

Proprietary to PostgreSQL

Moving your data to Open Source

Josh Williams

Who is this Josh?

- Technical manager at Nexus Technology in Columbus, Ohio.
- Nexus: Small/Medium Business IT Consulting and Outsourcing
- Light user of PostgreSQL for 4-5 years now.
- But...



Don't shoot! I'm on your side now!

Why?

- Software is constantly moving, evolving:
Proprietary Upgrades (New Features): \$\$\$
- Software has a lifespan:
 - Support and Maintenance Agreements = \$\$+
- Closed Source Administration = \$\$
 - \$\$ → Happy Vendor Executives
- Open Source Administration = \$\$
 - \$\$ → Happy Employees, Happy Community
- PostgreSQL = <3

All We Have To Do

- CREATE OR REPLACE RDBMS;
- ERROR: syntax error at or near "ID10T"

Our Example

- Spam Quarantine / Statistics Engine
- Moving from MS SQL Server 2000, ~15 GB
- Target: PostgreSQL 8.3 on Windows 2003
- Three Components
 - Mail Indexing Process, In House Software, C#
 - User Web Console, In House, ASP.NET
 - Mail Server Software, Commercial – Connects to DB to get user's white/black list info
- Reasons to move
 - SQL Server locking, concurrency problems
 - Use unavailable features: Partitioning
 - 2000?! PG 8.3 is 8 years more advanced!

Proprietary to PostgreSQL Procedure Outline

- Application Compatibility
- Compare DB System Features
- Rewrite Embedded Logic
- Migrate Schema
- Transfer Data
- Make the Application Switch

Application Connectivity

- PostgreSQL DB Driver available?
- Does the application support PostgreSQL?
 - Proprietary software
 - Vendor not receptive
- Database abstraction layer may be available
- In Our Example:
 - Web / Index Process, .NET: ODBC (or Npgsql)
 - Mail Server: ODBC only – but hey, it works!

Compare DBMS Capabilities

- Data Types
- Syntax
 - SERIAL vs IDENTITY vs AUTO_INCREMENT ...
 - Functions used and equivalents
 - table1 JOIN table2 USING(id_column)
 - SELECT TOP *n* ... versus SELECT ... LIMIT *n*
- Procedural Languages, Embedded Logic
 - PL/SQL → PL/pgSQL
 - Stored Procedures?
 - Anonymous blocks?!
- Partitioning (Inheritance, Rule System)

Our Example: Data Types

The “Big” Table

- \d message
 - mid character(20)
 - mdate datetime
 - subject varchar(250)
 - recipient varchar(200)
 - fromline varchar(200)
 - msize integer
 - score numeric(4,2)
 - scoreflags varchar(200)
 - status integer
- datetime → timestamp
- Otherwise, table fits
- Issues in other tables:
 - bit → boolean
 - bit → integer?
 - uniqueidentifier → UUID (contrib)

Our Example: Syntax

- Syntax Changes:
 - SUBSTRING(), CHAR_INDEX() → POSITION()
Luckily not used in application
 - User defined-functions:
 - SQL Server requires calls to be user-qualified.
 - If you don't want to touch the app, create a schema.
 - Quoted identifiers: No more [Square Brackets]
- Procedure Changes:
 - We have 10 functions/SP's, maybe 2 non-trivial.
 - Rewrite app to execute SELECT function();

Opportunity for Rewrite

- New feature set has now been identified...
- Can the application benefit from any?
- How much effort to change the application?
- Keep in mind the rewrite may fix all that we've discussed (Connectivity, Syntax, etc...)

Rework Embedded DB Logic

- Stored Procedures – Out!
- Functions – In!
 - Compatibility: Out parameters (8.1)
 - Return multiple cursor pointers (result sets)
- Select appropriate language:
 - PL/pgSQL, PL/Perl, PL/PHP
 - Whatever works for you!

Example: Function Comparison

MS SQL...

```
CREATE FUNCTION [InTimeWindow] (@now AS datetime, @start AS datetime, @end AS datetime)
RETURNS bit AS
BEGIN
    DECLARE @currenttime datetime;
    DECLARE @shortstart datetime;
    DECLARE @shortend datetime;

    SET @currenttime = @now - CAST(FLOOR(CAST(CAST(@now AS datetime) AS float)) AS datetime)
    SET @shortstart = @start - CAST(FLOOR(CAST(CAST(@start AS datetime) AS float)) AS datetime)
    SET @shortend = @end - CAST(FLOOR(CAST(CAST(@end AS datetime) AS float)) AS datetime)

    IF (@currenttime > @shortstart AND @currenttime < @end) RETURN 1

    RETURN 0
END
```

PL/pgSQL...

```
CREATE FUNCTION intimewindow(now time, starttime time, endtime time)
RETURNS boolean AS
$BODY$
BEGIN
    RETURN now BETWEEN starttime AND endtime;
END;
$BODY$ LANGUAGE plpgsql IMMUTABLE;
```

Rework PL Use in Application

- Stored Procedures → SELECT function();
 - ORM: Building a Stored Procedure command ~ Building a prepared statement
- Anonymous Blocks: Same thing
 - Will likely make app code less complicated!

Migrate Schema

- Find a good utility, pull out SQL statements
 - SQLFairy has been recommended
- Old *NIX tools (like sed) are your friend
- Or if your schema is small enough...
 - Tweak for PostgreSQL in a text editor
 - Take advantage of proper data types
- Load in to PG (in a transaction!)
- Load in some sample data, Test, Test, Test!

Techniques to Cut Down On Downtime

- Data on Old System == Data on New System
- Replication
 - 3rd Party (has to speak both languages)
 - Home Grown
- Link old system to new
 - (or vice versa?)
 - dbilink, ODBC connector

Transfer Data

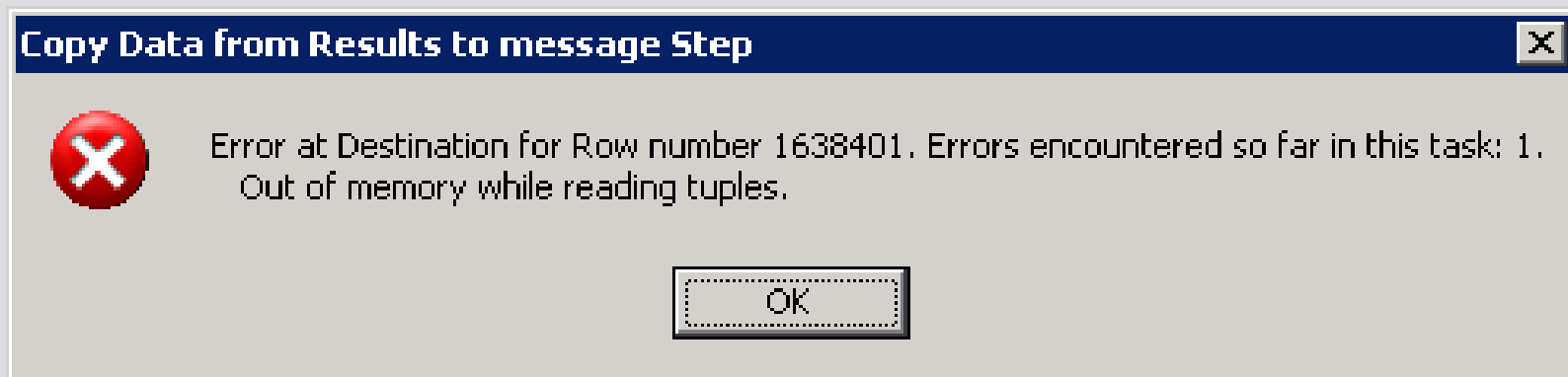
- If Replication/Links work, take your time:
- ETL 3rd Party utility
 - Has to talk to both systems, easier to find
- Replication 3rd utility
 - May be more difficult to find, reliable?
- Home grown utility
- Test, test again!
- If not...

Transfer Data AND Make Application Switch

- Plan & Invoke Downtime
- Place old system in read-only state
- Do the move...
- Test!
- Switch out Application/Connection Settings
- If Something Bad Happens, Don't Panic!
 - Roll application back, old database is still there
 - Figure out what went wrong, adjust your plan

Next Step In Our Example...

- Time for ETL:
- SQL Server → DTS → ODBC → PG
- An hour later:



Attempt #2...

- DTS → Text Files → COPY FROM ...
- The “big” table wrote out an 8 GB text file.
- Took a few tries...
 - Who knew email subjects could have line feeds?
 - Or tabs in SMTP “MAIL FROM” (it's spam, go fig)
- Incidentally: 15G SQL Server ~ 18.5G PG

Proprietary to PostgreSQL Procedure Outline

- Application Compatibility
- Compare DB System Features
- Rewrite Embedded Logic
- Migrate Schema
- Transfer Data
- Make the Application Switch
- Get some sleep

The End

- Discuss!

Proprietary to PostgreSQL

Contact Josh:
joshwilliams@ij.net

Slides up at
<http://www.dbahumor.com/P2P/>