



US 20070060325A1

(19) **United States**

(12) **Patent Application Publication**
Gradek

(10) **Pub. No.: US 2007/0060325 A1**

(43) **Pub. Date: Mar. 15, 2007**

(54) **FANTASY SPORTS LIVE**

Publication Classification

(75) Inventor: **Michael Gradek**, Redmond, WA (US)

(51) **Int. Cl.**
A63F 9/24 (2006.01)

Correspondence Address:
AMIN, TUROCY & CALVIN, LLP
24TH FLOOR, NATIONAL CITY CENTER
1900 EAST NINTH STREET
CLEVELAND, OH 44114 (US)

(52) **U.S. Cl.** **463/29**

(57) **ABSTRACT**

A live scoring that can be employed without plugins thereby enhancing reliability and scalability is provided. Moreover, the live scoring system utilizes a pull model which does not require persistent connections to a live scoring source, server or network. Furthermore, in view of the inherent characteristic of live scoring systems being resource intensive applications, e.g., require a large amount of processing power to calculate the scores of all players for all teams across all leagues, a client-side scoring/calculation system is employed. The distributed approach used by the system can particularly offload the burden of processing the updates most of the time to clients, without sacrificing down-level browsers' ability to view the latest scores.

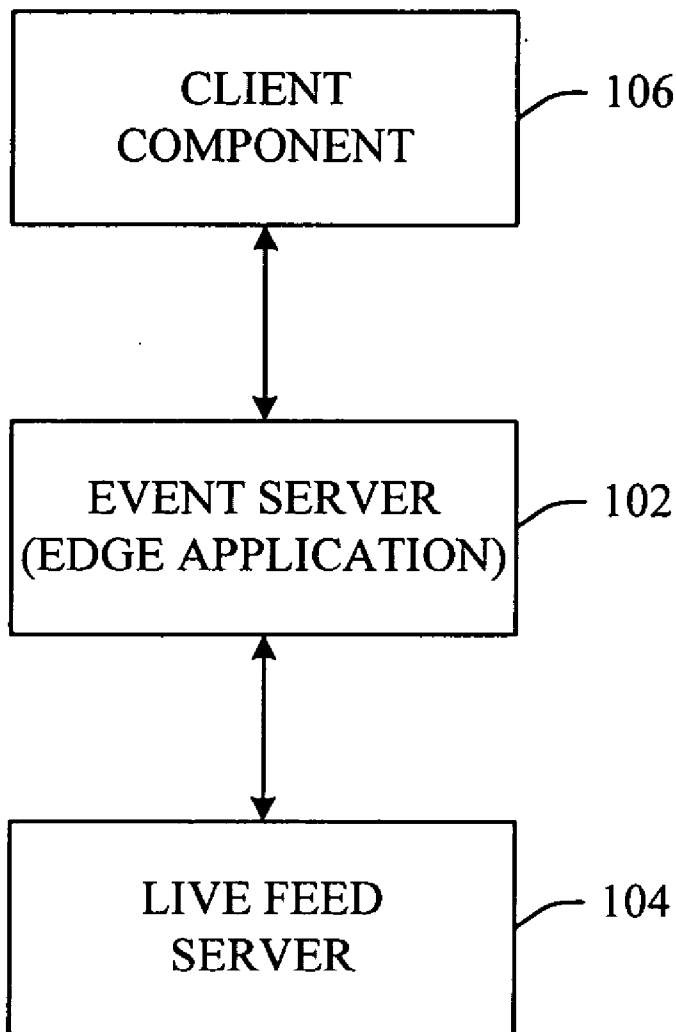
(73) Assignee: **Microsoft Corporation**, Redmond, WA

(21) Appl. No.: **11/327,046**

(22) Filed: **Jan. 6, 2006**

Related U.S. Application Data

(60) Provisional application No. 60/711,259, filed on Aug. 25, 2005.



100

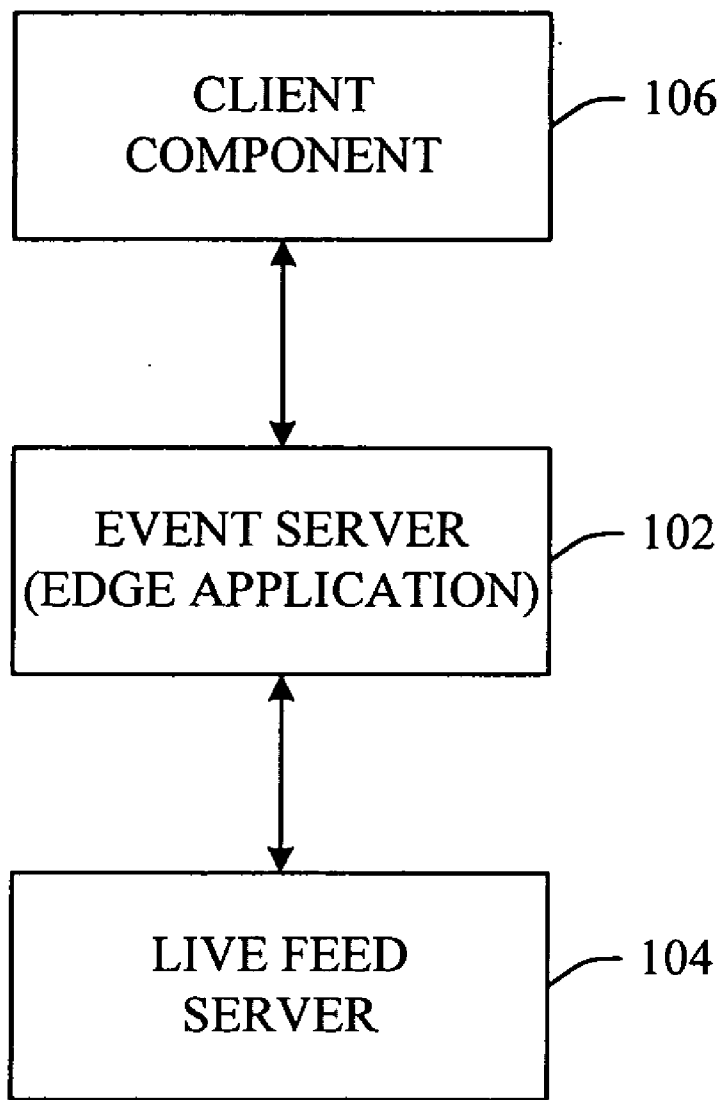


FIG. 1

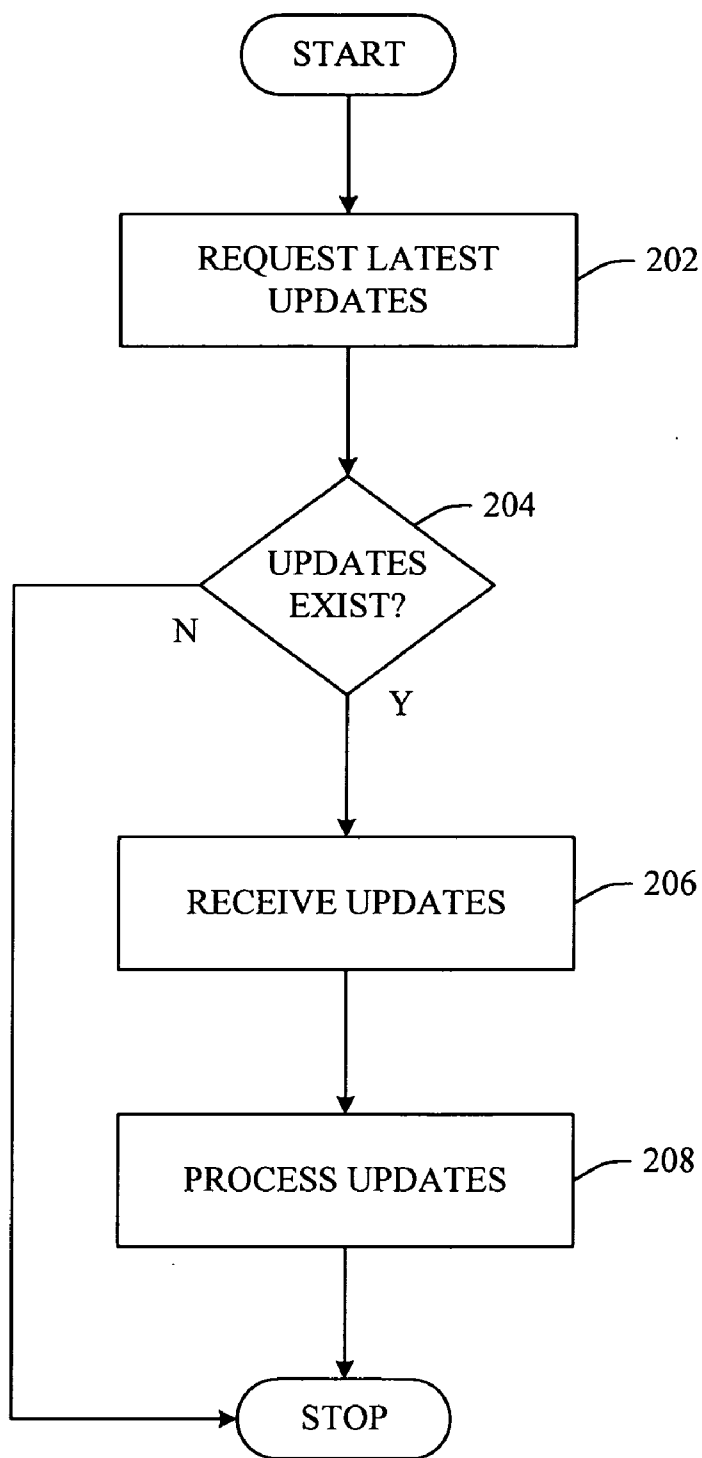


FIG. 2

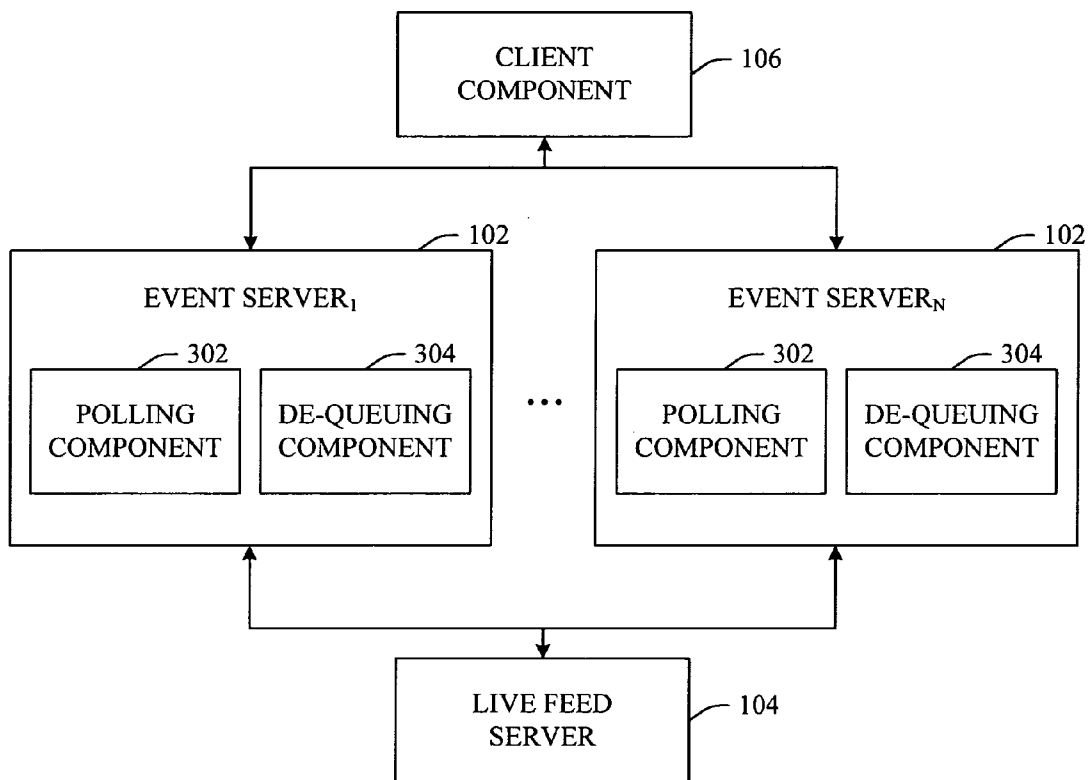


FIG. 3

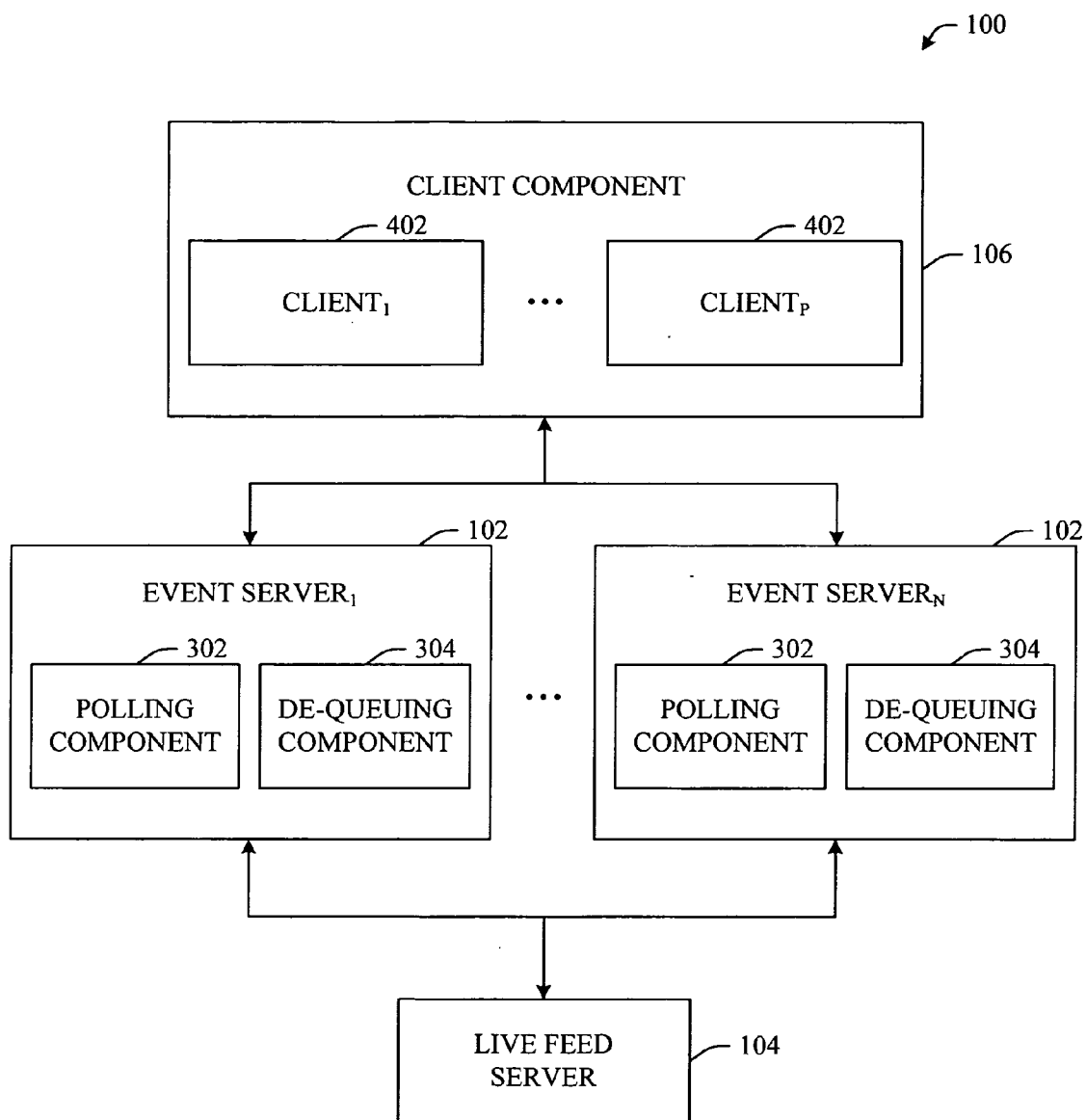


FIG. 4

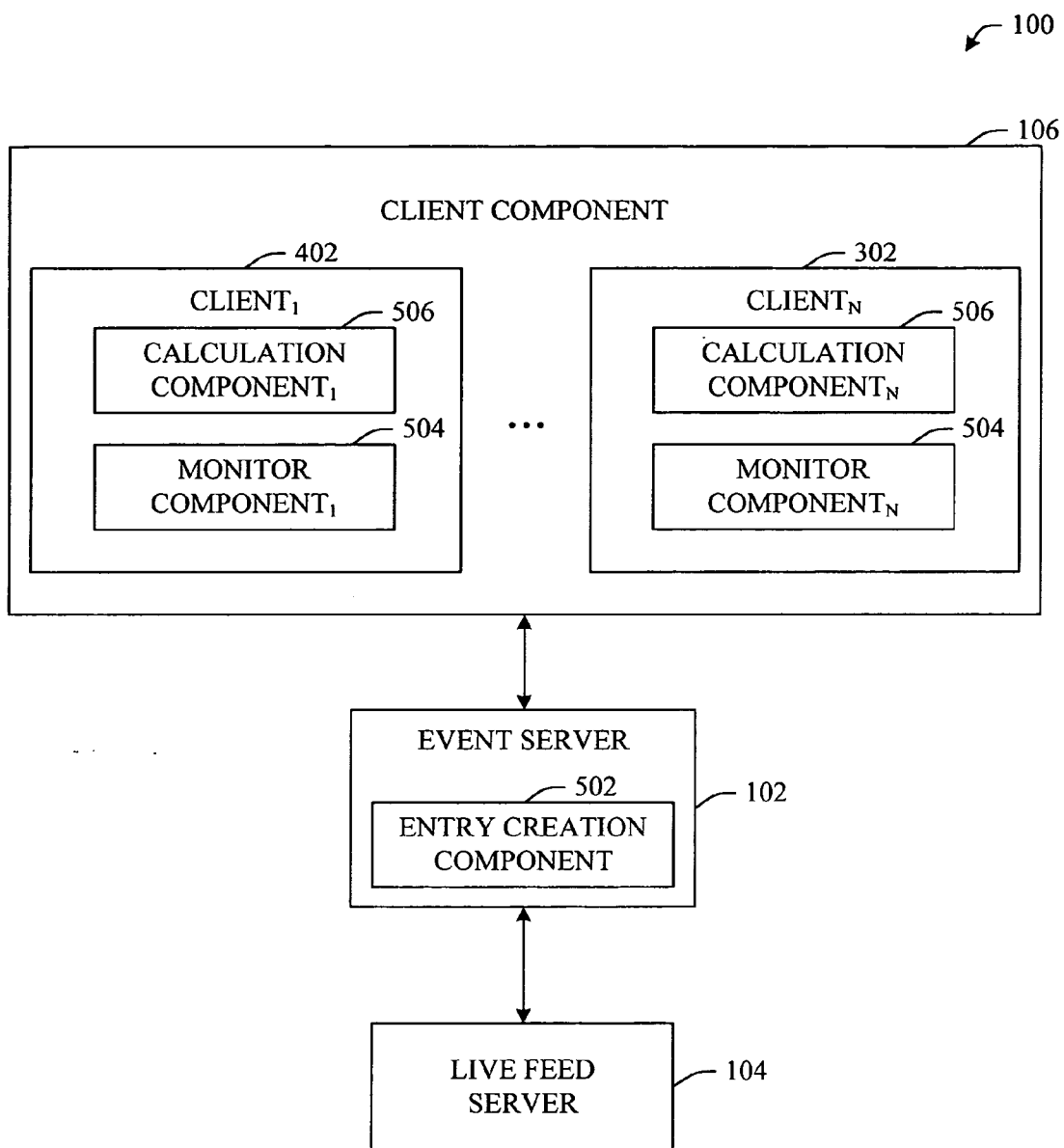


FIG. 5

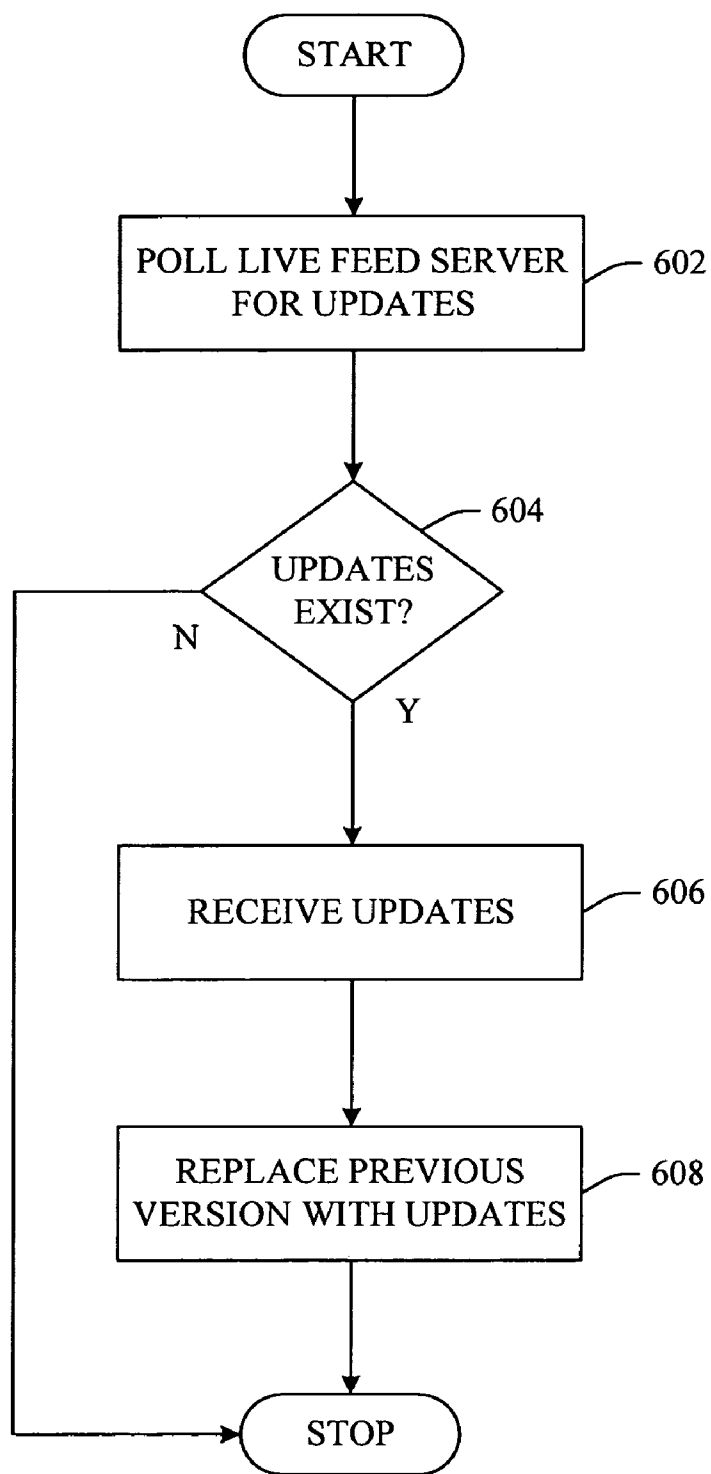


FIG. 6

700

LEAGUE HOME MY TEAM SCORES AND RESULTS PLAYERS RESEARCH COMMISH TOOLS

Fat Slobs 121 at Double Deuce 102

Game Trax · Head to Head · Recap · Trophy Room · Bulletin Board

SLOBS 132 RIESS 111 Final	QUINCY 279 STRAW 104 In progress	BT DBL 2S 27	TEN 10 MIN 27 10:45 4th	TEN 20 MIN 27 10:45 4th	TEN 30 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th
SLOBS 132 RIESS 111 Final	QUINCY 279 STRAW 104 In progress	BT DBL 2S 27	TEN 10 MIN 27 10:45 4th	TEN 20 MIN 27 10:45 4th	TEN 30 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th

Fat Slobs (9-3-1)
Yet to Play: 4/9
Minutes Remaining: 240

Pos.	Starters	Status	Pts
QB	P. Manning (IND - RB)	@MIN Final	34
RB	P. Holmes (KC - RB)	MIA Final	11

The Noppers (7-4-0)
Yet to Play: 6/9
Minutes Remaining: 300

Pos.	Starters	Status	Pts
QB	P. Manning (IND - RB)	@MIN Final	34
RB	P. Holmes (KC - RB)	MIA Final	11

Scoring Plays			
P. Holmes	11 RuYd, 1 pt.	P. Manning (IND - RB)	313 PaYds, 4 TD, 1 Int
P. Holmes	11 RuYd, 1 pt.	P. Holmes (KC - RB)	59 RuYd, 1 TD

FIG. 7

800

ML TEAM SCORES AND RESULTS PLAYERS RESEARCH COMMISSION TOOLS

t Double Deuce

to Head · Recap · Trophy Room · Bulletin Board

Y279	BT	24	TEN	10	TEN	20	TEN	30	TEN	24	TEN	24	TEN	24	TEN	24	TEN	24
v 104	DBL 2S	27	MIN	27	MIN	27	MIN	27	MIN	27	MIN	27	MIN	27	MIN	27	MIN	27
grass			10:45	4th	10:45	4th	10:45	4th	10:45	4th	10:45	4th	10:45	4th	10:45	4th	10:45	4th

e first year of toughened steroid testing, home runs major leagues for the first time since 2002.

er Todd Jones doesn't think it's a coincidence. He's connection.

I hate it, but there has been a correction made in the numbers are going to suffer for a couple of years," he said admit it because I didn't want to. I'm as disappointed as at it's going to end up showing to be the truth. But it's got lame to get back to an even playing field. I just didn't was."

home runs were hit in games through Sunday, down 8.8 16 average in the first five weeks of last season, according bureau. It's the lowest level for the first five weeks since and below the 2.14 average of the last decade.

is too short a statistical sample to draw any conclusions," seball's chief operating officer. "There are a myriad of fluence that, including the cold and wet spring training we weather in the first few weeks of the season."

gels bench coach Joe Maddon, who has been in ill since 1975, thinks testing "could be tied to it somehow."

ly to what extent, but you see people maybe not as big as e past or don't look as strong as maybe they had been in Some balls are hit pretty well, but they're not carrying like the past couple of years. So, just being a thinking person, 's been going on, you have to tie them together a little bit."

down 5 percent, from 9.72 to 9.23, and hits declined 3.1 per game to 17.80. The major-league batting average

Team	W-L-T	Pts	Strk
Fat Slobs	9-6-4	5412	W6
Double Deuce	9-6-4	5412	W6

Pos	Fat Slobs	Double Deuce
QB	B. Roethlisberger	B. Favre
RB	B. Roethlisberger	B. Favre
RB	B. Roethlisberger	B. Favre
WR	B. Roethlisberger	B. Favre
WR	B. Roethlisberger	B. Favre
WR	B. Roethlisberger	B. Favre
TE	B. Roethlisberger	B. Favre
K	B. Roethlisberger	B. Favre
D/ST	Rams	Buccaneers

Player	Injury
B. Roethlisberger (PIT - QB)	QUESTIONABLE Pulled hamstring
B. Roethlisberger (PIT - QB)	OUT Tom ACL

FIG. 8

900

Fat Slobs 121 at Double Deuce 102

Game Trax - Head to Head - Recap - Trophy Room - Bulletin Board

SLOBS 132	QUINCY 279	BT DBL 2S 24	TEN 10	TEN 20	TEN 30	TEN 24	TEN 24	TEN 24	TEN 24	TEN 24
RIESS 111	STRAW 104	DBL 2S 27	MIN 27	MIN 27	MIN 27	MIN 27	MIN 27	MIN 27	MIN 27	MIN 27
Final	In-progress		10:45 4th	10:45 4th	10:45 4th	10:45 4th	10:45 4th	10:45 4th	10:45 4th	10:45 4th

SLOBS 132	QUINCY 279	BT DBL 2S 27	TEN 10	TEN 20	TEN 30	TEN 24	TEN 24	TEN 24	TEN 24	TEN 24
RIESS 111	STRAW 104	DBL 2S 27	MIN 27	MIN 27	MIN 27	MIN 27	MIN 27	MIN 27	MIN 27	MIN 27
Final	In-progress		10:45 4th	10:45 4th	10:45 4th	10:45 4th	10:45 4th	10:45 4th	10:45 4th	10:45 4th



Fat Slobs (9-3-1)
 Yet to Play: 4/9
 Minutes Remaining: 240



The Hoppers (7-4-0)
 Yet to Play: 6/9
 Minutes Remaining: 300

Pos.	Starters	Status	Pts	Pos.	Starters	Status	Pts
QB	P. Manning (IND - RB)	@MIN Final	34	QB	P. Manning (IND - RB)	@MIN Final	34
	313 PaYds, 4 TD, 1 Int				313 PaYds, 4 TD, 1 Int		
RB	P. Holmes (KC - RB)	MIA Final	11	RB	P. Holmes (KC - RB)	MIA Final	11
	59 RuYd, 1 TD				59 RuYd, 1 TD		
RB	E. James (IND - RB)	@WAS 1:00	7	RB	E. James (IND - RB)	@WAS 1:00	7
	76 RuYd				76 RuYd		
WR	P. Burress (PIT - WR)	SEA 1:00	16	WR	P. Burress (PIT - WR)	SEA 1:00	16
	103 ReYd, 1 TD				103 ReYd, 1 TD		
WR	E. Kennison (KC - WR)	@OAK 1:00	9	WR	E. Kennison (KC - WR)	@OAK 1:00	9
	111 ReYd, 1 FL				111 ReYd, 1 FL		
WR	T. Holt (STL - WR)	GB 12:32 1st	12	WR	T. Holt (STL - WR)	GB 12:32 1st	12
	66 ReYd, 1 TD				66 ReYd, 1 TD		
TE	T. Gonzalez (KC - TE)	DAL 0:48 2nd	8	TE	T. Gonzalez (KC - TE)	DAL 0:48 2nd	8
	88 ReYd				88 ReYd		
K	S. Janikowski (OAK - K)	@KC 8:00	11	K	S. Janikowski (OAK - K)	@KC 8:00	11
	3/3 FG, 2/2 XP				3/3 FG, 2/2 XP		
D/ST	Steelers	IND 2:00 4th	19	D/ST	Steelers	IND 2:00 4th	19
	202 YdsA, 2 Ints, 1 DTD, 3 PtsA				202 YdsA, 2 Ints, 1 DTD, 3 PtsA		
Total			121	Total			121

Pos.	Bench	Status	Pts	Pos.	Bench	Status	Pts
BN	P. Manning (IND - RB)	@MIN Final	3	BN	P. Manning (IND - RB)	@MIN Final	3
	313 PaYds, 4 TD, 1 Int				313 PaYds, 4 TD, 1 Int		
BN	P. Holmes (KC - RB)	MIA Final	6	BN	P. Holmes (KC - RB)	MIA Final	6
	59 RuYd, 1 TD				59 RuYd, 1 TD		
BN	E. James (IND - RB)	@WAS 1:00	7	BN	E. James (IND - RB)	@WAS 1:00	7
	76 RuYd				76 RuYd		
BN	P. Burress (PIT - WR)	SEA 1:00	2	BN	P. Burress (PIT - WR)	SEA 1:00	2
	76 RuYd				76 RuYd		
BN	P. Burress (PIT - WR)	SEA 1:00	6	BN	P. Burress (PIT - WR)	SEA 1:00	6
	76 RuYd				76 RuYd		
BN	P. Burress (PIT - WR)	SEA 1:00	4	BN	P. Burress (PIT - WR)	SEA 1:00	4
	76 RuYd				76 RuYd		
Total			28	Total			28


FIG. 9


1000

Fat Slobz 121 at Double Deuce 102

Game Trax · [Head to Head](#) · [Recap](#) · [Trophy Room](#) · [Bulletin Board](#)

SLOBS 132 RIESS 111 Final	QUINCY279 STRAW 104 In-progress	BT DBL 2S 27	TEN 10 MIN 27 10:45 4th	TEN 20 MIN 27 10:45 4th	TEN 30 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th
SLOBS 132 RIESS 111 Final	QUINCY279 STRAW 104 In-progress	BT DBL 2S 27	TEN 10 MIN 27 10:45 4th	TEN 20 MIN 27 10:45 4th	TEN 30 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th

 **Fat Slobz (9-3-1)**
 Yet to Play: 4/9
 Minutes Remaining: 240

 **The Hoppers (7-4-0)**
 Yet to Play: 6/9
 Minutes Remaining: 300

Pos.	Starters	Status	Pts
QB	<u>P. Manning (IND - RB)</u> 313 PaYds, 4 TD, 1 Int	@MIN Final	34
RB	<u>P. Holmes (KC - RB)</u> 59 RuYd, 1 TD	MIA Final	11
RB	<u>E. James (IND - RB)</u> 76 RuYd	@WAS 1:00	7
WR	<u>P. Burruss (PIT - WR)</u> 103 ReYd, 1 TD	SEA 1:00	16
WR	<u>E. Kennison (KC - WR)</u> 111 ReYd, 1 FL	@OAK 1:00	9
WR	<u>T. Holt (STL - WR)</u> 66 ReYd, 1 TD	GB 12:32 1st	12
TE	<u>T. Gonzalez (KC - TE)</u> 88 ReYd	DAL 0:48 2nd	8
K	<u>S. Janikowski (OAK - K)</u> 3/3 FG, 2/2 XP	@KC 8:00	11
D/ST	<u>Steelers</u> 202 YdsA, 2 Ints, 1 DTD, 3 PtsA	IND 2:00 4th	19
Total			121

Scoring Plays

- P. Holmes 11 RuYd, 1 pt
- P. Holmes 11 RuYd, 1 pt
- P. Holmes 11 RuYd, 1 pt
- P. Holmes 11 RuYd, 1 pt
- P. Holmes 11 RuYd, 1 pt
- P. Holmes 11 RuYd, 1 pt
- P. Holmes 11 RuYd, 1 pt
- P. Holmes 11 RuYd, 1 pt
- P. Holmes 11 RuYd, 1 pt

Pos.	Starters	Status	Pts
QB	<u>P. Manning (IND - RB)</u> 313 PaYds, 4 TD, 1 Int	@MIN Final	34
RB	<u>P. Holmes (KC - RB)</u> 59 RuYd, 1 TD	MIA Final	11
RB	<u>E. James (IND - RB)</u> 76 RuYd	@WAS 1:00	7
WR	<u>P. Burruss (PIT - WR)</u> 103 ReYd, 1 TD	SEA 1:00	16
WR	<u>E. Kennison (KC - WR)</u> 111 ReYd, 1 FL	@OAK 1:00	9
WR	<u>T. Holt (STL - WR)</u> 66 ReYd, 1 TD	GB 12:32 1st	12
TE	<u>T. Gonzalez (KC - TE)</u> 88 ReYd	DAL 0:48 2nd	8
K	<u>S. Janikowski (OAK - K)</u> 3/3 FG, 2/2 XP	@KC 8:00	11
D/ST	<u>Steelers</u> 202 YdsA, 2 Ints, 1 DTD, 3 PtsA	IND 2:00 4th	19
Total			121

Pos.	Bench	Status	Pts
BN	<u>P. Manning (IND - RB)</u> 313 PaYds, 4 TD, 1 Int	@MIN Final	3
BN	<u>P. Holmes (KC - RB)</u> 59 RuYd, 1 TD	MIA Final	6
BN	<u>E. James (IND - RB)</u> 76 RuYd	@WAS 1:00	7
BN	<u>P. Burruss (PIT - WR)</u> 76 RuYd	SEA 1:00	2
BN	<u>P. Burruss (PIT - WR)</u> 76 RuYd	SEA 1:00	6
BN	<u>P. Burruss (PIT - WR)</u> 76 RuYd	SEA 1:00	4

Pos.	Bench	Status	Pts
BN	<u>P. Manning (IND - RB)</u> 313 PaYds, 4 TD, 1 Int	@MIN Final	3
BN	<u>P. Holmes (KC - RB)</u> 59 RuYd, 1 TD	MIA Final	6
BN	<u>E. James (IND - RB)</u> 76 RuYd	@WAS 1:00	7
BN	<u>P. Burruss (PIT - WR)</u> 76 RuYd	SEA 1:00	2
BN	<u>P. Burruss (PIT - WR)</u> 76 RuYd	SEA 1:00	6
BN	<u>P. Burruss (PIT - WR)</u> 76 RuYd	SEA 1:00	4

FIG. 10




1100

Fat Slobz 121 at Double Deuce 102

Game Trax · Head to Head · Recap · Trophy Room · Bulletin Board


SLOBS 132	QUINCY 279	BT 24	TEN 10	TEN 20	TEN 30	TEN 24	TEN 24	TEN 24	TEN 24	TEN 24
RIESS 111	STRAW 104	DBL 2S 27	MIN 27	MIN 27	MIN 27	MIN 27	MIN 27	MIN 27	MIN 27	MIN 27
Final	In-progress		10:45 4th	10:45 4th	10:45 4th	10:45 4th	10:45 4th	10:45 4th	10:45 4th	10:45 4th


 **Fat Slobz (9-3-1)**
 Power Ranking Trophy

 **The Hoppers (7-4-0)**
 Earl Credit Trophy
 Training Staff Trophy

Pos.	Starters	Status	Pts
QB	P. Manning (IND - RB)	@MIN Final	34
	313 PaYds, 4 TD, 1 Int		
RB	P. Holmes (KC - RB)	MIA Final	11
	59 RuYd, 1 TD		
RB	E. James (IND - RB)	@WAS 1:00	7
	76 RuYd		
WR	P. Burress (PIT - WR)	SEA 1:00	16
	103 ReYd, 1 TD		
WR	E. Kennison (KC - WR)	@OAK 1:00	9
	111 ReYd, 1 FL		
WR	T. Holt (STL - WR)	GB 12:32 1st	12
	66 ReYd, 1 TD		
TE	T. Gonzalez (KC - TE)	DAL 0:48 2nd	8
	88 ReYd		
K	S. Janikowski (OAK - K)	@KC 8:00	11
	3/3 FG, 2/2 XP		
D/ST	Steelers	IND 2:00 4th	19
	202 YdsA, 2 Ints, 1 DTD, 3 PtsA		
Total			121

Players of the Game

P. Manning
 21 Pts, 315 PaYd, 2 PaTD

A. Housmanzadeh
 35 Pts, 135 RuYd, 3 RuTD, 77 ReYd

Pos.	Starters	Status	Pts
QB	P. Manning (IND - RB)	@MIN Final	34
	313 PaYds, 4 TD, 1 Int		
RB	P. Holmes (KC - RB)	MIA Final	11
	59 RuYd, 1 TD		
RB	E. James (IND - RB)	@WAS 1:00	7
	76 RuYd		
WR	P. Burress (PIT - WR)	SEA 1:00	16
	103 ReYd, 1 TD		
WR	E. Kennison (KC - WR)	@OAK 1:00	9
	111 ReYd, 1 FL		
WR	T. Holt (STL - WR)	GB 12:32 1st	12
	66 ReYd, 1 TD		
TE	T. Gonzalez (KC - TE)	DAL 0:48 2nd	8
	88 ReYd		
K	S. Janikowski (OAK - K)	@KC 8:00	11
	3/3 FG, 2/2 XP		
D/ST	Steelers	IND 2:00 4th	19
	202 YdsA, 2 Ints, 1 DTD, 3 PtsA		
Total			121

Pos.	Bench	Status	Pts
BN	P. Manning (IND - RB)	@MIN Final	3
	313 PaYds, 4 TD, 1 Int		
BN	P. Holmes (KC - RB)	MIA Final	6
	59 RuYd, 1 TD		
BN	E. James (IND - RB)	@WAS 1:00	7
	76 RuYd		
BN	P. Burress (PIT - WR)	SEA 1:00	2
	76 RuYd		
BN	P. Burress (PIT - WR)	SEA 1:00	6
	76 RuYd		
BN	P. Burress (PIT - WR)	SEA 1:00	4
	76 RuYd		
Total			28

Pos.	Bench	Status	Pts
BN	P. Manning (IND - RB)	@MIN Final	3
	313 PaYds, 4 TD, 1 Int		
BN	P. Holmes (KC - RB)	MIA Final	6
	59 RuYd, 1 TD		
BN	E. James (IND - RB)	@WAS 1:00	7
	76 RuYd		
BN	P. Burress (PIT - WR)	SEA 1:00	2
	76 RuYd		
BN	P. Burress (PIT - WR)	SEA 1:00	6
	76 RuYd		
BN	P. Burress (PIT - WR)	SEA 1:00	4
	76 RuYd		
Total			28

FIG. 11

1200

Pat Slobs at Double Deuce

Game Tray · Head to Head · Preview · Trophy Room · Bulletin Board

SLOBS 132 RIESS 111 Final	QUINCY 279 STRAW 104 In-progress	BT DBL 2S 24 27	TEN 10 MIN 27 10:45 4th	TEN 20 MIN 27 10:45 4th	TEN 30 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th
SLOBS 132 RIESS 111 Final	QUINCY 279 STRAW 104 In-progress	BT DBL 2S 24 27	TEN 10 MIN 27 10:45 4th	TEN 20 MIN 27 10:45 4th	TEN 30 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th	TEN 24 MIN 27 10:45 4th

Ent Slobs (9-3-1) Season Pts: 453		Ent Slobs (9-3-1) Season Pts: 453	
Starters		Pts.	Fav.
QB	B. Favrrrr (QB - QB) Even if it takes him a while to really get going, Berkman's return will give the Astros offense a huge lift. He's batting third and playing first base with Jeff Bagwell still out tonight. Mike Lamb remains in left field. Last Wk: 241 PaYd, 2 TD, 22 RuYd, 3 Int Proj.: 275 PaYd, 3 TD, 10 RuYd, 1 Int	18	
RB	B. Favrrrr (QB - QB) Even if it takes him a while to really get going, Berkman's return will give the Astros offense a huge lift. He's batting third and playing first base with Jeff Bagwell still out tonight. Mike Lamb remains in left field. Last Wk: 241 PaYd, 2 TD, 22 RuYd, 3 Int Proj.: 275 PaYd, 3 TD, 10 RuYd, 1 Int	18	✓
RB	B. Favrrrr (QB - QB) Even if it takes him a while to really get going, Berkman's return will give the Astros offense a huge lift. He's batting third and playing first base with Jeff Bagwell still out tonight. Mike Lamb remains in left field. Last Wk: 241 PaYd, 2 TD, 22 RuYd, 3 Int Proj.: 275 PaYd, 3 TD, 10 RuYd, 1 Int	18	
WR	B. Favrrrr (QB - QB) Even if it takes him a while to really get going, Berkman's return will give the Astros offense a huge lift. He's batting third and playing first base with Jeff Bagwell still out tonight. Mike Lamb remains in left field. Last Wk: 241 PaYd, 2 TD, 22 RuYd, 3 Int Proj.: 275 PaYd, 3 TD, 10 RuYd, 1 Int	18	✓
WR	B. Favrrrr (QB - QB) Even if it takes him a while to really get going, Berkman's return will give the Astros offense a huge lift. He's batting third and playing first base with Jeff Bagwell still out tonight. Mike Lamb remains in left field. Last Wk: 241 PaYd, 2 TD, 22 RuYd, 3 Int Proj.: 275 PaYd, 3 TD, 10 RuYd, 1 Int	18	✓
WR	B. Favrrrr (QB - QB) Even if it takes him a while to really get going, Berkman's return will give the Astros offense a huge lift. He's batting third and playing first base with Jeff Bagwell still out tonight. Mike Lamb remains in left field. Last Wk: 241 PaYd, 2 TD, 22 RuYd, 3 Int Proj.: 275 PaYd, 3 TD, 10 RuYd, 1 Int	18	
TE	B. Favrrrr (QB - QB) Even if it takes him a while to really get going, Berkman's return will give the Astros offense a huge lift. He's batting third and playing first base with Jeff Bagwell still out tonight. Mike Lamb remains in left field. Last Wk: 241 PaYd, 2 TD, 22 RuYd, 3 Int Proj.: 275 PaYd, 3 TD, 10 RuYd, 1 Int	18	✓
K	B. Favrrrr (QB - QB) Even if it takes him a while to really get going, Berkman's return will give the Astros offense a huge lift. He's batting third and playing first base with Jeff Bagwell still out tonight. Mike Lamb remains in left field. Last Wk: 241 PaYd, 2 TD, 22 RuYd, 3 Int Proj.: 275 PaYd, 3 TD, 10 RuYd, 1 Int	18	✓
	Falcons Even if it takes him a while to really get going, Berkman's return will give the Astros		

FIG. 12

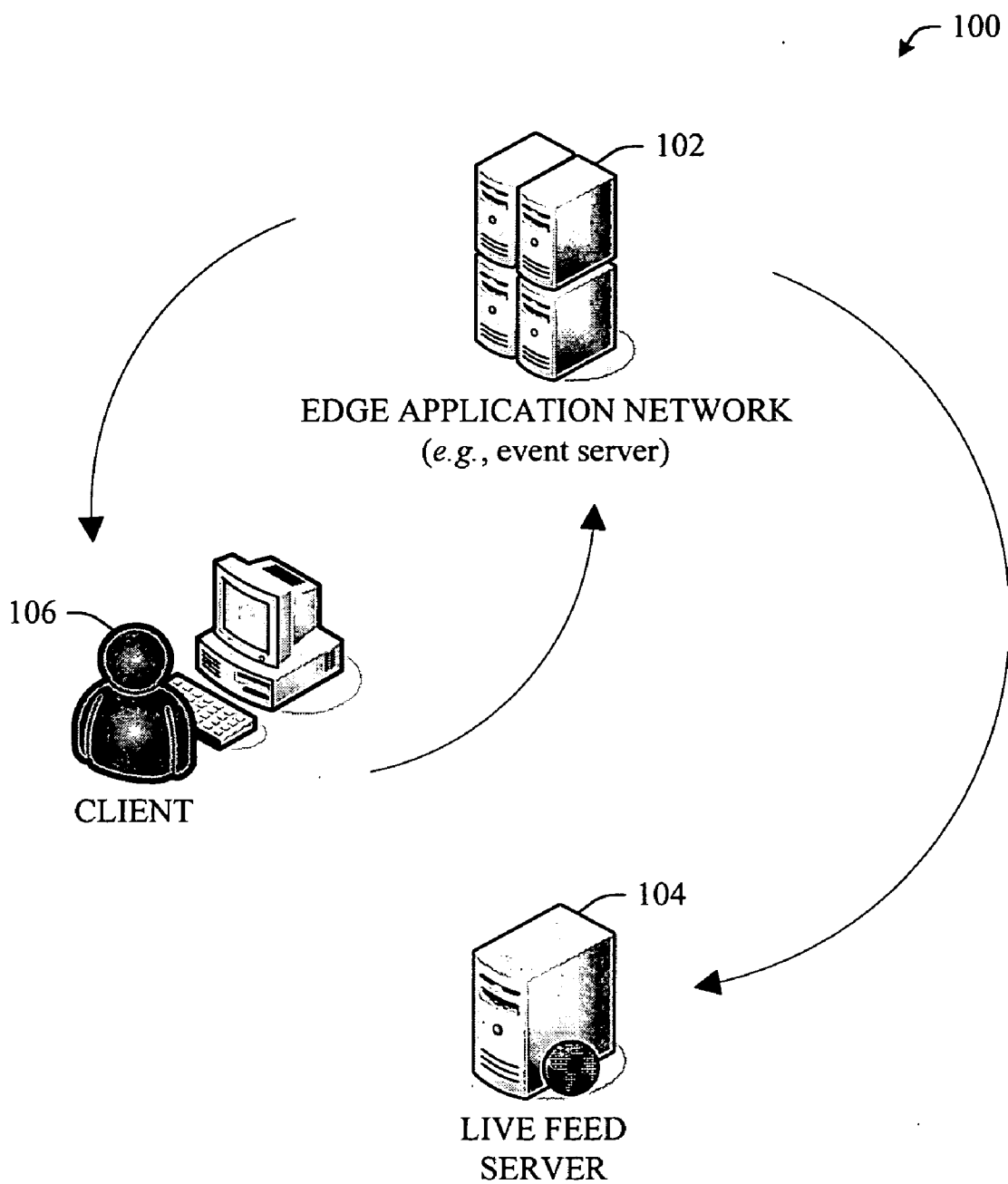


FIG. 13

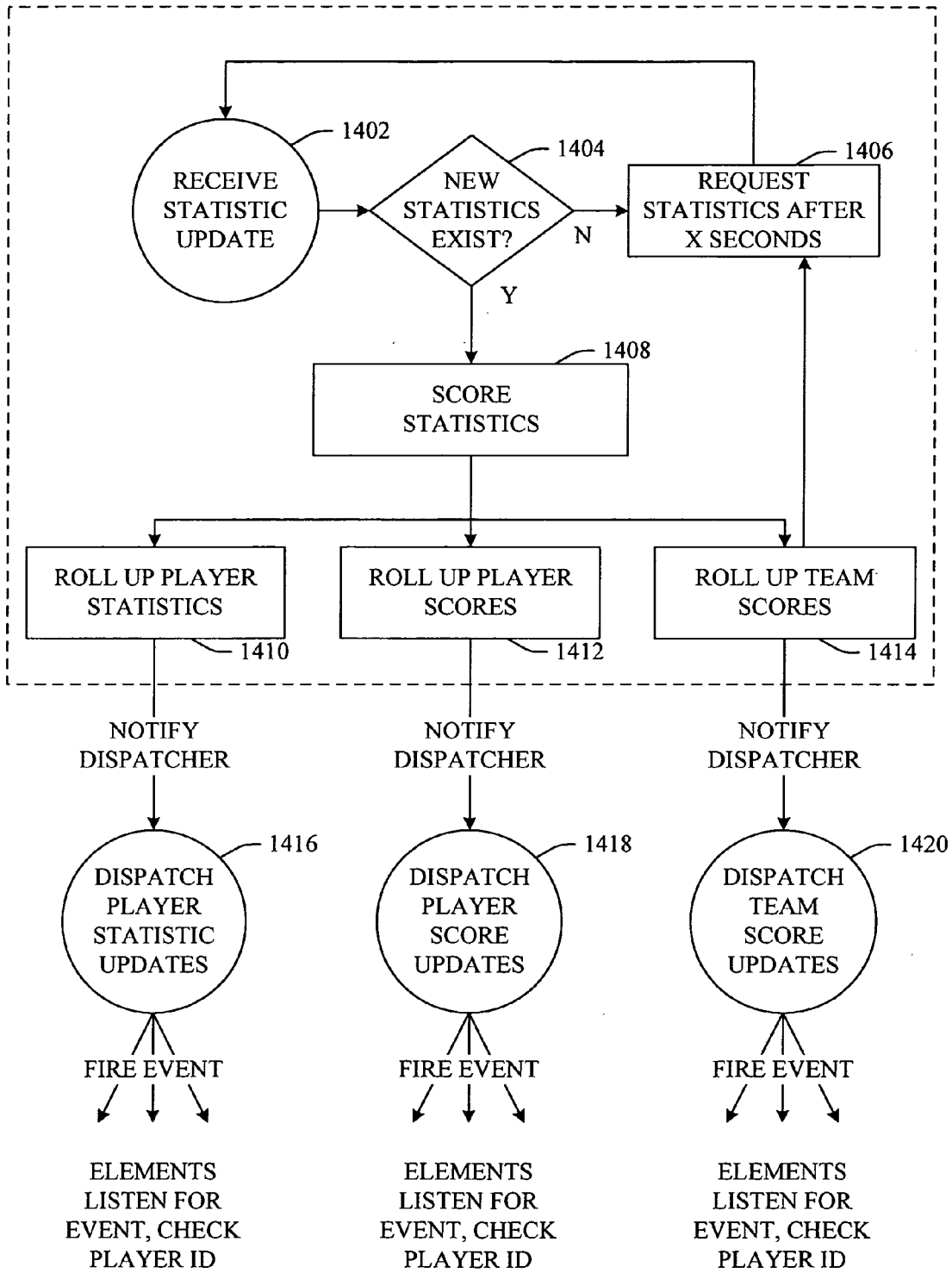


FIG. 14

1500

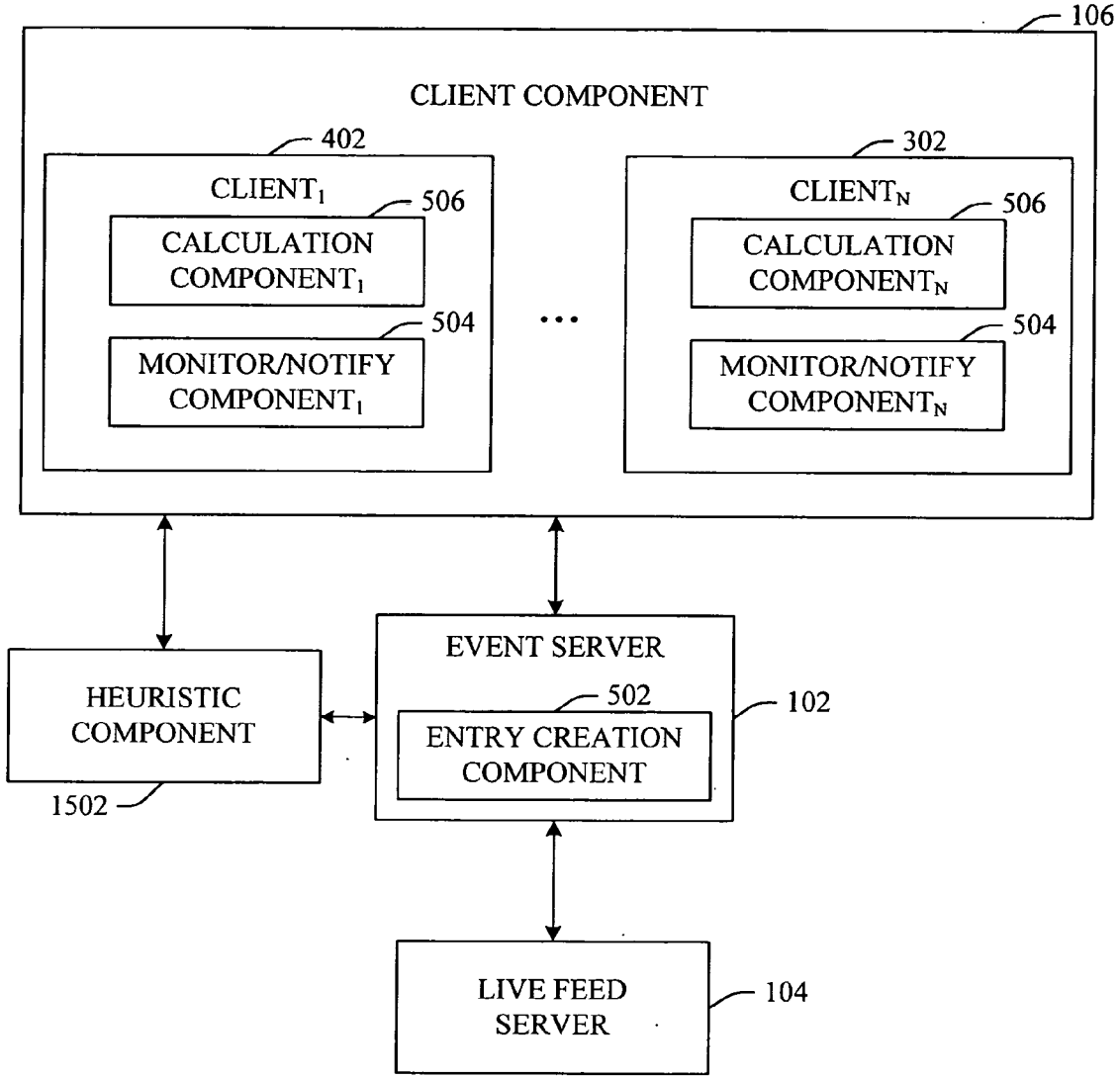


FIG. 15

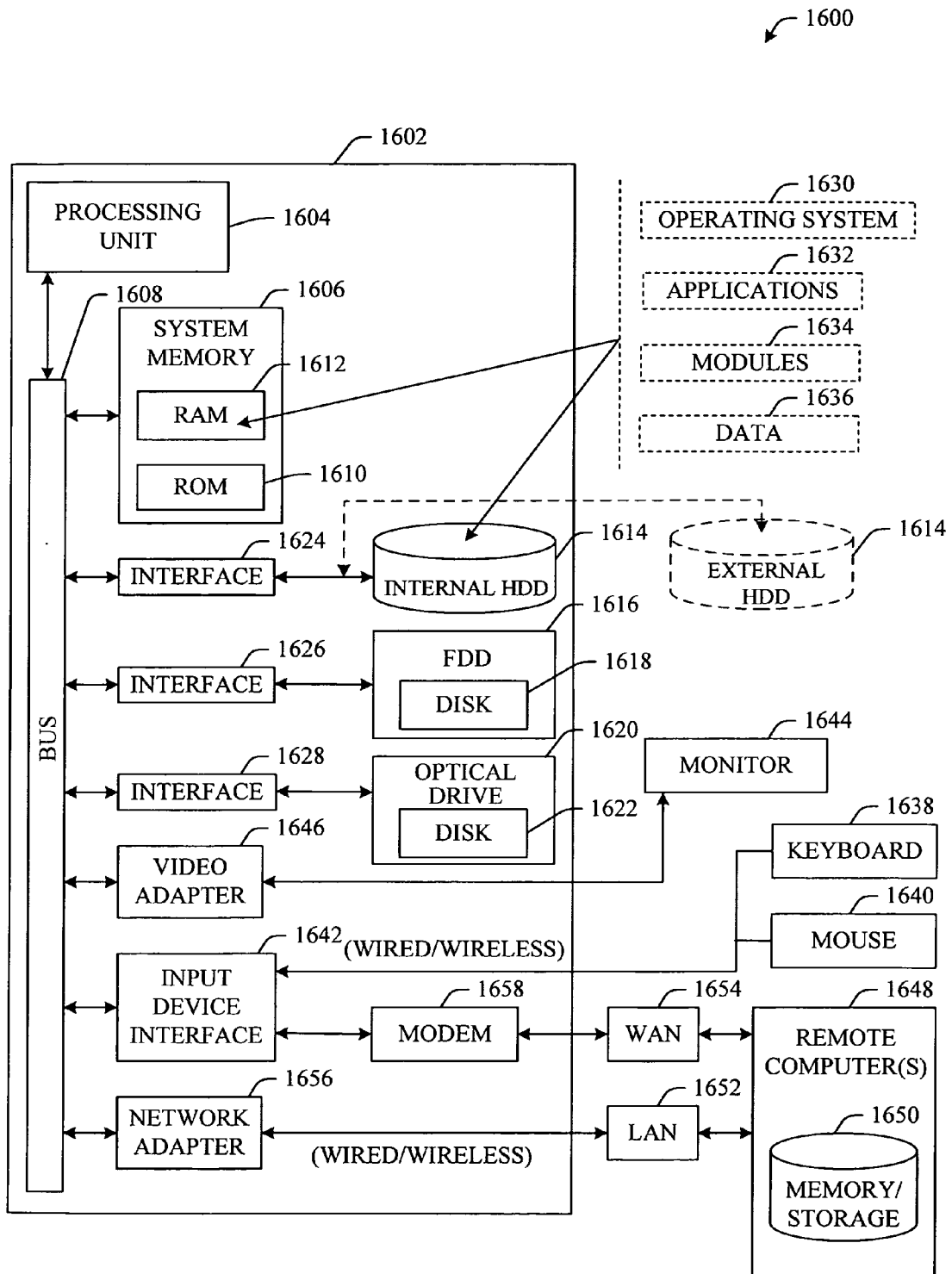


FIG. 16

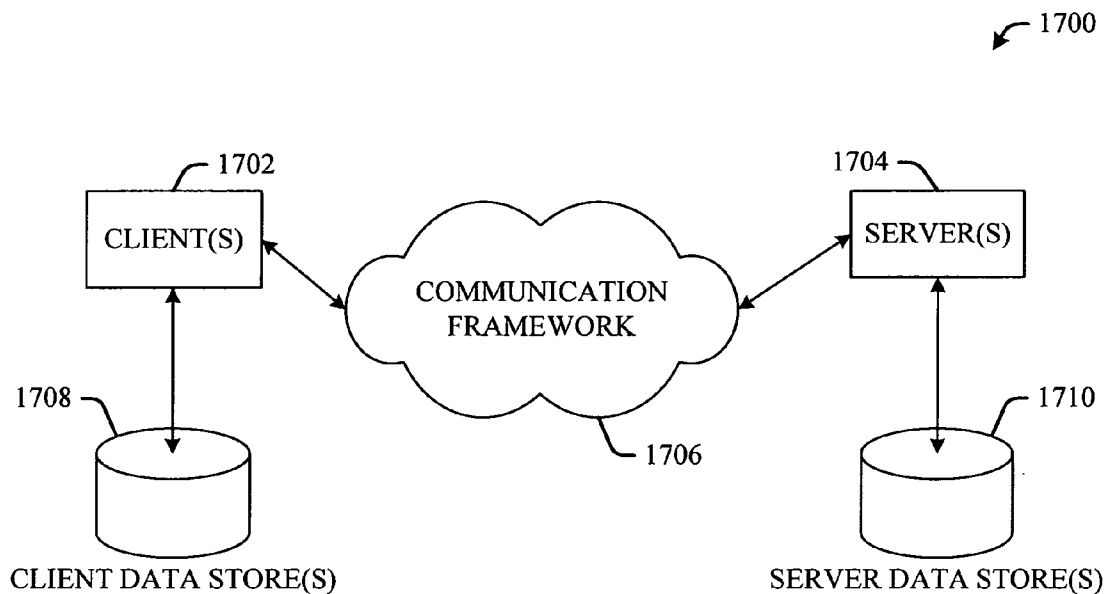


FIG. 17

FANTASY SPORTS LIVE

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Patent application Ser. No. 60/711,259 entitled “FANTASY SPORTS LIVE” and filed Aug. 25, 2005. The entirety of the above-noted application is incorporated by reference herein.

BACKGROUND

[0002] Fantasy sports have increased in popularity worldwide. A fantasy sport is a game where participants (e.g., “fantasy owners”) build a team that competes against other fantasy owners based upon statistics generated by individual players or teams of an organized sport. Most often, a fantasy sport converts statistical performance of members (e.g., players) of a fantasy team into points generated by the fantasy team. Generally, this conversion can be manually calculated. Other more complex variations employ a computer model of actual games based upon statistical input generated by the sport.

[0003] Fantasy sports can be modeled after any organized sport. Common examples can include sports such as baseball, football and hockey. Other not-so-common examples of fantasy sports have been known to include basketball, soccer, golf, auto racing and even cricket. Essentially, any sport that includes quantifiable statistics can be modeled in the form of a fantasy sport.

[0004] Probably the most common of the fantasy sports is fantasy football. Fantasy football is a game in which the participants can act as team owners by drafting and trading players. Most often, the players are representative of real life National Football League (NFL) players however, the players can also be representative of players in other real life leagues or organizations (e.g., Canadian Football League (CFL), National Collegiate Athletic Association (NCAA)). When playing the game, points are scored based upon real life statistics representative of the players’ performance in games.

[0005] A fantasy football league can be modeled after the real life sport whereas teams play each other in a “head-to-head” format deeming the winner the team with the most points. As well, team records are most often kept and a champion is determined based upon the season records. Oftentimes, the league holds playoffs or bowl games to determine a champion. Other variations exist where a champion is determined based upon a total number of points per team instead of team records. In any case, all variations are based upon one common factor, tying points per play to an actual “real life” performance on the field.

[0006] The ever-increasing popularity of the Internet has greatly affected fantasy sports. For instance, fantasy football has evolved from a recreational activity into a multi-million dollar industry partially due to the rise of the Internet. The Internet enables participants world-wide to compete against each other in the same fantasy sports league. Today, fantasy football, which utilizes statistics from the NFL, is probably the most popular of the fantasy sports.

[0007] As described above, in calculating points in fantasy football, participants usually earn point commensurate with

performance in their weekly games. For example, a common scoring format can be 1 point for 20 passing yards, 1 point for 10 rushing yards, 1 point for 10 receiving yards, 8 points for a touchdown, -4 points for a turnover, 1 point for a reception, and so on.

[0008] Although computers, and more specifically the Internet, are frequently used to enable fantasy sports leagues, conventional systems oftentimes calculate points at a central location in a retrospective manner. Alternatively, conventional live scoring mechanisms required browser clients to download applications or to use plugins. Due to their plugin nature, conventional systems are subject to instability caused by inter-process communication. Furthermore, used in this particular manner, conventional systems often require persistent connections due to the push model employed to update their data.

SUMMARY

[0009] The following presents a simplified summary of the invention in order to provide a basic understanding of some aspects of the invention. This summary is not an extensive overview of the invention. It is not intended to identify key/critical elements of the invention or to delineate the scope of the invention. Its sole purpose is to present some concepts of the invention in a simplified form as a prelude to the more detailed description that is presented later.

[0010] The invention disclosed and claimed herein, in one aspect thereof, comprises a live scoring that can be employed without plugins thereby enhancing reliability and scalability. Moreover, the live scoring system utilizes a pull model which does not require persistent connections to a live scoring source, server or network. In another aspect, the system is capable of recovering from lossy connectivity.

[0011] Furthermore, because live scoring systems are resource intensive applications, e.g., require a large amount of processing power to calculate the scores of all players for all teams across all leagues, a client-side scoring/calculation system is employed to further enhance performance of the system. The distributed approach used by the system described herein can particularly offload the burden of processing the updates most of the time to clients, without sacrificing down-level browsers’ ability to view the latest scores. It will be understood that this distributed approach can increase the solution’s scalability.

[0012] In yet another aspect thereof, an artificial intelligence component is provided that employs a probabilistic and/or statistical-based analysis to prognose or infer an action that a user desires to be automatically performed.

[0013] To the accomplishment of the foregoing and related ends, certain illustrative aspects of the invention are described herein in connection with the following description and the annexed drawings. These aspects are indicative, however, of but a few of the various ways in which the principles of the invention can be employed and the subject invention is intended to include all such aspects and their equivalents. Other advantages and novel features of the invention will become apparent from the following detailed description of the invention when considered in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. 1 illustrates an exemplary block diagram of a system that facilitates live scoring of fantasy sports.

[0015] FIG. 2 illustrates a flow diagram of procedures that facilitate requesting and receiving statistical updates in accordance with an aspect of the invention.

[0016] FIG. 3 illustrates an architectural diagram of a system that employs a polling component and a de-queuing component in accordance with an exemplary live scoring system.

[0017] FIG. 4 illustrates a block diagram that illustrates a scalability aspect of the live scoring system having multiple client components and multiple event server components.

[0018] FIG. 5 illustrates an architectural diagram of a system that employs a calculation component and a monitor component in accordance with a client-side scoring system.

[0019] FIG. 6 illustrates an exemplary flow chart of procedures that facilitate polling and receiving updates from a live feed server in accordance with an aspect of the invention.

[0020] FIG. 7 illustrates an exemplary user interface (UI) page that accommodates “fantasy matchups tracker”, “pro matchups tracker” and “my matchup tracker” in accordance with an aspect of the invention.

[0021] FIG. 8 illustrates a UI preview in accordance with an aspect of the fantasy sports system.

[0022] FIG. 9 illustrates an exemplary UI page of a pre-game state view in accordance with an aspect of the invention.

[0023] FIG. 10 illustrates an in-game state view in accordance with an aspect of the fantasy sports system.

[0024] FIG. 11 illustrates a post-game state view in accordance with an aspect of the fantasy sports system.

[0025] FIG. 12 illustrates an exemplary “head-to-head” view of the fantasy sports game.

[0026] FIG. 13 illustrates an exemplary architecture of a live scoring system.

[0027] FIG. 14 is a flow diagram of procedures that facilitate a process of updating the page when receiving live scoring updates in accordance with an aspect of the live scoring system.

[0028] FIG. 15 illustrates an architecture including a heuristic-based component that can automate functionality in accordance with an aspect of the invention.

[0029] FIG. 16 illustrates a block diagram of a computer operable to execute the disclosed architecture.

[0030] FIG. 17 illustrates a schematic block diagram of an exemplary computing environment in accordance with the subject invention.

DETAILED DESCRIPTION

[0031] The invention is now described with reference to the drawings, wherein like reference numerals are used to refer to like elements throughout. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the subject invention. It may be evident, however, that the invention can be practiced without these specific details. In

other instances, well-known structures and devices are shown in block diagram form in order to facilitate describing the invention.

[0032] As used in this application, the terms “component” and “system” are intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component can be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a server and the server can be a component. One or more components can reside within a process and/or thread of execution, and a component can be localized on one computer and/or distributed between two or more computers.

[0033] As used herein, the term to “infer” or “inference” refer generally to the process of reasoning about or inferring states of the system, environment, and/or user from a set of observations as captured via events and/or data. Inference can be employed to identify a specific context or action, or can generate a probability distribution over states, for example. The inference can be probabilistic—that is, the computation of a probability distribution over states of interest based on a consideration of data and events. Inference can also refer to techniques employed for composing higher-level events from a set of events and/or data. Such inference results in the construction of new events or actions from a set of observed events and/or stored event data, whether or not the events are correlated in close temporal proximity, and whether the events and data come from one or several event and data sources.

[0034] While certain ways of displaying information to users are shown and described with respect to certain figures as screenshots, those skilled in the relevant art will recognize that various other alternatives can be employed. The terms “screen,” “web page,” and “page” are generally used interchangeably herein. The pages or screens are stored and/or transmitted as display descriptions, as graphical user interfaces, or by other methods of depicting information on a screen (whether personal computer, PDA, mobile telephone, or other suitable device, for example) where the layout and information or content to be displayed on the page is stored in memory, database, or another storage facility.

[0035] Referring initially to the drawings, FIG. 1 illustrates a system 100 that facilitates live scoring related to fantasy sports. More particularly, the system 100 can mitigate the effects of flash crowds, as live scoring, by definition, results in flash crowds. It will be appreciated that flash crowds are most often encountered when many users request the same type of information (e.g., their respective fantasy scores) at the same time (e.g., during a live sports game). In furtherance of reducing the implications of flash crowds and facilitating live scoring, the system 100 generally includes an event server 102 that can be employed as an interface between a live feed server 104 and a client component 106.

[0036] As can be better understood upon a discussion of the figures that follow, the subject live scoring mechanisms do not require plugins in order to effect scoring techniques. As such, the subject mechanisms (e.g., system 100) can generally increase the system stability since there is no need to have communication between the browser process and the plugin process. Furthermore, the system 100 can be

employed by browsers that are not sufficiently modern or do not have the plugins (or meet the requirements to run the plugins). Indeed, the down-level experience can still provide browsers with scoring data they are capable of interpreting so long as they support hypertext markup language (HTML).

[0037] In accordance with system 100, the elimination of the requirement for persistent connections can free up the servers (e.g., event server 102) to handle more distinct connections for the same time period. Therefore the servers (e.g., event server 102) can service more requests from clients (e.g., client component 106) which generally increase scalability. The architecture of system 100 illustrated in FIG. 1 enables recovery from lossy connectivity which in turn allows browsers to cope elegantly with network congestion rather than crashing the application due to loss of persistent connectivity.

[0038] Furthermore, this system architecture 100 illustrated in FIG. 1 also results in convergence of the data requests. Over time, all browsers on the system can request the same data, resulting in a natural cache optimization, which can also support enhanced scalability. The distributed approach (e.g., system 100) to processing the updates frees the servers to handle other tasks, such as handling connections, again, further increasing scalability.

[0039] Continuing with the discussion of the system 100, in operation, the event server 102 can retrieve live scoring update information from the live feed server 104. It will be understood and appreciated that this retrieval can be prompted via a monitoring mechanism—each of which will be described in connection with the figures that follow. Once updated, one or more clients (e.g., client component 106) can send a request to the event server 102 to update (or initiate) information contained within the client 106.

[0040] Effectively, an indexed de-queuing mechanism can be employed by the event server 102 thereby returning only a subset of the information to the client 106. In other words, in one aspect, the client 106 would send a request to the event server 102 for information. The request would include the last known identification of an event. In response, the client 106, in accordance with a novel de-queuing process of the event server 102, would only receive information that represents a subset of the information present on the event server 102 but, not present on the client 106. Although this aspect employs information that identifies the last known identification sent in the request, other tracking and indexing methodologies can be employed in accordance with alternative aspects. These alternative aspects are to be considered a part of this disclosure and claims appended hereto.

[0041] In one aspect, the live feed server 104 employs cache objects to store a complete list of event data. As will be understood, oftentimes, a cache is employed that can increase speed and efficiency of data transfer. It will be appreciated that other methods of storing and/or maintaining the data by the live feed server 104 can be employed without departing from the spirit and scope of the mechanism(s) described herein.

[0042] FIG. 2 illustrates a methodology of obtaining real-time data in accordance with an aspect of the innovation. While, for purposes of simplicity of explanation, the one or more methodologies shown herein, e.g., in the form of a flow

chart, are shown and described as a series of acts, it is to be understood and appreciated that the subject invention is not limited by the order of acts, as some acts may, in accordance with the invention, occur in a different order and/or concurrently with other acts from that shown and described herein. For example, those skilled in the art will understand and appreciate that a methodology could alternatively be represented as a series of interrelated states or events, such as in a state diagram. Moreover, not all illustrated acts may be required to implement a methodology in accordance with the invention.

[0043] As can be understood from FIG. 2, a repeating sequence of events that can be employed is shown. At 202, a request is received for latest updates. This request can be in accordance with any fantasy sport and data (e.g., statistics) related thereto. This request can be manually triggered by a user and/or application. As well, a rule can be employed to pre-program (or artificial intelligence to infer) a retrieval interval by which information is requested from a server. In still another aspect, a monitoring component can be employed whereas a request is sent when new/updated data is present.

[0044] At 204, a determination is made if updates exist. In other words, a requestor (e.g., client) can request updates related to a fantasy sport. Accordingly, the system can analyze the request with respect to data currently available on the client. If new data (e.g., updates) does not exist, a stop block is reached. If a determination is made at 204 that updates are available, the system proceeds.

[0045] At 206, the updates are received. As will be described in greater detail infra, in one aspect, the updates can be pulled from an event server as illustrated in FIG. 1. Once received, the updates are processed at 208. By processing the updates, conversion and scoring can be effected with respect to statistical data received. One novel feature of the subject scoring mechanism is that processing/scoring can be effected at the client-side of the system. As will be further understood upon a discussion of the figures that follow, an event server and/or application can poll a live feed server on a regular interval or as desired by a user.

[0046] Referring now to FIG. 3, an alternative architectural diagram of system 100 is shown. As shown, system 100 can include 1 to N event server components, where N is an integer. It is to be understood and appreciated that event servers 1 to N can be referred to collectively and/or individually as event server 102. Additionally, each event server 102 can include a polling component 302 and a de-queuing component 304.

[0047] The live feed server 104 can maintain a single master copy of the complete listing of events in a cache object. Accordingly, the event servers 102 can all request the same data from the live feed server 104. Thus, the live feed server 104 can satisfy all requests using a single cache entry. This cache entry can be updated using an external process that normalizes data from another data provider.

[0048] As shown, the event servers 102 can employ a polling component 302 to poll the live feed server 104 at a regular interval, thereafter replacing internal versions of the data maintained within the event server 102. Essentially, this polling process can effect a cache replication. It will be appreciated that by employing the polling component 302,

together with the de-queuing component **304**, the event server(s) **102** can bring the cached data closer to the end consumer (e.g., player), the client component **106**. This architecture and methodology associated therewith can effectively reduce the latency in satisfying requests made by the client.

[**0049**] With reference now to FIG. 4, an alternative aspect of system **100** is shown. More particularly, client component **106** illustrated in FIG. 4 can include 1 to P clients, where P is an integer. It is to be appreciated that 1 to P client components can be referred to collectively or individually as client components **402**. Moreover, although FIG. 4 illustrates client(s) **402** in communication with one event server **102**, it is to be understood that each client **402**, or group of clients **402** can be connected to one or more event servers **102** at any one time.

[**0050**] As discussed supra, the architecture of system **100** can be particularly useful in preventing flash crowds. It will be appreciated that flash crowd phenomenon can occur on the Internet when a web site catches the attention of a large number of people. Under these conditions the web site will be subject to an unexpected and overloading surge of traffic.

[**0051**] For example, suppose all clients **402** were to request the entire list of updates at a regular interval, the amount of data exchanged could quickly become unwieldy for flash crowd conditions. To address these potential surges of traffic (e.g., flash crowds), network applications (e.g., event servers **102**) are capable of performing indexed de-queuing via the de-queuing component **304**. Indexed de-queuing is a process by which a subset of the total available data is returned. This de-queuing process can be better understood upon a reading of the following scenario. It is to be understood and appreciated that the following scenario is provided solely to add context to the invention and is not intended to be limiting in any way.

[**0052**] Assume the following sequence of events exist on the event server **102**, {A,B,C,D,E,F}. The client **402**, having no information about the current sequence of events, can make an unqualified request: "Retrieve all events to date". In response, the event server **102** can return all of the events to date, {A,B,C,D,E,F}.

[**0053**] Accordingly, now, the client **402** knows the latest event is F. Therefore, the next request the client **402** makes to the event server **102** can be "Retrieve all events since F." If no events have occurred after F, the event server **102** returns an empty set. A few moments later, data on the event server **102** is updated and now contains {A, B, C, D, E, F, G, H}. The client **402** reiterates the request to "Retrieve all events since F." The client **402** now receives the set {G, H} from the event server **102**, due to the process of indexed de-queuing (e.g., de-queuing component **304**).

[**0054**] Essentially, indexed de-queuing can refer to a specific entry within a sequence and retrieve all entries that follow the specific entry. In one aspect, the sequence can be chronologically ordered, but is not ordered using timestamps.

[**0055**] Referring again to the chronologically ordered event sequence, one reason for ordering scheme is that clients are able to retrieve data based on the last item they received (e.g., last known identification (ID)). Using a request for the next record in a sequence, e.g., the Hypertext

Transfer Protocol (HTTP) GET protocol request, the system can create a cacheable entry valid so long as the underlying data does not change.

[**0056**] Thus, if the underlying data is {A,B,C,D,E,F,G,H}, and a client **402** requests to "Retrieve all events since F", the response {G, H} can be served to all clients **402** making the same request so long as the underlying data does not change. In accordance with the chronologically ordered events, it is to be understood that the mechanisms described herein can alleviate clock synchronization requirements related to timestamp based retrieval. Moreover, these mechanisms can also alleviate a requirement of having multiple timestamp qualified requests map to the same cache entry. Rather, a single cache entry can be used per qualified request thereby alleviating the effects of potential flash crowds.

[**0057**] Continuing with the example and with reference to FIG. 5, since all clients **402** will be requesting the same data from event server(s) **102**, only specifying a different last known ID, there can be several cache entries created initially. However, after receiving the response from the event server(s) **102**, the clients **402** have an extremely high probability of having all the same data once the response is processed. Therefore, the first client **402** to make a request to the event server **102** will trigger the creation of a cache entry on the event server **102**. An entry creation component **502** can be employed by the event server with respect to the request thus, all subsequent clients **402** will be served the same data so long as the underlying data is not changed.

[**0058**] Once the underlying data is changed, the first client **402** to make a request will trigger the creation (e.g., via entry creation component **502**) of a new cache entry that can again satisfy multiple client requests. Thus, the system **100** can quickly and naturally evolve into a state where all requests issued by clients **402** in the system **100** converge to a very small set of cache entries on the event server **102**. It is to be appreciated that this can facilitate scalability of the system **100**. In other words, it is to be understood that the more clients **402** there are, the faster the cache entry is created, and the more clients **402** are served from the cache entry.

[**0059**] Alternatively, a monitor component **504** can be employed to monitor the event server **102** in relation to one or more clients **402**. Although the monitor component **504** is illustrated inclusive of each of the clients **402**, it is to be understood and appreciated that a monitor component **504** can be included within (or stand alone) the event server **102** thereby prompting the of availability of updates. Accordingly, each client **402** could monitor the event server **102** and retrieve updates accordingly via the novel pull model described herein.

[**0060**] As illustrated in FIG. 5, each client component **402** can include a calculation component **506** that facilitates analyzing the data received from the event server **102** thereafter generating an output. For example, the calculation component **506** can be employed to generate fantasy points based at least in part upon the data received or obtained from the event server **102**. Effectively, the calculation component **506** facilitates client-side scoring of the updates from the event server **102**. Effectively, scoring is processed on each individual client **506** rather than on a central server (e.g., event server **102**).

[**0061**] In order to perform client-side scoring in a way that is faithful to what the results would have been had the

process run on the server (e.g., event server **102**), a number of criteria can be met by the subject live scoring system **100**. First, the system **100** provides for a mechanism by which initial values for scores and statistics are available and returned upon construction of the page by the server. As well, updates to the statistics are available through requests made by the clients **402** to the event server **102**. A mechanism by which to score the updates compatible with popular modem browsers can be accomplished by leveraging scripting applications and client-side extensible stylesheet language (XSL) transforms (e.g., via the calculation component **506**). Furthermore, the subject mechanism(s) employ a set of rules used to calculate scores from the update that are available by creating a client-side initiated XML HTTP Request for the data. Although specific client-side calculation mechanisms are described herein with respect to information pulled from the event server(s), it is to be understood and appreciated that the novel client-side processing functionality described herein can include additional calculation mechanisms without departing from the spirit and/or scope of the innovation and claims appended hereto.

[**0062**] As described supra, a listener/notifier event model (e.g., monitor component **504**) can be used on the client **402** to allow loose coupling of the scoring components (e.g., calculation component **506**) and the user interface (UI) elements. In accordance with the monitor component **504**, each element on the page can register for scoring update events and can contain data that allows it to determine if the event is relevant to the element.

[**0063**] For example, an element that tracks a player score would look for events fired about that player. Similarly, an element tracking a team score would look for events fired about any of the players that are part of the team. Furthermore, a single element can be responsible for creating the requests to the event server **102** thereafter firing any events based on the data it receives and processes. As described supra, the element can contain data indicating the last known ID used to create the page in the UI.

[**0064**] It is to be appreciated that this ID can be the bridge between the requests from the client **402** for scoring and the server side (e.g., event server **102**) processing of scores. The ID can also be used to begin the process of retrieving updates from the event server **102**, and is updated as per the indexed de-queuing process.

[**0065**] Updates can contain the difference between a statistic current cumulative value and a previous cumulative value (the difference being referred to as the “delta”), a unique event ID, an identifying code for the statistic and an identifier for the player or team earning that statistic. The delta, current cumulative value and code can be used to calculate the value by which the score for an individual player must be updated. This value can be calculated on the client-side of the system via calculation component **506**. Moreover, this information is bundled into an event, and the event is fired. All elements listening for that event verify if the event is specific to a player they need to track. If so, they update their display accordingly, otherwise they do nothing. This methodology supports scoring techniques including, but not limited to, range-based and linear scoring evaluation techniques as described herein. This process will be better understood upon a review of FIG. **14** infra.

[**0066**] One benefit of using the page data to initialize the client-side system is that if a client **402** is not modem enough

to support the technology used by the solution, the client **402** can still get live scores (those that are initially returned on the page), although at a slower rate than modem clients supporting the technology used by the system.

[**0067**] FIG. **6** illustrates a methodology of updating from a live feed server (e.g., **104**). At **602**, a live feed server can be polled for updates. A determination is made at **604** if updates exist. If not, a stop block is reached.

[**0068**] If a determination is made at **604** that updates do exist, the updates are received at **606**. At **608**, previous versions of the data are replaced with the updates. Thus, the clients can request updates thereby receiving the most current data available at any given time.

[**0069**] Turning now to a discussion of a specific example that employs the live scoring mechanisms of the aforementioned system, the example described below provides detailed coverage of a game and its events, mixing editorial content as well as statistics and fantasy points. As will be described, views of a game can be available before, during and after a game with content changing accordingly.

[**0070**] In accordance with the exemplary aspect, there are a few elements that can be found on many of the UI pages. Most often, these pages display the same information. It will be appreciated that all of the recurring elements can be powered by the live scoring system described supra. FIG. **7** illustrates an exemplary UI page that accommodates “fantasy matchups tracker”, “pro matchups tracker” and “my matchup tracker.”

[**0071**] Referring initially to the “fantasy matchups tracker” portion **702** of the exemplary UI page **700**, the fantasy matchups tracker lists the 2 fantasy teams playing against each other, their respective score, as well as an indicator determining whether or not all pro matchups **704** required to finalize the fantasy matchups have been completed. Clicking on a matchup will bring the user to see that matchup in its appropriate state. The pro matchups tracker **704** lists the 2 teams playing against each other, their respective score, as well as the time remaining at the last time the score was sampled. Clicking on a matchup will bring the user to see that matchup in its appropriate state.

[**0072**] As illustrated in FIG. **7** each fantasy matchup for the league can include opponent names, opponent scores, and matchup status (e.g., Not Started, In-Progress, Final). Turning now to the “my matchup tracker” section **706** of the exemplary UI, specific information concerning an individual player’s matchups can be highlighted.

[**0073**] Continuing with the example of FIG. **7**, the “my matchup tracker” section **706** lists the 2 fantasy teams playing against each other, their respective score, as well as an indicator determining whether or not all pro matchups required to finalize the fantasy matchups have been completed.

[**0074**] Many of the elements on the UI pages leverage live scoring functionality in that the page need not be completely refreshed to provide updated information. To this end, certain elements of the page can be updated through script at a higher frequency than would be possible if the entire page needed to be refreshed. In one example, the UI are state-sensitive. Additionally, there can be different pages for matchups that have yet to start, for those that are under way

and for those that have ended. Each of these scenarios is described in greater detail below with reference to these states as well as supporting exemplary UI pages.

[0075] FIG. 8 illustrates a UI preview 800 in accordance with an aspect of the fantasy sports system. As shown, the preview 800 is mainly editorial, however, it does add information about the current seasonal standings, who will be playing and which players may or may not play as per the illustration. The preview supporting page 800 can be made available all week. It will be appreciated that the editorial text can be provided from a sports news partner. Similarly, the standings can be provided by a standings/statistical partner. The lineup and the injury reporting information can be provided using a team roster and player service partner respectively. Finally, the recurring elements can be provided by the live scoring system described herein.

[0076] Turning now to FIG. 9, an exemplary UI page of a pre-game state view 900 is shown. More particularly, the pre-game state view illustrates is team roster, player and schedule intensive, as per the illustration.

[0077] FIG. 10 illustrates an in-game state view 1000 in accordance with an aspect of the fantasy sports system. Essentially, the in-game state view 1000 is roster, player and schedule intensive and provides a listing of the last scoring plays made as per the illustration.

[0078] The post-game state view 1100 is illustrated in FIG. 11. This state view 1100 is roster, player and schedule intensive and provides a listing of the last plays made as per the illustration. It is to be appreciated that scoring data can be provided by the live scoring service, described supra, until the next data.

[0079] FIG. 12 illustrates an exemplary "head-to-head" view 1200 of the fantasy sports game. The head to head view 1200 is roster and player intensive, and can provide editorial content for each player, as well as exposing which is the preferred player.

[0080] As described above, the subject fantasy sports system can facilitate client-side live scoring. In order to satisfy traffic spikes or flash crowds (e.g., over 1 million users potentially requesting data every 10 seconds) during game times, a simple, scalable method of scoring players for any league can be employed. One novel feature of the innovation is to serve the same optimized data to all clients and have clients do the processing.

[0081] This novel approach has several benefits, for both customers and providers alike. First, the approach is a particularly high scalable approach that can require less hardware and less bandwidth than conventional systems. Additionally, cacheable data can allow leveraging event servers to perform content delivery thereby increasing responsiveness for users. Moreover, exposing live data in such an open manner allows for development of extremely interesting applications.

[0082] As described supra, fantasy sports scoring can refer to the act of turning one number into another using a mapping function. Live scoring can be simply described as turning one number into another number instantaneously and/or "on the fly". Using delta-based updates, the subject innovation can employ live scoring on a single feed to keep a page's players' scores updated at all times.

[0083] As described herein, the client-side live scoring implementation includes a scoring system that contains mappings of scoring methods (and their codes) to points. Additionally, a live statistics feed is included having statistics marked with scoring method codes. The innovation also includes web page elements that register for scoring updates and dispatchers that fire events for a given player score and statistic updates and team score and statistic updates.

[0084] It is to be understood that one novel feature of the subject live scoring system is to serve the same data to all clients. As such, using edge networks, or event servers (e.g., 102 of FIG. 13), the subject innovation can provide users with quick response time as well as free up server resources for more useful tasks rather than handling an overwhelming amount of requests for the same data. It is to be appreciated that an edge network 102 can refer to a network located on the periphery of a centralized network. In operation, the edge network 102 can retrieve event-related information from the central, or core, network.

[0085] The exemplary architecture of FIG. 13 can be employed to keep the data on the edge network (e.g., event server(s) 102) fresh and up-to-date. As illustrated, the repeating sequence of events that takes place can be quite simple. First, user 106 can request latest events (e.g., every 10 second interval) from the edge network 102. Next, the user 106 can receive the latest events. At regular intervals (e.g., every 5 to 10 seconds), the edge network 106 can poll the live feed server 104 for an entire list of events. Every successful fetch from an edge network 102 replaces the entire list of events on the particular event server 102.

[0086] Turning now to a discussion of the client-side script strategy, using the client-side Dynamic-HTML (DHTML) library, the subject innovation can devise an event driven page. The diagram illustrated in FIG. 14 describes a process of updating the event driven page when receiving live scoring updates.

[0087] At 1402, a statistic update is received. A determination is made at 1404 if new statistics exist in the updates. If not, a request can be sent after a predetermined period (e.g., 10 seconds) to retrieve new updates. On the other hand, if at 1404, a determination is made that new statistics exist, the client can proceed to score the statistics at 1408. For example, in one aspect, XSL transforms can be employed to score statistics. Again, it is important to note that client-side scoring can be accomplished via other means in addition to the XSL transforms described herein. These additional mechanisms and mapping functions are to be included within the scope of this disclosure and claims appended hereto.

[0088] Once calculated, the scored statistics can be employed to roll up player statistics 1410, roll up player scores 1412 and roll up team scores 1414. A dispatcher can be notified and the player statistic updates, player score updates and team score updates can be dispatched at 1416, 1418 and 1420 respectively.

[0089] It is to be understood that the client-side scoring can consume an extremely optimized data schema. For example, in one aspect, the schema can satisfy the requirements with respect to statistics, time and updates. More particularly, the schema would be employed to return statistics for a given player or team acting as a player (e.g.,

defense, special teams). Additionally, the schema can be employed to facilitate returning the time left in all currently live games (e.g., professional games). Finally, the schema can be employed to return the scoring updates for all currently live games.

[0090] The optimization described herein means that the data consumed may not be understandable by casual observers. To this end, a translation is in order. The table below summarizes the information, in accordance with one particular aspect, that can be returned in the live feed together with its corresponding meaning.

Optimized Name	Original Item Name	Parent	Sample Value
A	Events	—	
EL	LiveStatEvent	Events	
I	EventId	LiveStatEvent	632522048489307799
C	GameCode	LiveStatEvent	20040911015
N	ScorePeriodNbr	LiveStatEvent	0
S	ScoreTime	LiveStatEvent	0
T	ProTeamId	LiveStatEvent	10
P	ProPlayerId	LiveStatEvent	3114
M	ScoringMethod-Code	LiveStatEvent	PaAtt
D	DeltaValue	LiveStatEvent	14.00
V	CumValue	LiveStatEvent	14.00
GL	LiveGameEvents	Events	
I	EventId	LiveGameEvents	632522048489307800
C	GameCode	LiveGameEvents	20040911015
N	ScorePeriodNbr	LiveGameEvents	0
S	ScoreTime	LiveGameEvents	0
T	ProTeamId	LiveGameEvents	10
D	DeltaValue	LiveGameEvents	1
V	CumValue	LiveGameEvents	7

[0091] It is to be appreciated from the above table the manner in which both live statistics (e.g., which fantasy player did what) and live game events are returned (e.g., how much a pro team earned from a scoring play). Continuing with the example of fantasy football, live statistic events can be tied to fantasy players (which include pro players, special teams, defensive teams, etc.) and can be scored by running the statistics through the client side scoring engine (e.g., calculation component 506 of FIG. 5). Before being output, the live statistics events will have been filtered to list only the statistics a player may earn.

[0092] Live game events can be tied to pro teams, and they are not scored. They can be used to signal the other events such as the time has passed in the game. Other live game events can signal when a team has scored. This event can identify and return the scoring team and the new points, as well as the total points for the game for that team. Moreover, live game events can signal when a game has started and when it has ended.

[0093] The client-side calculation can employ any desired methods including, but not limited to delta and cumulative sum methodologies. In order to use delta-based update successfully on range-based scoring methods, both the amount by which the statistic changed (i.e., delta) and its current total, after the change (i.e., cumulative sum), are returned. This model can also account for corrections.

[0094] Since the scoring data in the example is returned as XML, an XSL transform can be used to produce the values

by which fantasy points must be updated. This XSL transform should meet the following requirements: league scoring system independent, use scoring method codes to match scoring methods to statistics, edge network cacheable, seasonal client-side cache duration, and produce same results as server-side algorithms.

[0095] FIG. 15 illustrates a system 1500 that employs a heuristic component 1502 which facilitates automating one or more features in accordance with the subject invention. The subject invention (e.g., in connection with retrieving statistics) can employ various heuristic and/or artificial intelligence (AI)-based schemes for carrying out various aspects thereof and for adapting behavior in accordance with operating characteristics. For example, a process for determining a rate of transmission, when/how to adjust data retrieval frequency or which statistics/updates to retrieve can be facilitated via heuristics and/or AI-based systems and processes. By way of further example, in one aspect, the system can determine if problem exists with respect to retrieved data, in which case, the rate of data transmission can be adjusted in order to address any current or inferred problem(s). In one example, problems can be determined in relation to a number of errors received. Accordingly, the system can evaluate and/or adjust when next to retrieve data (e.g., frequency).

[0096] With respect to AI, a classifier is a function that maps an input attribute vector, $x=(x_1, x_2, x_3, x_4, \dots, x_n)$, to a confidence that the input belongs to a class, that is, $f(x)=confidence(class)$. Such classification can employ a probabilistic and/or statistical-based analysis (e.g., factoring into the analysis utilities and costs) to prognose or infer an action that a user desires to be automatically performed. In the case of scoring/statistical systems, for example, attributes can be words or phrases or other data-specific attributes derived from the words or phrases (e.g., presence of key terms), and the classes can be categories or areas of interest.

[0097] A support vector machine (SVM) is an example of a classifier that can be employed. The SVM operates by finding a hypersurface in the space of possible inputs, which the hypersurface attempts to split the triggering criteria from the non-triggering events. Intuitively, this makes the classification correct for testing data that is near, but not identical to training data. Other directed and undirected model classification approaches include, e.g., naïve Bayes, Bayesian networks, decision trees, neural networks, fuzzy logic models, and probabilistic classification models providing different patterns of independence can be employed. Classification as used herein also is inclusive of statistical regression that is utilized to develop models of priority.

[0098] As will be readily appreciated from the subject specification, the subject invention can employ classifiers that are explicitly trained (e.g., via a generic training data) as well as implicitly trained (e.g., via observing user behavior, receiving extrinsic information). For example, SVM's are configured via a learning or training phase within a classifier constructor and feature selection module. Thus, the classifier(s) can be used to automatically learn and perform a number of functions, including but not limited to determining according to a predetermined criteria when to prompt for updates, which updates to retrieve, etc.

[0099] Referring now to FIG. 16, there is illustrated a block diagram of a computer operable to execute the dis-

closed architecture for live fantasy sports scoring. In order to provide additional context for various aspects of the subject invention, FIG. 16 and the following discussion are intended to provide a brief, general description of a suitable computing environment 1600 in which the various aspects of the invention can be implemented. While the invention has been described above in the general context of computer-executable instructions that may run on one or more computers, those skilled in the art will recognize that the invention also can be implemented in combination with other program modules and/or as a combination of hardware and software.

[0100] Generally, program modules include routines, programs, components, data structures, etc., that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the inventive methods can be practiced with other computer system configurations, including single-processor or multi-processor computer systems, minicomputers, mainframe computers, as well as personal computers, hand-held computing devices, microprocessor-based or programmable consumer electronics, and the like, each of which can be operatively coupled to one or more associated devices.

[0101] The illustrated aspects of the invention may also be practiced in distributed computing environments where certain tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules can be located in both local and remote memory storage devices.

[0102] A computer typically includes a variety of computer-readable media. Computer-readable media can be any available media that can be accessed by the computer and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer-readable media can comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disk (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computer.

[0103] Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism, and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer-readable media.

[0104] With reference again to FIG. 16, the exemplary environment 1600 for implementing various aspects of the

invention includes a computer 1602, the computer 1602 including a processing unit 1604, a system memory 1606 and a system bus 1608. The system bus 1608 couples system components including, but not limited to, the system memory 1606 to the processing unit 1604. The processing unit 1604 can be any of various commercially available processors. Dual microprocessors and other multi-processor architectures may also be employed as the processing unit 1604.

[0105] The system bus 1608 can be any of several types of bus structure that may further interconnect to a memory bus (with or without a memory controller), a peripheral bus, and a local bus using any of a variety of commercially available bus architectures. The system memory 1606 includes read-only memory (ROM) 1610 and random access memory (RAM) 1612. A basic input/output system (BIOS) is stored in a non-volatile memory 1610 such as ROM, EPROM, EEPROM, which BIOS contains the basic routines that help to transfer information between elements within the computer 1602, such as during start-up. The RAM 1612 can also include a high-speed RAM such as static RAM for caching data.

[0106] The computer 1602 further includes an internal hard disk drive (HDD) 1614 (e.g., EIDE, SATA), which internal hard disk drive 1614 may also be configured for external use in a suitable chassis (not shown), a magnetic floppy disk drive (FDD) 1616, (e.g., to read from or write to a removable diskette 1618) and an optical disk drive 1620, (e.g., reading a CD-ROM disk 1622 or, to read from or write to other high capacity optical media such as the DVD). The hard disk drive 1614, magnetic disk drive 1616 and optical disk drive 1620 can be connected to the system bus 1608 by a hard disk drive interface 1624, a magnetic disk drive interface 1626 and an optical drive interface 1628, respectively. The interface 1624 for external drive implementations includes at least one or both of Universal Serial Bus (USB) and IEEE 1394 interface technologies. Other external drive connection technologies are within contemplation of the subject invention.

[0107] The drives and their associated computer-readable media provide nonvolatile storage of data, data structures, computer-executable instructions, and so forth. For the computer 1602, the drives and media accommodate the storage of any data in a suitable digital format. Although the description of computer-readable media above refers to a HDD, a removable magnetic diskette, and a removable optical media such as a CD or DVD, it should be appreciated by those skilled in the art that other types of media which are readable by a computer, such as zip drives, magnetic cassettes, flash memory cards, cartridges, and the like, may also be used in the exemplary operating environment, and further, that any such media may contain computer-executable instructions for performing the methods of the invention.

[0108] A number of program modules can be stored in the drives and RAM 1612, including an operating system 1630, one or more application programs 1632, other program modules 1634 and program data 1636. All or portions of the operating system, applications, modules, and/or data can also be cached in the RAM 1612. It is appreciated that the invention can be implemented with various commercially available operating systems or combinations of operating systems.

[0109] A user can enter commands and information into the computer **1602** through one or more wired/wireless input devices, e.g., a keyboard **1638** and a pointing device, such as a mouse **1640**. Other input devices (not shown) may include a microphone, an IR remote control, a joystick, a game pad, a stylus pen, touch screen, or the like. These and other input devices are often connected to the processing unit **1604** through an input device interface **1642** that is coupled to the system bus **1608**, but can be connected by other interfaces, such as a parallel port, an IEEE 1394 serial port, a game port, a USB port, an IR interface, etc.

[0110] A monitor **1644** or other type of display device is also connected to the system bus **1608** via an interface, such as a video adapter **1646**. In addition to the monitor **1644**, a computer typically includes other peripheral output devices (not shown), such as speakers, printers, etc.

[0111] The computer **1602** may operate in a networked environment using logical connections via wired and/or wireless communications to one or more remote computers, such as a remote computer(s) **1648**. The remote computer(s) **1648** can be a workstation, a server computer, a router, a personal computer, portable computer, microprocessor-based entertainment appliance, a peer device or other common network node, and typically includes many or all of the elements described relative to the computer **1602**, although, for purposes of brevity, only a memory/storage device **1650** is illustrated. The logical connections depicted include wired/wireless connectivity to a local area network (LAN) **1652** and/or larger networks, e.g., a wide area network (WAN) **1654**. Such LAN and WAN networking environments are commonplace in offices and companies, and facilitate enterprise-wide computer networks, such as intranets, all of which may connect to a global communications network, e.g., the Internet.

[0112] When used in a LAN networking environment, the computer **1602** is connected to the local network **1652** through a wired and/or wireless communication network interface or adapter **1656**. The adapter **1656** may facilitate wired or wireless communication to the LAN **1652**, which may also include a wireless access point disposed thereon for communicating with the wireless adapter **1656**.

[0113] When used in a WAN networking environment, the computer **1602** can include a modem **1658**, or is connected to a communications server on the WAN **1654**, or has other means for establishing communications over the WAN **1654**, such as by way of the Internet. The modem **1658**, which can be internal or external and a wired or wireless device, is connected to the system bus **1608** via the serial port interface **1642**. In a networked environment, program modules depicted relative to the computer **1602**, or portions thereof, can be stored in the remote memory/storage device **1650**. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers can be used.

[0114] The computer **1602** is operable to communicate with any wireless devices or entities operatively disposed in wireless communication, e.g., a printer, scanner, desktop and/or portable computer, portable data assistant, communications satellite, any piece of equipment or location associated with a wirelessly detectable tag (e.g., a kiosk, news stand, restroom), and telephone. This includes at least Wi-Fi and Bluetooth™ wireless technologies. Thus, the commu-

nication can be a predefined structure as with a conventional network or simply an ad hoc communication between at least two devices.

[0115] Wi-Fi, or Wireless Fidelity, allows connection to the Internet from a couch at home, a bed in a hotel room, or a conference room at work, without wires. Wi-Fi is a wireless technology similar to that used in a cell phone that enables such devices, e.g., computers, to send and receive data indoors and out; anywhere within the range of a base station. Wi-Fi networks use radio technologies called IEEE 802.11 (a, b, g, etc.) to provide secure, reliable, fast wireless connectivity. A Wi-Fi network can be used to connect computers to each other, to the Internet, and to wired networks (which use IEEE 802.3 or Ethernet). Wi-Fi networks operate in the unlicensed 2.4 and 5 GHz radio bands, at an 11 Mbps (802.11a) or 54 Mbps (802.11b) data rate, for example, or with products that contain both bands (dual band), so the networks can provide real-world performance similar to the basic 10BaseT wired Ethernet networks used in many offices.

[0116] Referring now to FIG. 17, there is illustrated a schematic block diagram of an exemplary computing environment **1700** in accordance with the subject fantasy sports scoring mechanisms. The system **1700** includes one or more client(s) **1702**. The client(s) **1702** can be hardware and/or software (e.g., threads, processes, computing devices). The client(s) **1702** can house cookie(s) and/or associated contextual information by employing the invention, for example.

[0117] The system **1700** also includes one or more server(s) **1704**. The server(s) **1704** can also be hardware and/or software (e.g., threads, processes, computing devices). The servers **1704** can house threads to perform transformations by employing the invention, for example. One possible communication between a client **1702** and a server **1704** can be in the form of a data packet adapted to be transmitted between two or more computer processes. The data packet may include a cookie and/or associated contextual information, for example. The system **1700** includes a communication framework **1706** (e.g., a global communication network such as the Internet) that can be employed to facilitate communications between the client(s) **1702** and the server(s) **1704**.

[0118] Communications can be facilitated via a wired (including optical fiber) and/or wireless technology. The client(s) **1702** are operatively connected to one or more client data store(s) **1708** that can be employed to store information local to the client(s) **1702** (e.g., cookie(s) and/or associated contextual information). Similarly, the server(s) **1704** are operatively connected to one or more server data store(s) **1710** that can be employed to store information local to the servers **1704**.

[0119] What has been described above includes examples of the invention. It is, of course, not possible to describe every conceivable combination of components or methodologies for purposes of describing the subject invention, but one of ordinary skill in the art may recognize that many further combinations and permutations of the invention are possible. Accordingly, the invention is intended to embrace all such alterations, modifications and variations that fall

within the spirit and scope of the appended claims. Furthermore, to the extent that the term “includes” is used in either the detailed description or the claims, such term is intended to be inclusive in a manner similar to the term “comprising” as “comprising” is interpreted when employed as a transitional word in a claim.

What is claimed is:

1. A computer implemented system that facilitates live scoring of a fantasy sport game, comprising the following computer executable components:

- a live feed server that maintains a plurality of statistics that relate to a fantasy sport game;
- an edge application that polls the live feed server for a subset of the plurality of statistics; and
- a client that pulls a subset of the plurality of statistics from the edge application, the client calculates a fantasy score based at least in part upon the subset of the plurality of statistics.

2. The computer implemented system of claim 1, the edge application includes a plurality of servers.

3. The computer implemented system of claim 2, the client sends an identification that represents a last record of the plurality of statistics received; the edge application returns a subset of the plurality of statistics based at least in part upon the identification.

4. The computer implemented system of claim 3, further comprising a de-queuing component that employs an indexed de-queuing mechanism that identifies the subset of the plurality of statistics based at least in part upon the identification.

5. The computer implemented system of claim 1, the live feed server employs a cache object to store the plurality of statistics.

6. The computer implemented system of claim 1, the edge application includes a polling component that facilitates accessing the live feed server in order to identify the subset of the plurality of statistics.

7. The computer implemented system of claim 1, the client includes a calculation component that facilitates client-side fantasy sport scoring.

8. The computer implemented system of claim 1, the client includes a monitor component that monitors the edge application with respect to a new statistic.

9. The computer implemented system of claim 1, further comprising a heuristics component that infers an action that a user desires to be automatically performed.

10. A computer implemented method comprising the following computer executable acts:

- identifying the most recent update of a sport statistic;
- transmitting a request to an event server; the request identifies a most recent update of the sport statistic;
- de-queuing a subset of a plurality of sport statistics based at least in part upon the most recent update of the sport statistic; and

transmitting the subset of the plurality of sport statistics to a client for processing.

11. The computer implemented method of claim 10, further comprising processing the subset of the plurality of sport statistics to establish a fantasy sports score, the act of processing occurs on the client.

12. The computer implemented method of claim 10, the act of de-queuing further comprises indexing a plurality of sport statistics based at least in part upon a chronological criteria.

13. The computer implemented method of claim 12, further comprising caching the plurality of sport statistics in a cache object.

14. The computer implemented method of claim 11, the act of processing includes calculating the fantasy sports score based at least in part upon a performance of a player.

15. The computer implemented method of claim 10, further comprising monitoring the event server to identify the most recent update of the sport statistic.

16. The computer implemented method of claim 10, further comprising inferring a frequency with respect to the act of transmitting the subset of the plurality of sport statistics to the client for processing.

17. A system that facilitates client-side fantasy sport scoring, comprising:

- means for maintaining data that corresponds to a performance of a player in an event;
- means for pulling a subset of the data based at least in part upon the player; and
- means for client-side processing of the subset of the data to establish a fantasy sport score.

18. The system of claim 17, further comprising means for identifying a most current record with respect to the data; the means for pulling the subset of the data is based at least in part upon the most current record.

19. The system of claim 18, further comprising means for caching the subset of the data.

20. The system of claim 19, further comprising means for heuristically determining a frequency of pulling the subset of the data.

* * * * *