http://tech.myemma.com
@emmaemailtech
Environment at Emma

- 1.6 TB, 1 cluster, Version 8.2 (RAID10)
- 1.1 TB, 2 clusters, Version 8.3 (RAID10)
- 8.4, 9.0 Dev
- Putting 9.0 into production (May 2011)
- pgpool, Redis, RabbitMQ, NFS
Other stats

- daily peaks: ~3000 commits per second
- average writes: 4 MBps
- average reads: 8 MBps
- From benchmarks we’ve done, load is pushing the limits of our hardware.
I say all of this with love.
Huge catalogs

- 409,994 tables
- Minor mistake in parent table definitions
- Parent table updates take 30+ minutes
not null default nextval('important_sequence'::text)

vs

not null default nextval('important_sequence'::regclass)
Huge catalogs

- Bloat in the catalog
- User-provoked ALTER TABLE
- VACUUM FULL of catalog takes 2+ hrs
Huge catalogs suck

- 9,019,868 total data points for table stats
- 4,550,770 total data points for index stats
- Stats collection is slow
Disk Management

- $PGDATA:
  - pg_tblspc (TABLESPACES)
  - pg_xlog
  - global/pg_stats
  - wal for warm standby
Problems we worked through with big schemas Postgres

- Bloat
- Backups
- System resource exhaustion
- Minor upgrades
- Major upgrades
- Transaction wraparound
Bloat Causes

- Frequent UPDATE patterns
- Frequent DELETEs without VACUUM
- a terabyte of dead tuples
SELECT schemaname, tablename, cc.reltuples, cc.relpages, bs, 
   CEIL((cc.reltuples*(datahdr+maxfractionsum*(nullhdr+ma-(case when nullhdr%ma=0 THEN ma ELSE nullhdr%ma END)))/(bs-20::float))) AS otta, 
   COALESCE(c2.relname,'?') AS iname, COALESCE(c2.reltuples,0) AS ituples, COALESCE(c2.relpages,0) AS ipages, 
   COALESCE(CEIL((c2.reltuples*(datahdr-12))/(bs-20::float)),0) AS iotta 
FROM ( 
   SELECT schemaname, tablename, hdr, ma, bs, 
      SUM((1-null_frac)*avg_width) AS datawidth, 
      MAX(null_frac) AS maxfractionsum, 
      hdr+ 
      SELECT 1+count(*)/8 AS const 
      FROM pg_stats s2 
      WHERE null_frac<>0 AND s2.schemaname = s.schemaname AND s2.tablename = s.tablename 
   ) AS nullhdr 
FROM pg_stats s, ( 
   SELECT current_setting('block_size')::numeric AS bs, 
      CASE WHEN substr((v,12,3) IN ('8.0','8.1','8.2') THEN 27 ELSE 23 END AS hdr, 
      CASE WHEN v = 'mingw32' THEN 8 ELSE 4 END AS ma 
   FROM (SELECT version() AS v) AS foo 
) AS const 
GROUP BY 1,2,3,4,5 
) AS foo 
  JOIN pg_class cc ON cc.relname = rs.tablename 
  JOIN pg_namespace nn ON cc.relnamespace = nn.oid AND nn.nspname = rs.schemaname AND nn.nspname <> 'information_schema' 
  LEFT JOIN pg_index i ON indrelid = cc.oid 
  LEFT JOIN pg_class c2 ON c2.oid = i.indexrelid 
WHERE tablename = 'addr' 
ORDER BY wastedbytes DESC LIMIT 1;

Use check_postgres.pl
https://github.com/bucardo/check_postgres/
Fixing bloat

- Wrote scripts to clean things up
- VACUUM (for small amounts)
- CLUSTER
- TRUNCATE (data loss!)
- Or most extreme: DROP/CREATE
- And then ran the scripts.
Backups

- `pg_dump` takes longer and longer
<table>
<thead>
<tr>
<th>backup</th>
<th>duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009-11-22</td>
<td>02:44:36.821475</td>
</tr>
<tr>
<td>2009-11-23</td>
<td>02:46:20.003507</td>
</tr>
<tr>
<td>2009-11-24</td>
<td>02:47:06.260705</td>
</tr>
<tr>
<td><strong>2009-12-06</strong></td>
<td><strong>07:13:04.174964</strong></td>
</tr>
<tr>
<td>2009-12-13</td>
<td>05:00:01.082676</td>
</tr>
<tr>
<td>2009-12-20</td>
<td>06:24:49.433043</td>
</tr>
<tr>
<td>2009-12-27</td>
<td>05:35:20.551477</td>
</tr>
<tr>
<td>2010-01-03</td>
<td>07:36:49.651492</td>
</tr>
<tr>
<td>2010-01-10</td>
<td>05:55:02.396163</td>
</tr>
<tr>
<td>2010-01-17</td>
<td>07:32:33.277559</td>
</tr>
<tr>
<td>2010-01-24</td>
<td>06:22:46.522319</td>
</tr>
<tr>
<td>2010-01-31</td>
<td>10:48:13.060888</td>
</tr>
<tr>
<td>2010-02-07</td>
<td>21:21:47.77618</td>
</tr>
<tr>
<td>2010-02-14</td>
<td>14:32:04.638267</td>
</tr>
<tr>
<td>2010-02-21</td>
<td>11:34:42.353244</td>
</tr>
<tr>
<td>2010-02-28</td>
<td>11:13:02.102345</td>
</tr>
</tbody>
</table>
Backups

• `pg_dump` fails
• patching `pg_dump` for `SELECT ... LIMIT`
• Crank down `shared_buffers`
• or...
Install 32-bit Postgres and libraries on a 64-bit system.

Install 64-bit Postgres/libs of the same version.

Copy “hot backup” from 32-bit sys over to 64-bit sys.

Run pg_dump from 64-bit version on 32-bit Postgres.
PSA

- Warm standby is not a backup
- Hot backup instances
- “You don’t have valid backups, you have valid restores.” (thanks @sarahnovotny)
- Necessity is the mother of invention...
Ship WAL from Solaris x86 -> Linux
It did work!
Running out of inodes

• UFS on Solaris
  “The only way to add more inodes to a UFS filesystem is:
  1. destroy the filesystem and create a new filesystem with a higher inode density
  2. enlarge the filesystem - growfs man page”

• Solution 0: Delete files.

• Solution 1: Sharding and bigger FS on Linux

• Solution 2: ext4 (soon!)
Running out of available file descriptors

- Too many open files by the database
- Pooling - pgpool or pgbouncer?
Minor upgrades

- Stop/start database
- CHECKPOINT() before shutdown
Major Version upgrades

- Too much downtime to dump/restore
- Write tools to migrate data
- Trigger-based replication
- pg_upgrade
Transaction wraparound avoidance

- autovacuum triggers are too small
- Watch age(datfrozenxid)
- Increase autovacuum_freeze_max_age
Thanks!

- We’re hiring! - selena@myemma.com
- Emma’s Tech Blog: http://tech.myemma.com
- My blog: http://chesnok.com
- http://twitter.com/selenamarie