ZODB: A Persistent Graph of Python Objects

By

Christopher Lozinski

© Christopher Lozinski
CC BY-NC 3.0 US

PythonLinks.info/zodb #1
Why Use a Graph Database?

Social Network

Computer Network

© Christopher Lozinski

CC BY-NC 3.0 US

PythonLinks.info/zodb #2
Neo 4J Property Graph Database
Neo, Graphagus

Sara follows Joe.
Sara follows Ben.
Sara likes bikes.
Sara likes cars.
Sara likes cats.
Aria follows Joe.
Maria loves Joe.
Maria likes cars.
Joe follows Sara.
Joe follows Maria.
Joe loves Maria.
Joe likes bikes.
Joe likes nature.

© Christopher Lozinski CC BY-NC 3.0 US PythonLinks.info/zodb #3
ArrangoDB

{  
  title: "Data modelling",
  text: "lorum ipsum...",
  author: "Mike Williamson",
  date: "2015-11-19",
  comments: [
    {
      author:"Mike's Mum",
      email:"mikes_mum@allthemums.com",
      text: "That's great honey",
    },
    {
      "author" : "spammer@fakeguccihandbags.com",
      "title" : "Brilliant",
      "text" : "Gucci handbags...",
    }
  ],
  tags: ["mongodb","modeling","nosql"]
}

© Christopher Lozinski

PythonLinks.info/zodb #4
ZODB

Graph of Persistent Python Objects
Graphagus: A Property Graph Database on ZODB

Nodes

Sara
Ben
Martha
Joe

Cats
Cars
Bikes

Edges

Sara follows Joe.
Sara follows Ben.
Sara likes bikes.
Sara likes cars.
Sara likes cats.
Aria follows Joe.
Maria loves Joe.
Maria likes cars.
Joe follows Sara.
Joe follows Maria.
Joe loves Maria.
Joe likes bikes.
Joe likes nature.
For Managing Complexity
The best Videos of PyCon USA 2018 Organized by Category

Here you have the best rated videos from the conference. Swipe or arrow down for the best talks in each category.

Score
10 Best Talks in this Category
100% 108 0 Big-O: How Code Slows as Data Grows
100% 58 0 Easy 2D Game Creation With Arcade
100% 56 0 How Netflix does failovers in 7 minutes flat
100% 46 0 Keynote
100% 44 0 Talking Django Asyn
100% 42 0 Get your resources faster, with importlib.resources
100% 39 0 How to Write Deployment-friendly Applications
100% 36 0 A Bit about Bytes: Understanding Python Bytecode
100% 36 0 Analyzing Data: What pandas and SQL Taught Me About Taking an Average
100% 36 0 Inside the Cheeseshop: How Python Packaging Works

Child Categories
Talks about Data Science (26)
Talks about artificial Intelligence (6)
Distributed Systems (15)
Keynote, Concluding Remarks, and Lightning Talks (8)
Python in the Organization, and in the community (13)
Python Skills Development (40)
Python Software (58)
Talks related to Testing (6)
import persistent

class TreeLeaf(persistent.Persistent):
    def __init__(self, title=''):  
        self.title = title

def render(self):
    return self.title
ZODB Tutorial: Add Leaf Objects

<table>
<thead>
<tr>
<th>Single Leaf Object</th>
<th>Multiple LEAF Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>#CREATE A SINGLE LEAF OBJECT</td>
<td>#CREATE MULTIPLE LEAF OBJECTS</td>
</tr>
<tr>
<td>leaf = TreeLeaf('Leaf')</td>
<td>Leaf1 = TreeLeaf('Green Leaf')</td>
</tr>
<tr>
<td>root.leaf=leaf</td>
<td>leaf2 = TreeLeaf('Red Leaf')</td>
</tr>
<tr>
<td></td>
<td>#ADD THEM TO THE ROOT</td>
</tr>
<tr>
<td></td>
<td>root['leaf1'] = leaf1</td>
</tr>
<tr>
<td></td>
<td>root['leaf2'] = leaf2</td>
</tr>
</tbody>
</table>

© Christopher Lozinski
CC BY-NC 3.0 US
PythonLinks.info/zodb #10
import ZODB, ZODB.FileStorage

db = ZODB.db('Data.fs')
connection = db.open()
root = connection.root

#DO SOMETHING

transaction.commit()
## It is just Python

(No SQL SELECT)

<table>
<thead>
<tr>
<th>#UPDATE THE Leaf</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>root['leaf1'].title = &quot;Yellow Leaf&quot;</code></td>
</tr>
<tr>
<td><code>transaction.commit()</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#STUPID QUERY</th>
</tr>
</thead>
</table>
| For `key, item in root.items()`:
  | `print (key, item)` |

<table>
<thead>
<tr>
<th># DELETE AN OBJECT</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>del root['leaf1']</code></td>
</tr>
<tr>
<td><code>transaction.commit()</code></td>
</tr>
</tbody>
</table>
ZODB is Magical

Creates the illusion that your Python Objects are Persistent
ZODB is a graph databases

<table>
<thead>
<tr>
<th>CREATE THE OBJECTS</th>
<th>IT IS A GRAPH DATABASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaf1 = TreeLeaf('Green Leaf')</td>
<td>leaf1.sibling = leaf2</td>
</tr>
<tr>
<td>leaf2 = TreeLeaf('Red Leaf')</td>
<td>leaf2.sibling = leaf1</td>
</tr>
<tr>
<td>#ADD THEM TO THE ROOT</td>
<td>transaction.commit()</td>
</tr>
<tr>
<td>root['leaf1'] = leaf1</td>
<td></td>
</tr>
<tr>
<td>root['leaf2'] = leaf2</td>
<td></td>
</tr>
</tbody>
</table>
Hierarchical Calculations

class Video(Persistent):
    def countLeaves(self):
        return 1

class Category (Container):
    def countLeaves(self):
        total=0
        for item in self.values():
            total+=item.countLeaves()
        self.branchSize = total
        return total
The Best Videos on any branch of the tree

<table>
<thead>
<tr>
<th>Score</th>
<th>Votes</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>93👍</td>
<td>Interactive 3D Visualization in Jupyter</td>
</tr>
<tr>
<td>100%</td>
<td>77👍</td>
<td>Practical Sphinx</td>
</tr>
<tr>
<td>100%</td>
<td>76👍</td>
<td>Taking Django Async</td>
</tr>
<tr>
<td>100%</td>
<td>73👍</td>
<td>Hands-on introduction to Deep Learning with Keras and Tensorflow</td>
</tr>
<tr>
<td>100%</td>
<td>63👍</td>
<td>Conversational AI with Rasa Core &amp; NLU</td>
</tr>
<tr>
<td>100%</td>
<td>61👍</td>
<td>PyViz: Dashboards for Visualizing 1 Billion Datapoints in 30 Lines of Python</td>
</tr>
<tr>
<td>100%</td>
<td>52👍</td>
<td>Inside the Cheeseshop: How Python Packaging Works (PyCon US 2018)</td>
</tr>
<tr>
<td>100%</td>
<td>52👍</td>
<td>Optimization with Cython: Ising Models (Part 2)</td>
</tr>
<tr>
<td>100%</td>
<td>47👍</td>
<td>Building new NLP solutions with spaCy and Prodigy (PyData Berlin)</td>
</tr>
</tbody>
</table>
ZODB Uses Pickle

Pickle

Module for (de)serialization: Storing complete Python objects into files and later loading them back.

- Supports almost all data types – good.
- Works only with Python – bad.

```python
import pickle
pickle.dump(object, openBinaryFile)  # Save object to an open file
object = pickle.load(openBinaryFile)  # Restore an object from an open file
```
File Storage

Objects are Written to the end of a File
Ghost Objects

Object 1
Object 2
Object 3
Object 4

Object 2.1

Object 4
Ghost
FileStorage Transactions

Transaction 1          Transaction 3
Transaction 2          Transaction 4
Object Versions

Version 1

Version 2

Version 3
Rel Storage
Storing ZODB Objects in a Relational Database.
PostgreSQL, Oracle, MySQL

<table>
<thead>
<tr>
<th>Object Id</th>
<th>Version Number</th>
<th>Pickle</th>
</tr>
</thead>
</table>

© Christopher Lozinski

CC BY-NC 3.0 US

PythonLinks.info/zodb #22
ZEO
Client Server Storage

The load balancer alternates transparently between "bobbo", "pinky" and "erit".

ZEO Client #1
Cache

ZEO Client #2
Cache

ZEO Client #3
Cache

ZEO Server
ZODB Advantages

by Jim Fulton

No Database to administer
No Database Administrator
No Database Schema
No ORM
No Referential Integrity Problems
Automatic Garbage Collection
No manual reads and writes
Traversals

/ Software / Libraries / Database Libraries

© Christopher Lozinski

CC BY-NC 3.0 US

PythonLinks.info/zodb #26
Zopache Canonical URL’s
# The best Videos of PyCon USA 2018 Organized by Category

Manage CkEdit
Category: home

Here you have the best rated videos from the conference. Swipe or arrow down for the best talks in each category.

<table>
<thead>
<tr>
<th>Score</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>66👍</td>
</tr>
<tr>
<td>100%</td>
<td>56👍</td>
</tr>
<tr>
<td>100%</td>
<td>46👍</td>
</tr>
<tr>
<td>100%</td>
<td>35👍</td>
</tr>
<tr>
<td>100%</td>
<td>31👍</td>
</tr>
<tr>
<td>100%</td>
<td>29👍</td>
</tr>
<tr>
<td>100%</td>
<td>28👍</td>
</tr>
<tr>
<td>100%</td>
<td>26👍</td>
</tr>
<tr>
<td>100%</td>
<td>25👍</td>
</tr>
</tbody>
</table>
ZODB vs Relational Databases

Students Table

<table>
<thead>
<tr>
<th>Student</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>084</td>
</tr>
<tr>
<td>Jane Blogs</td>
<td>100</td>
</tr>
<tr>
<td>John Smith</td>
<td>182</td>
</tr>
<tr>
<td>Mark Antony</td>
<td>219</td>
</tr>
</tbody>
</table>

Activities Table

<table>
<thead>
<tr>
<th>ID</th>
<th>Activity1</th>
<th>Cost1</th>
<th>Activity2</th>
<th>Cost2</th>
</tr>
</thead>
<tbody>
<tr>
<td>084</td>
<td>Tennis</td>
<td>$36</td>
<td>Swimming</td>
<td>$17</td>
</tr>
<tr>
<td>100</td>
<td>Squash</td>
<td>$40</td>
<td>Swimming</td>
<td>$17</td>
</tr>
<tr>
<td>182</td>
<td>Tennis</td>
<td>$36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>219</td>
<td>Swimming</td>
<td>$15</td>
<td>Golf</td>
<td>$47</td>
</tr>
</tbody>
</table>
## ZODB vs Relational Databases

<table>
<thead>
<tr>
<th>ZODB</th>
<th>Relational Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td>for item in node: print (item)</td>
<td>You have to do a database join across every single table. 10 Tables: App, Category,</td>
</tr>
<tr>
<td></td>
<td>City, Company, Country, iFrame Link, Job, Link, Region, Product, and Video.</td>
</tr>
<tr>
<td>Zcatalog, Hypatia, repoze.catalog</td>
<td>Select statement</td>
</tr>
</tbody>
</table>
CREATING INDEXES
Using Repoze.catalog

1) catalog = Catalog()

2) def get_area(object, default):
   return getattr(object, 'area', default)

3) catalog['area'] = CatalogFieldIndex(get_area)

4) leaf-1 = Leaf(area=20)

5) catalog.index_doc(1, leaf-1)

6) catalog.reIndex (1, leaf-1)

7) numdocs, results = catalog.query(Range(20,40))
ZODB Storage Options

ZODB

ZEO

RelStorage

NewtDB

File and Blob Storage

Oracle

MySQL

PostgreSQL

© Christopher Lozinski
CC BY-NC 3.0 US
PythonLinks.info/zodb #32
NewtDB
Persisten Python + PostgreSQL Indexes
## Plone ZODB Users

### US Plone Sites
- Federal Bureau of Investigation (FBI)
- Central Intelligence Agency (CIA)
- Intellectual Property Rights Center
- US Department of Energy
- USDA Forest Service
- Fermi National Accelerator Lab (Fermilab)
- NASA Science
- Continental Airlines
- UCLA
- Yale University
- Harvard
- The Pennsylvania State University
- University of Notre Dame
- University of Virginia
- University of California - Davis
- University of North Carolina
- University of Louisville
- Novell
- Akamai
- eBay
- Google
- Walmart
- Marriott
- ...and many more.

### Worldwide Plone Sites
- Brazilian Government
- 2016 Olympics Brazil
- The British Postal Museum and Archive
- The New Zealand Treasury
- Konica Minolta Printers - Australia
- National Sports Council - Spain
- National Library of South Africa
- University of Oxford
- University of Toronto
- Academy of Performing Arts - Prague
- Open Society Foundation
- Amnesty International
- OXFAM
- Lufthansa
- Nokia
- Clean Clothes Campaign
- RIPE
- Cambridge University
- Royal College of Surgeons
- Oxford University Clinical Academic Graduate School
- ... and many more

© Christopher Lozinski

CC BY-NC 3.0 US

PythonLinks.info/zodb #34
Speed
by Jim Fulton

1000’s of Transactions per second
For simple transactions relational databases are slightly faster
For complex transactions ZODB is faster.

Try to write 10 different classes, in ZODB just append to a file, an RDB requires 10
different writes.
Scalability

Several hundred newspaper content-management systems and web sites were hosted using a multi-database configuration with most data in a main database and a catalog database. The databases had several hundred gigabytes of ordinary database records plus multiple terabytes of blob data.

For larger systems move to NEO (neo.nexedi.com). Up to 80TB in production. 150TB (and growing) in test. Of course it takes time to move that much data around.
Number of Objects

18,446,744,073,709,551,616 Objects
Security

ZeroDB Decrypts Everything on the Client.
MQTT Pub/Sub For Real-Time Chat
MQTT.org

Subscribers

© Christopher Lozinski
CC BY-NC 3.0 US
# ZODB is for Small Companies

<table>
<thead>
<tr>
<th>Big Companies</th>
<th>Small Companies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Big Data</td>
<td>Quality Data</td>
</tr>
<tr>
<td>Database-Centric</td>
<td>Object-Centric</td>
</tr>
<tr>
<td>Manage Complexity</td>
<td>KISS. <a href="https://www.pythonlinks.info/zodb">Pug</a></td>
</tr>
<tr>
<td>Armies of Transient Developers</td>
<td>One Senior Dedicated Developer</td>
</tr>
<tr>
<td>Fad Driven</td>
<td>Technology Driven</td>
</tr>
</tbody>
</table>
Contact Information
Follow @PythonLinks on Twitter

Christopher Lozinski
Http://PythonLinks.info
EMail: lozinski@PythonLinks.info
Twitter: @PythonLinks
Skype: clozinski
US Phone: +1 (650) 614 1836
EU Phone: +48 12 361 3136

© Christopher Lozi
Chat Logs After Batching

Object 1    Object 3
          Object 2

Object 1
Version 2

Object 2
Version 2

Object 3
Version 2
Follow @PythonLinks on Twitter

© Christopher Lozi

Python Links.info
Contact Information

Follow @PythonLinks on Twitter

Christopher Lozinski
Http://PythonLinks.info
EMail: lozinski@PythonLinks.info
Twitter: @PythonLinks
Skype: clozinski
US Phone: +1 (650) 614 1836
EU Phone: +48 12 361 3136
PythonLinks is a Content Aggregation System

PyCon UK
PyCon CZ
PyData
YouTube
SciPi
EuroPython
PyCon US
PythonLinks.info

© Christopher Lozinski
CC BY-NC 3.0 US
Content Aggregation System

Individual Blogories

- Github Markdown
- Web GUI
- Blog RSS
- Plone XML

Group Blogory

Merged Tree

Plone is a Content Management System.

Blogory is also a Content Aggregation System