Lessons Learned

Scaling a Social Network

michael.glaesemann@myyearbook.com
senior data architect

O’Reilly MySQL Conference & Expo 2011
2011–04–13
Who?
日本
PostgreSQL
community
Why?
myYearbook.com
casual social network
founded in 2006
Google Analytics

<table>
<thead>
<tr>
<th>by gender</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>52%</td>
<td>female</td>
</tr>
<tr>
<td>48%</td>
<td>male</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>by age</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>37%</td>
<td>13–17</td>
</tr>
<tr>
<td>27%</td>
<td>18–24</td>
</tr>
<tr>
<td>13%</td>
<td>25–34</td>
</tr>
<tr>
<td>12%</td>
<td>35–44</td>
</tr>
<tr>
<td>11%</td>
<td>45+</td>
</tr>
<tr>
<td>myYearbook</td>
<td>Facebook</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>70% meet people</td>
<td>80% keep in touch with friends</td>
</tr>
<tr>
<td>50% play games</td>
<td>35% meet people</td>
</tr>
<tr>
<td>40% keep in touch with friends</td>
<td>30% share photos</td>
</tr>
<tr>
<td>35% flirt/date</td>
<td>25% play games</td>
</tr>
<tr>
<td>rank</td>
<td>site</td>
</tr>
<tr>
<td>------</td>
<td>---------------------</td>
</tr>
<tr>
<td>1</td>
<td>myYearbook</td>
</tr>
<tr>
<td>2</td>
<td>MEEZ</td>
</tr>
<tr>
<td>3</td>
<td>Zwinky</td>
</tr>
<tr>
<td>4</td>
<td>Hearst Teen Network</td>
</tr>
<tr>
<td>5</td>
<td>Quizilla</td>
</tr>
</tbody>
</table>
### comScore page views
**July 2009**

<table>
<thead>
<tr>
<th>rank</th>
<th>site</th>
<th>views (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>GaiaOnline</td>
<td>1,105</td>
</tr>
<tr>
<td>21</td>
<td>Chase</td>
<td>1,056</td>
</tr>
<tr>
<td>22</td>
<td>ESPN</td>
<td>984</td>
</tr>
<tr>
<td><strong>23</strong></td>
<td>myYearbook</td>
<td><strong>953</strong></td>
</tr>
<tr>
<td>24</td>
<td>Wikipedia</td>
<td>903</td>
</tr>
<tr>
<td>25</td>
<td>Onemanga</td>
<td>840</td>
</tr>
<tr>
<td>26</td>
<td>Mapquest</td>
<td>833</td>
</tr>
<tr>
<td>27</td>
<td>Foxsports</td>
<td>815</td>
</tr>
</tbody>
</table>

### comScore time spent
**July 2009**

<table>
<thead>
<tr>
<th>rank</th>
<th>site</th>
<th>minutes (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Amazon</td>
<td>806</td>
</tr>
<tr>
<td>21</td>
<td>CNN</td>
<td>744</td>
</tr>
<tr>
<td>22</td>
<td>GaiaOnline</td>
<td>709</td>
</tr>
<tr>
<td>23</td>
<td>Bing</td>
<td>707</td>
</tr>
<tr>
<td>24</td>
<td>MSNBC</td>
<td>691</td>
</tr>
<tr>
<td><strong>25</strong></td>
<td>myYearbook</td>
<td><strong>678</strong></td>
</tr>
<tr>
<td>26</td>
<td>Iwin</td>
<td>670</td>
</tr>
<tr>
<td>27</td>
<td>NickJr</td>
<td>665</td>
</tr>
</tbody>
</table>
2007
100M HTTP requests/month
1 PostgreSQL server
1 Apache/PHP server
static content servers
1 outage per night
2008
2.5G page views
30 PostgreSQL servers
120 Apache/PHP servers
8 ActiveMQ message brokers
99.84% uptime
2010

2.5G HTTP requests/month

Database servers
connection poolers, routers

Web servers
Apache/PHP, Tornado/Python, lighthttpd

Message brokers
ActiveMQ, RabbitMQ

Monitoring, R&D, other servers
rapid growth
resources
API
bottleneck
time
Relative data access latencies

1  CPU Register
1–2  L1 cache
6–10  L2 cache
25–50  main memory (RAM)
10,000,000  hard disk
100,000,000  LAN
1,000,000,000  WAN
fsync
memory
more activity?
views/month 2007 100M → 2009 1.5G
more TPS
more servers
more connections
more configuration
more pain!
reduce TPS!

→ memcached

1.2 TB

get 140K/s, set 15K/s
Skype
PL/Proxy
pgBouncer
less configuration
fewer connections

- app-server 1
- app-server 2
- app-server 3
- postgres 1
- PL/Proxy
- postgres 2
fewer connections!

app-server 1 -> postgres 1
app-server 2 -> postgres 1
app-server 3 -> postgres 1
postgres 1 -> postgres 2
postgres 1 -> postgres 3
postgres 1 -> postgres 4
reduce TPS/server
less configuration

app-server 1

app-server 2

app-server 3

pgbouncer 1

pgproxy

internal pgbouncer

postgres 1

postgres 2

postgres 3

postgres 4
SOA
REST
DB API
death
failure is a choice
network
measure
overview
learn
logical
versus
physical
thought
word
deed
Fusion-io

direct IO

buffered IO

application buffer

Fusion-io card

CPU

Postgres buffer

kernel buffer

Fusion-io card

CPU
improve
automate
test
Lessons Learned
Scaling a Social Network

michael.glaesemann@myyearbook.com
Senior Data Architect

O’Reilly MySQL Conference & Expo 2011
2011–04–13
partition
sharding
inter and intra server
roll off old data