

# How to Import Large Data into Odoo

Thibault Francois • Technical consultant Team  
Leader

# Agenda

- ① Choose the right tool
- ② Load method
- ③ Speed up the import
- ④ Implementation
- ⑤ Conclusion



Choose the right  
tool

# Available Tools

- Import wizard in odoo
  - Slow, timeout, no automation
- SQL query
  - The fastest, but bypass the orm, need direct access to database
- RPC: Create and Write
  - Work everywhere, complicated, 1 record = 1 Call
- RPC: Load

# RPC api: Load method

- Exact same behavior as import wizard
- Can be executed in batch : 1 call = N records
- Take care of linking data based on external ID
  - Id defined prior the import
- No need to worry about creation or update
- Can be used for incremental import
- Can be automated

Let's see how to use it with a large amount of data



# Load method : Howto

# Method Signature

```
class BaseModel(object):
    @api.model
    def load(self, fields, data):
        """
        Attempts to load the data matrix, and returns a list of ids (or
        ``False`` if there was an error and no id could be generated) and a
        list of messages.

        The ids are those of the records created and saved (in database), ...

        :param fields: list of fields to import, at the same index as the corresponding data
        :param data: row-major matrix of data to import
        :returns: {ids: list(int)|False, messages: [Message]}
        """
```

## How to call it with xmlrpc api:

```
models = xmlrpclib.ServerProxy('{} /xmlrpc/2/object'.format(url))
models.execute_kw(db, uid, password, 'res.partner', 'load', [
    #Header
    ['id', 'name'],
    #Data
    [['my_partner.external_id1', 'John Doe'],
     ['my_partner.external_id2', 'Foo Bar'], ]
], {'context': {}})
```

# How to load 50K Partners

You cannot load 50K lines with one call

- Split your list in small batch
  - Too small batch lead to a lot of transaction and network overhead
  - Too big batch lead to timeout or memory error
- If one line failed, the whole batch rollback
  - Need to keep track of failed batch
  - Need to replay with smaller batch size: ideally 1



# Ideal Batch Size

On laptop, 10K partner

Batch Size	Total Time (10K)	Per record (10K)
1	589s	59ms
10	381s	38.1ms
50	310s	31ms
100	316s	31.6ms
200	303s	30.3ms
500	289s	28.9ms
1000	303s	30.3ms

Good batch size is between 50 and 500 lines.  
Adjust to data quality.



# How to speed up the import

# On RPC Client's layer

- Send simultaneous request
  - Depend on the number of workers available on the server
  - Risk of concurrent update
- Do not import compute or related field
- Some value in the context allow to avoid unnecessary or unwanted behavior

```
{
  #Prevent to create mail message
  'tracking_disable' : True,
  #Handle multi company
  'force_company' : company_id,
  #since V10 : allow to append value to many2many
  #[(4, id1, _), (4, id2, _),...] instead of [(6, 0, ids)]
  'update_many2many': True,
  #Other sepcific to some models
  ...
}
```

# On Odoo Server's layer

- Send simultaneous request does not help if there is no worker to answer on the server
- nb worker  $\geq$  nb concurrent request
- Rule: nb worker =  $2 * \text{nb physical core} + 1$
- You may need to leave worker for others

# Number of concurrent request

4 Cores (HT), batch size=100

100K record		
Con. Req.	Total	per record
1	1495s	15ms
2	762s	7.6ms
4	429s	4.3ms
6	367s	3.7ms
8	342s	3.4ms
9	341s	3.4ms
15	372s	3.7ms

9 workers give the best result

# On Postgresql's Layer

- Use **PgTune** to set up memory parameters

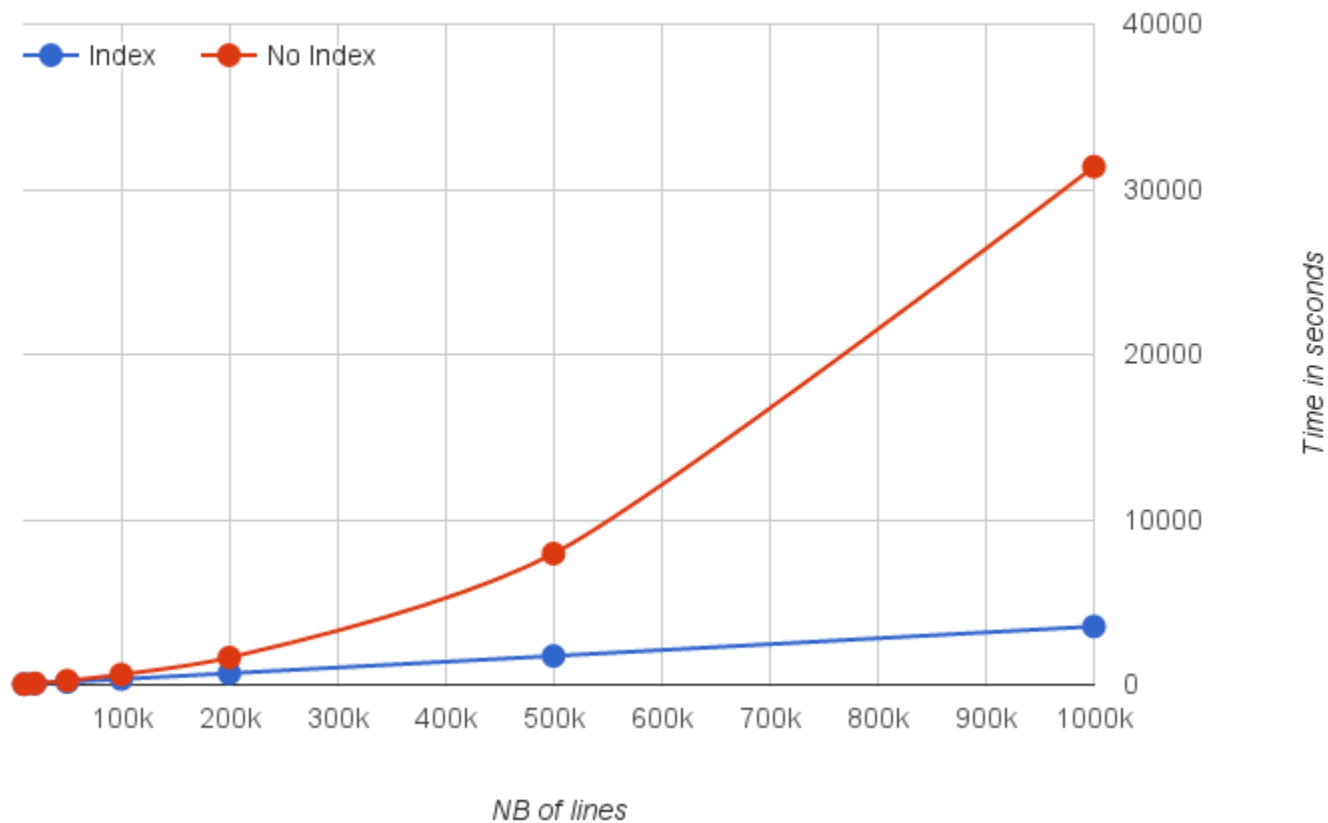
```
#Example for 8GB system
shared_buffers = 2GB
effective_cache_size = 6GB
work_mem = 64MB
maintenance_work_mem = 512MB
wal_buffers = 16MB
```

- For import you can disable synchronous\_commit

```
synchronous_commit = off
```

- Don't forget to reactivate it after your import
- Indexes can be missing : time per batch increase during the process
  - Find them with pgbager
  - Add them

# Missing commercial\_partner\_id index: $O(n)$ vs $O(n^2)$



# 4 Implementation



# odoo\_import\_thread.py

- available on github [tfrancoi/odoo\\_csv\\_import](#)
- Read a CSV file cut into batch
- Import batches with x simultaneous threads
- Store failed batch in a new CSV
- Based on [openerp-client-lib](#)
- Work on Odoo saas

# Usage

```
-c CONFIG, --config CONFIG      Configuration File that contains connection parameters
--file FILENAME                 File to import
--model MODEL                   Model to import
--worker WORKER                 Number of simultaneous connection
--size BATCH_SIZE               Number of line to import per connection

--context CONTEXT               context that will be passed to the load function, need
                                to be a valid python dict
```

## Connection parameter file

```
[Connection]
hostname = localhost
database = load
login = admin
password = admin
protocol = xmlrpc
port = 8069
uid = 1
```

# Advance Usage

```
--fail           Fail mode

--groupby GROUP  Group data per batch with the same value for the given
                 column in order to avoid concurrent update error
--ignore IGNORE  list of column separate by comma. Those column will be
                 remove from the import request
```

- Fail mode use as input the output of the normal mode, set worker and batch size to 1
  - Extract in a third file the real wrong lines
- groupby allow to group record that have the same value for a given column in the same batch. If the batch exceed batch\_size, it's cut into sub batch imported sequentially
  - If you pick the right field it help to mitigate the concurrent update problem
  - For example you want to group your sale order line by sale order

# Demo

# Conclusion

- RPC call Load work well and fast
- Feel free to use `odoo_import_thread.py`
- Script everything you'll probably need to run the import more than once
- Export can be speed up the same way
- We only discuss the loading aspect of data migration problem
- The extraction and transformation remain the hardest part

# Any question?



#odooexperience

Thibault Francois (tfr@odoo.com)

 @tfrancoi