coach together using same coached.  Join table awards to get the max number and using same coachID.  **HIVE:**  SELECT m.firstName,m.lastName,c.year,c.games,c.win,c.lose,c.tier,a.awards  FROM t3\_master m  join  (  select coachid ,year,g as games,w as win,l as lose,t as tier  from t3\_coaches  ) c  on (m.coachid = c.coachid)  join  (  select coachid ,count(coachid) as awards  from t3\_awardscoaches  group by coachid  ) a  on (m.coachid = a.coachid)  where c.win in  (  select max(w)  from t3\_coaches  )  **Result:**    **Log:**      **Time cost 72sec** |

|  |
| --- |
| **PIG:**  t3\_coaches = load '/tmp/as2/Coaches.csv' using PigStorage(',');  coaches\_raw = FILTER t3\_coaches BY $1 > 0;  coaches = FOREACH coaches\_raw GENERATE $0 as coachID, $1 as year, $3 as lgID,  $6 as g, $7 as w, $8 as l,$9 as t;  coach\_group = GROUP coaches ALL;  with\_max = FOREACH coach\_group GENERATE MAX(coaches.w);  max\_coaches = FILTER coaches BY w == with\_max.$0;  t3\_master = load '/tmp/as2/Master.csv' using PigStorage(',');  for\_master = FOREACH t3\_master GENERATE $1 as coachID, $3 as firstname, $4 as lastname;  master = FILTER for\_master BY coachID != 'coachID';  join\_coachmaster = JOIN max\_coaches BY $0, master BY $0;  join\_coachmaster2 = FOREACH join\_coachmaster GENERATE $8 as firstname, $9 as lastname,$1 as year,  $3 as g,$4 as w,$5 as l,$6 as t;  t3\_awardscoaches = load '/tmp/as2/AwardsCoaches.csv' using PigStorage(',');  for\_awardscoaches = FOREACH t3\_awardscoaches GENERATE $0 as coachID;  awardscoaches = FILTER for\_awardscoaches BY coachID != 'coachID';  group\_awardscoaches = GROUP awardscoaches BY coachID;  count\_awardscoaches = FOREACH group\_awardscoaches GENERATE $0 as coachID,  COUNT(awardscoaches.coachID) as numAwards;  join\_awards = JOIN join\_coachmaster BY $0,count\_awardscoaches BY $0;  join\_awards2 = FOREACH join\_awards GENERATE $8 as firstname, $9 as lastname,$1 as year,  $3 as g,$4 as w,$5 as l,$6 as t,$11 as awards;  DUMP join\_awards2;  **Result:**    **Log:**      **Time cost: 1min 18sec** |

## **Task 4**

|  |
| --- |
| **a)**  **Pre-preparation:**  create table t4\_scoring (playerID STRING, year int, tmID STRING,G int,A int,Pts int);  insert overwrite table t4\_scoring SELECT  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){1}', 1) playerID,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){2}', 1) year,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){4}', 1) tmID,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){8}', 1) G,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){9}', 1) A,  regexp\_extract(col\_value, '^(?:([^,]\*)\,?){10}', 1) Pts  from temp\_scoring;    **HIVE:**  select t.tname as teamName,s.g,s.a,s.pts  from t2\_a\_teams t  join  (  select tmid ,sum(g)as g,sum(a) as a, sum(pts) as pts  from t4\_scoring  group by tmid  ) s  on  (t.tmid = s.tmid)  group by t.tname,s.g,s.a,s.pts;  **Result:**      **Log:**    **Time cost 42sec** |

|  |
| --- |
| **PIG:**  t4\_teams = load '/tmp/as2/Teams.csv' using PigStorage(',');  teams\_raw = FOREACH t4\_teams GENERATE $2 as tmID, $18 as teamname, $13 as pts;  teams = FILTER teams\_raw BY tmID != 'tmID';  group\_teamid = GROUP teams BY tmID;  sum\_teamid = FOREACH group\_teamid GENERATE group as tmID,teams.$1 as teamname;  sum\_teamid2 = FOREACH sum\_teamid GENERATE $0 as tmID,FLATTEN($1) as teamname;  sum\_teamid3 = DISTINCT sum\_teamid2;  t4\_scoring = load '/tmp/as2/Scoring.csv' using PigStorage(',');  scoring\_raw = FOREACH t4\_scoring GENERATE $0 as playerID, $1 as year, $3 as tmID,  $7 as g, $8 as a, $9 as pts;  scoring = FILTER scoring\_raw BY playerID != 'playerID';  scoring1 = FOREACH scoring GENERATE $0 as playerID, $2 as tmID,$3 as g, $4 as a,$5 as pts;  join\_player = JOIN scoring1 BY tmID, sum\_teamid3 BY tmID;  group\_join = GROUP join\_player BY teamname;  sum\_scoring = FOREACH group\_join GENERATE group as teamname,SUM(join\_player.pts) as pts,  SUM(join\_player.g) as g,SUM(join\_player.a) as a;  DUMP sum\_scoring;  **Result:**      **Log:**      **Time cost: 16min 58sec** |

|  |
| --- |
| **b)**  **Assumption:**  Join scoring and table team together using same teamID.  Join with table master to get the player and join with scoring to get the highest pts.  **HIVE:**  select y4.tname as teamName,y3.firstname as firstname,y3.lastname as lastname,y1.year as year,y4.pts as pts,y1.g as g,y1.a as a  from t4\_scoring y1  join  (  select tmid,tname  from t2\_a\_teams  group by tmid,tname  ) y2  on (y1.tmid = y2.tmid)  join  (  select playerid,firstname,lastname  from t3\_master  group by playerid,firstname,lastname  ) y3  on (y1.playerid = y3.playerid)  join  (  select s3.tname,max(s3.pts) as pts  from (  select y2.tname,y1.pts  from t4\_scoring y1  join  (  select tmid,tname  from t2\_a\_teams  group by tmid,tname  ) y2  on (y1.tmid = y2.tmid)  ) as s3  group by s3.tname  ) y4  on (y2.tname = y4.tname and y1.pts = y4.pts);  **Result:**      **Log:**    **PIG**  t4\_teams = load '/tmp/as2/Teams.csv' using PigStorage(',');  teams\_raw = FOREACH t4\_teams GENERATE $2 as tmID, $18 as teamname;  teams = FILTER teams\_raw BY tmID != 'tmID';  group\_teamid = GROUP teams BY tmID;  sum\_teamid = FOREACH group\_teamid GENERATE group as tmID,teams.$1 as teamname;  sum\_teamid2 = FOREACH sum\_teamid GENERATE $0 as tmID,FLATTEN($1) as teamname;  sum\_teamid3 = DISTINCT sum\_teamid2;  t4\_scoring = load '/tmp/as2/Scoring.csv' using PigStorage(',');  scoring\_raw = FOREACH t4\_scoring GENERATE $0 as playerID, $1 as year, $3 as tmID,  $7 as g, $8 as a, $9 as pts;  scoring = FILTER scoring\_raw BY playerID != 'playerID';  join\_player = JOIN scoring BY tmID, sum\_teamid3 BY tmID;  sum\_scoring1 = FOREACH join\_player GENERATE $0 as playerID,$7 as teamname,$1 as year,$5 as pts,$3 as g,$4 as a;  group\_max = GROUP sum\_scoring1 BY (teamname);  max\_pts = FOREACH group\_max GENERATE group,MAX(sum\_scoring1.pts) as max;  max\_pts1 = FOREACH max\_pts GENERATE FLATTEN($0) as teamname,$1 as max;  max\_pts2 = JOIN sum\_scoring1 BY (teamname,pts),max\_pts1 BY (teamname,max);  max\_pts3 = FOREACH max\_pts2 GENERATE $0 as playerID,$1 as teamname,$2 as year,$3 as pts,$4 as g,$5 as a;  t4\_master = load '/tmp/as2/Master.csv' using PigStorage(',');  for\_master = FOREACH t4\_master GENERATE $0 as playerID, $3 as firstname, $4 as lastname;  master = FILTER for\_master BY playerID != 'playerID';  join\_master = JOIN max\_pts3 BY playerID, master BY playerID;  final = FOREACH join\_master GENERATE $1 as teamname,$7 as firstname,$8 as lastname,  $2 as year,$3 as pts,$4 as g,$5 as a;  DUMP final;  **Result:**      **Log:**      **Time cost: 2min 18sec** |

**Conclusion:**

**PIG and HIVE are doing same thing but using different method or scheme. In all these tasks above, PIG seems faster than HIVE, but that only because HIVE queries have more JOIN than pig. If the query of HIVE is optimized then we believe it can have better performance.**

## **Task 5**

|  |
| --- |
| **a)**  Using task 3a PIG script:  t3\_awardscoaches = load '/tmp/as2/AwardsCoaches.csv' using PigStorage(',');  for\_awardscoaches = FOREACH t3\_awardscoaches GENERATE $0 as coachID;  awardscoaches = FILTER for\_awardscoaches BY coachID != 'coachID';  group\_awardscoaches = GROUP awardscoaches BY coachID;  count\_awardscoaches = FOREACH group\_awardscoaches GENERATE $0 as coachID,  COUNT(awardscoaches.coachID) as awardsNum;  records\_group = GROUP count\_awardscoaches ALL;  with\_max = FOREACH records\_group GENERATE MAX(count\_awardscoaches.$1);  max\_awardscoaches = FILTER count\_awardscoaches BY awardsNum == with\_max.$0;  t3\_master = load '/tmp/as2/Master.csv' using PigStorage(',');  for\_master = FOREACH t3\_master GENERATE $0 as playerID, $1 as coachID, $3 as firstname,  $4 as lastname, $18 as pos,$19 as birthYear, $20 as birthMon,$21 as birthDay,$22 as birthCountry;  master = FILTER for\_master BY playerID != 'playerID';  join\_max = JOIN max\_awardscoaches by $0,master by $1;  t3a\_final = FOREACH join\_max GENERATE $4 as firstname,$5 as lastname, $7 as birthYear,  $8 as birthMon,$9 as birthDay,$10 as birthCountry,$1 as awardsNum;  DUMP t3a\_final;  Save as pig file and upload to dropbox  URL:  https://www.dropbox.com/s/o3v9zytr10ok6mk/task3a.pig?dl=0  Using virtual machine to run the script  **pig 68217 ms**    **Tez 20360ms**    **Conclusion:**  **Using TEZ is far more faster than not using TEZ**  **T3 a) hive**  SELECT m.firstName,m.lastName,m.birthYear,m.birthMon,m.birthDay,m.birthCountry,a.awards  FROM t3\_master m  join  (  select coachid ,count(coachid) as awards  from t3\_awardscoaches  group by coachid  ) a  on (m.coachid = a.coachid)  where a.awards in  (  select max(b.countCoach)  from (  select count(coachid) as countCoach  from t3\_awardscoaches  group by coachid) b  )  ;  **With TEZ**    **Without TEZ**      **Conclusion:**  As you can see from the history log, same query with TEZ is 18 seconds and without TEZ is 100 seconds, which means TEZ can really reduce the duration of query excution.  **c)CBO**  Using Task3C’s query because Task3A only has one join.  SELECT m.firstName,m.lastName,c.year,c.games,c.win,c.lose,c.tier,a.awards  FROM t3\_master m  join  (  select coachid ,year,g as games,w as win,l as lose,t as tier  from t3\_coaches  ) c  on (m.coachid = c.coachid)  join  (  select coachid ,count(coachid) as awards  from t3\_awardscoaches  group by coachid  ) a  on (m.coachid = a.coachid)  where c.win in  (  select max(w)  from t3\_coaches  )  **ANALYZE**  ANALYZE TABLE t3\_master COMPUTE STATISTICS;  ANALYZE TABLE t3\_master COMPUTE STATISTICS FOR COLUMNS coachid,firstname,lastname;  ANALYZE TABLE t3\_coaches COMPUTE STATISTICS;  ANALYZE TABLE t3\_coaches COMPUTE STATISTICS FOR COLUMNS coachid,year,g,w,l,t;  ANALYZE TABLE t3\_awardscoaches COMPUTE STATISTICS;  ANALYZE TABLE t3\_awardscoaches COMPUTE STATISTICS FOR COLUMNS coachid;    As you can see the CBO is used on this query.  **Comparision Table:**  With TEZ and CBO    With TEZ but without CBO    Without TEZ    **Conclusion:**  Using TEZ is faster than without it, and using CBO on TEZ can save more time. CBO analyze the query before it is excuted and that save lots time for query to be excuted.    **Counter**    **DAG Graphic** |

# Part 2

**Report**

## 1. Introduction

In the past, most of the medical data were in the form of paper, such as official medical records, billing records, case records, prescription drug records, and so on. With the development of data storage, computing platforms, and mobile Internet, the current trend is that a large outbreak of medical data occurred and fast electronic digital. All the medical data mentioned above are in different degrees of digital conversion. The body of this report points out the challenges, the proposed approaches, findings and results of evaluation by analyzing all the four papers. After that, a comparison will be shown at the end of body which about the analysis results. Then the conclusion which about the future of big data about medicine will provide.

## 2. Current state of the art

### 2.1 Challenges

Dion, AbdelMalik and Mawudeku (2015) introduced the use of large data by the Global Public Health Intelligence Network (GPHIN). GPHIN detects the international infectious disease and other public health events with the big data. Their paper also determines the possible future development direction of GPHIN. Meanwhile, because of the continuous progress of big data, including the use of social media and other potential new big data sources. The analysis ability of GPHIN has been improved. Moreover, the improvement of big data enhances the role of GPHIN in the international epidemics prediction.

In the modern medical system, Schneeweiss (2014) thinks it is necessary to develop the application based on health care big data. This application should be able to automatically analyze a large number of health data sources, and then provide the data needed by the doctor, and finally gives the results of the analysis presented in the form of reports. But the medical system has not been able to integrate these applications into the work. In the health data, there is still a standardization problem existing on electronic medical records, which hinders the improvement of data processing capacity. At the same time, a lot of data is in the state of separation, the physical databases need to establish a national and even global health network to ensure the adequacy of data resources through a unified protocol.

Pang & Argyle (2016) discussed the challenging of cancer research and clinical oncology is the complexity of this disease will constantly hinder progress. But there are many hurdles described can be overcome to the benefit of all species. The main finding of this paper is data science and large data set analysis are important to the understanding of cancer at the cell and population level.  It allows the integration of clinical and biological data to improve the treatment outcomes and design to specific therapies. Also, precision medicine was coined in human medicine as a model that designed for healthcare customization, with medical decisions, practices, and tailored to individual patients. It is likely that the embrace of innovative technologies and veterinary medicine across disciplines might overcome the challenges.

In Calyam, Mishara, Antequera, Chemodanov, Berryman, Zhu, Abbott and Skubic’s paper, they presented PhysicalTherapy-as-a-Service (PTaaS) which is a 3D real-time interactive system. The therapists monitor patients’ performance of exercise assessment over high-speed network of a remote way. The authors adopted synchronous big data analytics method and showed how the therapists was able to monitor patients’ status which mainly using the features in the PTaaS interface.

### 2.2 The proposed approaches

Dion, AbdelMalik and Mawudeku (2015) through the investigation to the activities of GPHIN to show that GPHIN has made contribution in the global public health. They gave two examples in the paper, one is that GPHIN has predicted the outbreak of Severe Acute Respiratory Syndrome (SARS) in China in 2003, the other one is that GPHIN has played an important role in the outbreak of the Ebola virus in West Africa. Moreover, the paper introduced GPHIN in different time period, early detection, ongoing monitoring, the exploration in potential new data sources, to prove the importance of relation between GPHIN and big data.

Schneeweiss (2014) illustrates the importance of the big data health system through an example. This example describes the practical role of big data in the health care process. Then, he provides the functions which health care system should be put forward by the method of step by step. Finally, his paper elaborates some of the problems encountered in this system. For some major issues such as standardization and data reside in separated silos, he puts forward some feasible solutions.

The method that used in Pang and Argyle’s paper is to examine and review the changes of cancer biology in the past 25 years, which incorporating the barriers of clinical progress in veterinary medicine. An example of the paper is a certain technology, imaging technology, which has been also discussed as an improvement of embracing of technologies with human and veterinary oncology. With different technologies, the improvement of techniques to data collection and analysis would be crucial for the training of next generation of veterinarians, so that to ensure the interpretation of potentially large data sets might be associated with individual patient.

In Calyam, Mishara, Antequera, Chemodanov, Berryman, Zhu, Abbott and Skubic’s paper, experiments were carried out on PTaaS, and they describe the results in some of the objective test cases. These results describe how the PTaaS interface for remote therapy to provide help which mainly on the of motion reaction, sway, time and timing of movements range for real-time analysis. Then these analysis feedback were provided to the patients and the therapists. After that, the therapists carry on the treatment to the patients through the feedback and guide the exercise of patients. Finally, they assess the PTaaS through the survey method to obtain the evaluation of user satisfaction and treatment effect.

### 2.3 Findings and results

Dion, AbdelMalik and Mawudeku (2015) found that GPHIN through the analysis of massive data, access to valuable products and services, has contributed to the public health in the global. The world will have a better forecasting tool to prevent the spread of the disease for the next international pandemic, based on the idea of such a technology and the data store technology. Integration of additional Big Data sources and advances in analytical capacity could further strengthen the GPHIN's capability for timely detection and early warning.

Schneeweiss (2014) provided the final target of this application is acquiring actionable insights into the great ways to treat the patients with data which generated in the care application. Focusing on target is a key to success in learning from big health care data. This system needs the correct analysis method and the necessary data management, but also needs to pay attention to the privacy and the standardization of the data.

One of the major contribution of Pang and Argyle’s (2016) paper is to emphasize the importance of technological development to data science project to a specific discipline. It also brings benefits to the whole community for thinking beyond the basic biological practices, but more than that to take research and applicant into account for increasing benefits to patients.

Calyam, Mishara, Antequera, Chemodanov, Berryman, Zhu, Abbott and Skubic’s (2016) results showed that they were able to distinguish scenarios where there was bias due to network quality effects, and thus enabled the therapist to successfully quantify and assess balance of the senior study participants we recruited in private homes. Remote physical therapy can be used in the future to a wide range of health care, which can improve medical efficiency definitely. Because of the requirement of high speed network, it depends on the development of the network, the system also faces many challenges.

### 2.4 Compare

All the papers above describe the relation between big data technology and medicine development. Calyam, Mishara, Antequera, Chemodanov, Berryman, Zhu, Abbott and Skubic’s paper and Schneeweiss’s paper provide the applications about how to improve the health care level with the help of big data system. The health care application and tool PTaaS both has the function about the health care data store and analysis, which can make the use of data more efficient and both of them has good prospects.  Dion, AbdelMalik and Mawudeku introduced the GPHIN in their paper, it is an Internet-based tool which use for the disease prediction. This tool also works for the medicine area, in different time period, early detection, ongoing monitoring, and the exploration in potential new data sources. In my point of view, because it plays a significant role in disease prevention, GPHIN worth of popularizing and applying. I summed it up in Pang and Argyle’s paper that data science and large data set analysis will be vital to understanding the complexity of cancer at the cell and population level. This is in the medicine area about cancer and clinical oncology which will reverse action in human health.

## 3. Conclusion

In summary, no matter developing application or building some tools or systems, all of them focus on using the big data to improve the medicine. Health service providers analyze clinical data, personal health data and public health data to mining potential relationship, to provide help for public health and personal health. At present, the medical health big data also faces many challenges, data security and standardization need to be managed by specialized agencies. However, with the development of technology, the combination of medical technology and big data will provide better service for human health.

## 4. References

1. Dion, M., AbdelMalik, P., & Mawudeku, A. (2015). Big data and the global public health intelligence network (GPHIN). *Canada Communicable Disease Report, 41*(9), 209-214.
2. Schneeweiss, S. (2014). Learning from big health care data. *The New England Journal of Medicine, 370*(23), 2161-3.
3. Calyam, P., Mishra, A., Antequera, R. B., Chemodanov, D., Berryman, A., Zhu, K., … Skubic, M. (2016). Synchronous big data analytics for personalized and remote physical therapy. *Pervasive and Mobile Computing*, *28*, 3–20.
4. Pang, L. Y., & Argyle, D. J. (2016). Veterinary oncology: Biology, big data and precision medicine.*The Veterinary Journal*, *213*, 38–45.
5. Dhar, V. (2014). Big data and predictive Analytics in health care. *Big Data*, 2(3), 113–116.
6. Herland, M., Khoshgoftaar, T. M., & Wald, R. (2014). A review of data mining using big data in health informatics. J*ournal Of Big Data*, 1(1), 2.
7. Big data for cancer research (2015). Clinical Medicine Insights: Oncology. doi:10.4137/cmo.s38279