A Bird and the Web

crossbreeding experience

Luxembourg, 2011

Sergey Mereutsa,
DQ Team,
Moldova
Welcome and thanks

Thanks to:

- Sita software
- IBPhoenix
- IBSurgeon
- All attended :)
About us

- Small outsourcing company, which is specializing in web-site development, games and some special software for TV and banks.
- We are 6 years on the market, did 150+ websites (50 for government institutions).
- Sites are from 5 to 500 pages (some news portals - over 100 000 pages).
- DB sizes - typical 50-100 MB, some 500-1500 MB, some 2-4 GB.
- Number of records to manipulate - from 10K to 25M.
Common tasks to solve in the web

1. Parallel access to DB - from tens to hundreds connections
2. Multi-user access with different permissions
3. Huge amount of short-life small-data queries (AJAX)
4. Load-balancing - data replication, cross-database queries
5. Dynamic or static: that is the question
Some statistics for one web-portal
no counts counted :)

Static, 25 days, 2 nodes,
partial AJAX trace:
1. 691 115 Visits
2. AVG 27 644 Visits/Day
3. **1 936 339** requests
4. 77 450 requests/Day
5. Peak: 117 841 req/Day

The same data as it was if site was dynamic - 1 page generation per request, near 15-20 selects per page.

---

Static, 25 days, 2 nodes,
full AJAX trace:
1. 703 147 Visits
2. 28 125 Visits/Day
3. **12 411 999** requests
4. 496 480 requests/Day
5. Peak: 707 761 req/Day

The same data as it was if site was dynamic - several data blocks per request, near 15-20 selects per page in total.
Games - is it serious?
some statistics for typical round

- DB file size - 850MB
- Number of active users - from 10 454 to 3 921
- Round length - 51 days
- Number of transactions - 20 713 104
- 406 139 transactions per day
- TIP was written on disk 31 579 112 times
- Most read table - 2 799 267 records
- Most read/write table - 3 458 313 records
- No NoSQL cache ... yet ;-)
- Galaxy name - Firebird :)


RO and RW transactions, lifetime and errors in the web environment

- Over 95% of queries just READ the data *
- Near all queries MUST be short - limited to lifetime of script execution - typical less than 0.5 seconds, 1 second is too much, 5 seconds is unacceptable!
- Users do not care about ANY problems with your DB - do not bore them with messages about lock conflicts!

* but not in our game
**BL in DB - to be or not to be?**

**NO BL in DB:**
- DB is portable to other RDBMS
- You have to implement BL each time you change access layer to DB
- You have to manage permissions to access data by your BL

**BL in DB:**
- DB is NOT portable
- You can easy change access layer from PHP to Delphi/C#/Whatever
- You do not care about data access security - just grant permissions to SP
Static or dynamic - what to choose?

Static site mode:
● +Fast
● +Scalable
● +Portable as archive
● +Practically no DB load
● -Delayed update
● -No user interaction

Dynamic mode:
● Reasonable Fast
● -Not Scalable
● -Not portable as archive
● -High DB load
● +Instant update
● +User interaction

Choose both and go ahead :)
Counters, ooh, counters!
two words about web and counters

- Never use count(1) from table(s)!
- Never use direct updates if you want real data!
- Need general counts? Parse logs and do background update!
- Need real time counters? Use Redis/Memcache/etc
- Avoid counters, if you can! ;)

Web: No SQL or NoSQL?

<table>
<thead>
<tr>
<th>No SQL: pure key-value:</th>
<th>NoSQL: Firebird+key-value:</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Fast, incredibly fast</td>
<td>+ Fast</td>
</tr>
<tr>
<td>+ Build-in replication</td>
<td>+ Build-in replication</td>
</tr>
<tr>
<td>Not relational</td>
<td>+ Relational</td>
</tr>
<tr>
<td>- no BL in DB *</td>
<td>+ BL in DB</td>
</tr>
<tr>
<td>- no ACID</td>
<td>+ ACID</td>
</tr>
</tbody>
</table>

Melt them together - and enjoy the speed.

* only key-value DB
Citius, Altius, Fortius: anatomy of the FireWeb 5.0 engine

- **FireWeb core logic**
  - PSQL
  - Stored Procedures
  - Triggers

- **Redis cache and counters**

- **Page elements display logic (templates) and modules**
Summary

- Firebird SQL can be used in the web :) 
- Use the power of PSQL 
- Optimize queries 
- Use RO transactions 
- Do not compete with key-value - use NoSQL!
QUESTIONS?

Never ask, if it is possible. It is.