Use of PostgreSQL in Municipal Government
City of Garden Grove, CA
City of Garden Grove

Founded 1874, incorporated 1956.
Population 174,715 (2009)
City of Garden Grove

Direct Services
● Police
● Fire
● Public Works
  ○ Water and Sewer
  ○ Storm Drainage
  ○ Engineering
  ○ Street Maintenance
  ○ Parks and Trees
  ○ Environmental Compliance
● Planning and Permitting
● Redevelopment
● Housing Authority
● Recreation

2010/2011 Budget
○ $87.8 million general
○ $209 million overall
City of Garden Grove

~900 users
450 desktops
70 mobile computers

Linux primary server OS since 1995

Began using PostgreSQL 7.1 in 2001

Information Technology Budget: 2.1 million
City Server OS History

- Use of computing for finance date back to 1960s
- Pick - database / application environment
  - 1977 Microdata Reality : 24 users
  - 1990 Data General Quad Processor Mini
    - $400,000 (1990 dollars)
    - ~ 400 users, dumb terminals
    - DGUX/Advanced Pick
  - 1994 x86/SCO
    - Ethernet and TPC/IP
  - 1995 Linux
How PostgreSQL fits in City database instances in 2000

<table>
<thead>
<tr>
<th>Brand</th>
<th>Servers</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pick</td>
<td>2</td>
<td>Accounting, Budget, Payroll, Business License, Building Permits, Utilities, HR, Fire Permits, Police Records, Housing Authority, Public Works, Code Enforcement, Substandard Housing, Land Use, Street Lighting, Trees, Warehouse Inventory and more.</td>
</tr>
<tr>
<td>Oracle</td>
<td>2</td>
<td>Computer aided dispatch (CAD)</td>
</tr>
</tbody>
</table>
# How PostgreSQL fits in City database instances in 2011

<table>
<thead>
<tr>
<th>Brand</th>
<th>Servers</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pick</td>
<td>1</td>
<td>Accounting, Budget, Payroll, Business License</td>
</tr>
<tr>
<td>Oracle</td>
<td>3</td>
<td>computer aided dispatch (CAD), computer aided drawing (CAD) utility network</td>
</tr>
<tr>
<td>MS-SQL</td>
<td>3</td>
<td>Jail management, property and evidence, public meeting video and minutes</td>
</tr>
<tr>
<td>MySQL</td>
<td>2</td>
<td>OpenAudit, Zimbra</td>
</tr>
<tr>
<td>SQLite</td>
<td>1</td>
<td>Wikis</td>
</tr>
<tr>
<td>FileMaker-Pro</td>
<td>2</td>
<td>Housing authority, In-car video library</td>
</tr>
<tr>
<td>4D</td>
<td>1</td>
<td>Fire inspection, hazardous waste tracking, training</td>
</tr>
</tbody>
</table>
Why PostgreSQL?

City had been using "multi-value" Pick database since 1970s.

In 2001...

MySQL didn't have sub-selects.

DB2 was complicated and didn't have TCL or PHP clients.

Oracle was too laborious.

PostgreSQL

● Free
● Fast
● Easy to install
● Easy to backup
● Worked with many languages
● Good Documentation
● Reliable
Why PostgreSQL?

In 2001 PostgreSQL was missing a few features:

ALTER TABLE .. DROP COLUMN -- added in 7.3, 2002

SCHEMAS -- added in 7.3

ALTER COLUMN TYPE -- added in 8.0, 2005
How PostgreSQL fits in

Commercial Applications - Postgres in the wild

**CanIT-Pro** : Spam filter
**Scalix** : Exchange replacement (being replaced)
**PermitCity** : Building permitting and inspection system
**L3 Communication's in-car video system**
**MapGuide Enterprise** : Web based GIS server
How PostgreSQL fits in

Open Source Applications - Postgres as a choice

**Drupal** : External websites
  - some effort required to select modules to work

**Squidguard** : Web logging
  - used after MySQL index corruption

**RT** : Request Tracker
  - Entity attribute model

**Snort** : Intrusion Detection System
How PostgreSQL fits in

Noteworthy in-house PG applications

**Document Archiving**: Agreements, Resolutions, Ordinances, Minutes, Deeds, Building Plans, etc.
- PG's full text search

**Water**: Water, sewer, billing, accounting, and cross connection systems
- 34,000 + services
- ~ 30 million in billings a year
- 12 years of transaction data online
- Extensive use of Pg functions, triggers,

**Recreation**: Facility and class booking application
- Complex scheduling handled through PG
- Written such that other agencies could use
How PostgreSQL fits in

Noteworthy in-house PG applications

**Public Works Workorders**
- used for all aspects of operations
  - facilities, streets, water distribution, sewer, environmental compliance, signs, trees, graffiti, etc.
- 270,000 workorders
- tracks labor and materials
- Alerts via SMS

**Police Records Management System**
- events, arrests, people, vehicles, cites, TC, FI
- DNA, calls for service, casefiles, etc.

**Municipal Code**
- pagnated and online output
How PostgreSQL fits in

Current DB Setup

PG 9.0.?

dumbo : pglive
  ● IBM 3650, 2 CPU
  ● PG instance is 40GB
  ● Storage is iSCSI / Equallogic
    ○ plain old 1Gb NIC

babar : pgquery
  ● Some virtual machine
  ● Hot slave, streaming replication
  ● We let users run SQL queries here
  ● Good for backups and dumps
  ● GIS Datasource
First uses of PostgreSQL

2001/2002

Homicide case management system
Sewer overflow tracking system
Utility maintenance tracking

- TCL / cgi : bad idea #1
- In-house web application framework : bad idea #2
- Images stored as blobs in PG : bad idea #3
- Paginated reporting using LaTex : bad idea #4
- Separate PG databases for PD and City : bad idea #5
First uses of PostgreSQL

2003

Evidence, FI, Graffiti Photos
- 100 gigs of blobs
  - interesting to backup
  - worked fine
  - no referential integrity
- Converted to BYTEA
  - takes more space
  - still hard to deal with backups
- Moved images out of database in 2005
  - Now 650GB
Ruby on Rails and PostgreSQL

- Started using in 2007
- Easy to start using
- Takes time to master
class Customer < ActiveRecord::Base
  has_many :customer_views
  has_many :accounts
  has_many :echecks
  has_and_belongs_to_many :services
  validates_length_of :state, :is => 2

  def before_save
    unless self.legacy_id.nil?
      if self.legacy_id.empty?
        self.legacy_id = nil
      end
    end
    self.address.upcase!
  end
end

app/models/customer.rb

- if @credit_history
  NB = @credit_history.size.to_s + " credit issues"
  = text_field_tag :deposit, number_with_precision(@deposit, :precision =>2), :size =>
  = label_tag :deposit
  %p
  Check one
  %br
  = f.check_box :owner
  = f.label :owner, "owns the property"
  %br
  = f.check_box :renter
  = f.label :renter, "rents or leases"
  %p
  = label_tag :open_on
  %br
  = calendar_date_select_tag 'open_on', (@off_order ? @off_order.order_on : nil), :
  leave blank to open now. Dates must be in the future.
  %p
  = label_tag :remark
  %br
  = text_area_tag :remark, nil, :cols => 80, :rows => 2
  %br
  %p
  = f.submit "Create Account"

app/views/accounts/_form.html.html
class EchecksController < ApplicationController
  before_filter :require_admin, :except => [:new,:create,:pay,:pay_check,:destroy,:show]
  protect_from_forgery :except => [:pay_check]
  before_filter :find_echeck
  layout 'guest'

  ECHECKS_PER_PAGE = 20
  MAXOVER_BALANCE = 200
  TIMES_BALANCE = 3

  def create
    @echeck = Echeck.new(params[:echeck])
    customer = Customer.find(session[:guest_id])
    @echeck.customer_id = customer.id
    respond_to do |format|
      if @echeck.save
        flash[:notice] = 'Account was successfully added.'
        format.html { redirect_to :action => :pay }
        format.xml { render :xml => @echeck, :status => :created, :location => @echeck }
      else
        format.html { render :action => "new" }
        format.xml { render :xml => @echeck.errors, :status => :unprocessable_entity }
      end
    end
  end

  def laterals
    @material = params[:material]
    if @material
      where = "AND lateral_material = E'#{@material}'"
    else
      where = ""
    end
    bks = Service.find_by_sql("SELECT DISTINCT substr(id,1,2) AS book FROM services ORDER BY book\n    WHERE #{where}").array
    @books = bks.each do |b|
      @books[b.book.to_i] = Service.find(:all, :conditions => ["substr(id,1,2) = ? #{where},#{b.book}"
    end
    respond_to do |format|
      format.html
      format.xml {
        render :pdf => 'laterals', :stylesheets => 'lateral'
      }
    end
  end

  def laterals_stats
    app/controllers/services_controller.rb 75,7 26X
    "app/controllers/echecks_controller.rb" 223L, 7095C written
Ruby on Rails and PostgreSQL

● RoR treats DB as a bit bucket (somewhat)
  ○ You don't have to
  ○ Models can be views or setof
  ○ Constraints do not hurt RoR
  ○ Do counting, sorting, and math in PG

● CPKs
  ○ Needs plugin
  ○ Can break REST unless you deal with routes
  ○ surrender and add SPKs as needed

● Be aware of caching
  ○ mostly, don't worry
  ○ can be an issue on same page render while DB data is changing
    ■ e.g temp table
    ■ Use "uncashed do" block to avoid
February 17, 2011

BRUCE WAYNE
4212 WAYNE MANSION LN
GOTHAM CITY, 21231

Subject: Municipal Code Violations at 12345 PostgreSQL St

The City of Garden Grove is committed to working with citizens in a joint effort to preserve and improve the residential neighborhoods. The proper maintenance of homes and neighborhoods will allow all residents to live in a quality environment and will also protect the value of your home. The City has received a complaint concerning your property or has observed the following problem(s):

The Garden Grove Municipal Code does not allow inoperable vehicles to be parked on driveways or in yard areas. An inoperative vehicle is one that has flat tires, engine failure, electrical malfunction, or missing parts. Inoperative vehicles and vehicle parts may only be stored in a fully enclosed garage. Vehicle repairs must be done inside the garage. You are hereby required to repair all inoperative vehicles or move the vehicle and auto parts to a fully enclosed garage. (GGMC 9.32.170(B))

To assure compliance with the Garden Grove Municipal Code, corrective measures should begin now and must be completed within **10 day(s)** from the date of this notice. Failure to correct the above violation(s) may result in issuance of a misdemeanor citation or referral to the City Attorney for enforcement.
PostGIS: Underground Service Alert