

py-pointless

A data serializer for Python/C

<http://code.google.com/p/py-pointless/>

Árni Már Jónsson, Dohop ehf.

arnimarj@gmail.com

Hvað?

- skrifar/les gögn, eins og JSON/Pickle
- sömu gagnatög og JSON, plús:
 - bitafylki, mengi, o.fl.
- mjög einfalt Python viðmót

Einfalt dæmi

```
>>> import pointless
```

```
>>> value = {'hello': 'world'}
```

```
>>> type(value)
```

```
<type 'dict'>
```

```
>>> pointless.serialize(value, 'file.pointless')
```

```
>>> value = pointless.Pointless('file.pointless').GetRoot()
```

```
>>> value
```

```
{'hello': 'world'}
```

```
>>> type(value)
```

```
<type 'pointless.PyPointlessMap'>
```

Mikilvægu smáatriðin

- eftir að skrár eru skrifaðar eru þær:
 - **read-only + relocatable**
 - eitt `mmap()` kall og sanity-check til að hlaða
- engin auka minnisþörf fyrr en einhver skoðar gögnin

Annað dæmi

```
>>> import pointless, sys
>>> value = [0 for i in xrange(10000000)]
>>> pointless.serialize(value, 'big.pointless')
>>> handle = pointless.Pointless('big.pointless')
>>> vector_proxy = handle.GetRoot()
>>> sys.getsizeof(value)
81528064
>>> sys.getsizeof(handle)
10000244
>>> sys.getsizeof(vector_proxy)
56
```

Af hverju ? (1/2)

- indexing ferlið 2008:
 - mörg Python/C tól
 - nokkur sérhönnuð skráarsnið í C
 - CSV og MySQL þarna líka
- í dag lesa öll tól py-pointless skrár
 - minni boilerplate kódi
 - auðveldara að gera tilraunir

Af hverju ? (2/2)

- leitarþjónar 2008:
 - margar stórar gagnagrindur
 - mikið minnis-overhead per instance
 - mjög lengi að ræsa (2 mín)
- í dag eru þær allar í py-pointless
 - Python kóðinn breyttist ekkert
 - sparar 4GB minni per instance

Önnur notkun

- snapshot af stateful server
 - gagnagrind í minni
 - `fork()`-a reglulega og skrifa út state
 - hleður yngast snapshotti í ræsingu
- útdeiling á reiknuðum gögnum
 - hleður gögnum eftir `inotify` callback
 - py-pointless sleppir GL á meðan

Spurningar?