PyPy 101

Berker Peksağ

26 Mayıs 2012

http://berkerpeksag.github.com/slides/pypy-101-pyist/

Berker Peksağ

- berkerpeksag.com
- @berkerpeksag
- github/berkerpeksag
- mozillians.org/berkerpeksag

Başlamadan Önce

- Bazı yerlerde fikir sahibi olunması için PyPy kaynak kodundan ufak örnekler verdim. Kodun tamamını okumak isteyenler için bağlantılarını da ekledim.
- Mümkün olduğunda Türkçe kelimeler kullanmaya çalıştım ancak *port*, *virtual machine*, *backend* vb. gibi artık alıştığımız ve Türkçesi pek *tercih edilmeyen* kelimeleri olduğu gibi kullandım.
- İmplementasyon Türkçe değil ama *kulak aşinalığı* nedeniyle sunum boyunca bol bol kullandım.

CPython == Python

- **CPython**, Python programlama dilinin **Guido van Rossum** tarafından `C` ile yazılan referans implementasyonu.

The Zen of Python vs. PyPy

If the implementation is hard to explain, it's a bad idea.

The Zen of Python vs. PyPy

If the implementation is hard to explain, it's a bad idea.

Except PyPy![1]

PyPy Öncesi

- Diğer tüm dinamik ve yorumlanabilir programlama dilleri gibi Python'un da yumuşak karnı hızı.
- PyPy bu açığı kapatmak için geliştirilmiş ilk proje değil.
- İlk örnekler **Psyco** ve **Stackless Python** projeleri.
- Son olarak yine JIT derleyicisi çözümünü kullanan **HotPy** projesi duyuruldu.

PyPy Öncesi

Psyco

- http://psyco.sourceforge.net/
- JIT derleyicisi
- C ile CPython eklentisi olarak yazıldı.
- Python 3 ve 64 bit desteği yok.
- Artık geliştirilmiyor.

PyPy Öncesi

Stackless Python

- http://www.stackless.com/
- CPython'dan dallanmış bir proje.
- Microthread.
- Coroutine.
- Google tarafından desteklendi.
- İmplementasyonun karmaşıklığı nedeniyle CPython ile birleştirilmesinden vazgeçildi.
- Halen bir grup gönüllü tarafından geliştiriliyor.
- Python 3 desteği var.
- Microthread desteği daha sonra CPython'a C modülü olarak **greenlet** adıyla port edildi.

PyPy Nedir

Temelde iki ana projeden oluşur:

1. **PyPy:** Python Virtual Machine.
2. **RPython:** Kendi VMlerinizi yazmak için bir dil.

PyPy Nedir

Ekip

- Çoğunluğu dinamik programlama dilleri implementasyonu ve JIT derleyicileri üzerine doktora veya doktora sonrası araştırma yapıyor.
- Geliştiricilerinden ikisi CPython projesinde de çekirdek/ana ekipte yer alıyor.

PyPy Nedir

İmplementasyon örnekleri

- JavaScript
- Haskell
- Scheme
- Smalltalk
- Converge

PyPy Nedir

CPython ile farkları

- `_md5`, `_io`, `_os` gibi orijinal olarak C ile yazılmış CPython modülerinin tamamı kullanılabilir...
- ...ancak performans nedeniyle(*GIL!*) tüm C modülleri RPython ile sıfırdan yazıldı.
- Eğer CPython için C ile yazılmış bir modülünüz varsa PyPy üzerinde kullanmak için RPython ile sıfırdan yazmak daha performanslı olacaktır.

RPython

- **Restricted Python**
- Bildiğimiz Python ile sözdizimsel olarak hiçbir farkı yok...
- **Static-typed** olması dışında! (CPython *duck-typed*)
- Ayrıca, `os`, `math` ve `time` modülleri haricinde CPython ile birlikte gelen modülleri kullanamazsınız(İstisna olarak C ile yazılan `_md5` gibi modüller gösterilebilir).

RPython

Yerleşik fonksiyonlar

- `int`, `float`, `str`, `ord`, `chr`, `range` vb. kullanılabilir.

Örnek: `int` fonksiyonunun RPython implementasyonu.

`pypy/annotation/builtin.py`

```
 1 def builtin_int(s_obj, s_base=None):
 2     if isinstance(s_obj, SomeInteger):
 3         assert (not s_obj.unsigned,
 4                 "instead of int(r_uint(x)),use intmask(r_uint(x))")
 5     assert (s_base is None or isinstance(s_base, SomeInteger)
 6             and s_obj.knowntype == str),
 7             "only int(v|string) or int(string,int) expected"
 8     if s_base is not None:
 9         args_s = [s_obj, s_base]
10     else:
11         args_s = [s_obj]
12     nonneg = isinstance(s_obj, SomeInteger) and s_obj.nonneg
13     return constpropagate(int, args_s, SomeInteger(nonneg=nonneg))
```

RPython

- Garbage collector
- Veri yapıları: `dict`, `tuple`, `list` vb.
- RPython, yazdığınız RPython programını C'ye çevirir.
- VM yazdığınız programlama dili için JIT derleyicisi var.

RPython

Backendler

- **C:** Varsayılan olarak çeviri işlemi C backend'i ile yapılır.
- **JVM:** Büyük oranda Polonyalı bir öğrenci tarafından yazıldı, şu an aktif olarak geliştirilmiyor.
- **CLR:** PyPy geliştiricilerinden Antonio Cuni'nin doktora tezi olarak yazdığı backend. Yine pek aktif olarak geliştirildiği söylenemez.

RPython

JIT derleyicisi

- Hibrid bir teknoloji.
- JVM ile programlama dilleri için geliştirilen derleyicilerde kullanım oranı arttı.
- SpiderMonkey ile birlikte dinamik programlama dillerinin implementasyonunda *yeni moda* oldu.

RPython

JIT derleyicisi

- Programlama dillerinde bulunan güzel özelliklerin çoğu gibi JIT derleyicisi ile ilgili ilk çalışmalar da, **LISP**'in *mucidi* **John McCarthy** tarafından yapıldı.
- *Yorumlanan programlarda*, program her çalıştığında tekrar tekrar makine koduna çevrilir.
- *Derlenen programlarda* ise, çalışma zamanından önce makine koduna çevrilir.
- Bir JIT derleyicisi, basitçe bu iki kavramın karışımdan oluşur.

RPython
Örnek

- RPython ile implemente edilen `_md5` modülünü kullanmak için yazılan bir wrapper.

`pypy/module/_md5/interp_md5.py`

```
 1 from pypy.rlib import rmd5
 2 from pypy.interpreter.baseobjspace import Wrappable
 3 from pypy.interpreter.typedef import TypeDef
 4 from pypy.interpreter.gateway import interp2app, unwrap_spec
 5
 6 class W_MD5(Wrappable, rmd5.RMD5):
 7     def __init__(self, space):
 8         self.space = space
 9         self._init()
10
11 # ...
```

RPython
Örnek

- CPython için C ile yazılan `_md5` modülünün RPython ile implemente edilmiş versiyonu.

`pypy/rlib/rmd5.py`

```
 1 class RMD5(object):
 2     """RPython-level MD5 object.
 3     """
 4     _mixin_ = True        # for interp_md5.py
 5
 6     def __init__(self, initialdata=''):
 7         self._init()
 8         self.update(initialdata)
 9
10     def _init(self):
11         """Set this object to an initial empty state.
12         """
13         self.count = r_ulonglong(0)   # total number of bytes
14         self.input = ""   # pending unprocessed data, < 64 bytes
15         self.uintbuffer = [r_uint(0)] * 16
16
17  # ...
```

RPython

Eksiler

- Belgeler epey detaylı ama düzensiz.
- Sanki dünyanın sonu gelmeden önce apar topar yedek alınmış gibi!

RPython

Eksiler (2)

- Virtual Machine her güncellendiğinde tüm programın en baştan çevirilmesi gerekiyor.
- PyPy VM'i için bu süre, iyi bir bilgisayarla **~40 dakika**!

PyPy'ın Geleceği

Geliştirilmesi devam eden üç farklı büyük proje var:

1. NumPy
2. Python 3
3. Software Transactional Memory

PyPy'ın Geleceği

NumPy

- http://buildbot.pypy.org/numpy-status/latest.html
- En aktif PyPy projesi.
- ~50%'lik bölümü port edildi.

PyPy'ın Geleceği

Python 3

- https://bitbucket.org/pypy/pypy/src/py3k
- Çoğu yerleşik fonkisyon kaldırıldı ya da güncellendi.
- Tüm testler Python 3'e port ediliyor.
- `str` vs. `bytes` vs. `unicode`.
- Python 3 sözdiziminin implementasyonu(yeni *Metaclass* sözdizimi gibi) tamamlandı.

PyPy'ın Geleceği

Software Transactional Memory

- https://bitbucket.org/pypy/pypy/src/stm-thread
- **Global Interpreter Lock** yerine alternatif bir çözüm.
- Direkt hafızaya erişmek yerine yerel bir thread'de gerekli işlemleri yapar.

Örnek:

```
1 def f(list1, list2):
2     x = list1.pop()
3     list2.append(x)
```

STM ile:

```
1 def f(list1, list2):
2     while True:
3         t = transaction()
4         x = list1.pop(t)
5         list2.append(t, x)
6         if t.commit():
7             break
```

PyPy'ın Geleceği

Eksiler

- Pek çok senaryoda Python'a göre *~5.5 kat* hızlı ama halen PyPy bug tracker'ı *"bu kod PyPy'da CPython'dan daha yavaş çalışıyor"* gibi hata kayıtlarıyla dolu.
- PyPy list element del insert too slow
- Significantly slow joins
- PyPy is slower on longs than it should

PyPy'ın Geleceği

Eksiler (2)

- Geliştirilmesi gönüllülük esasından ziyade bağışlara göre devam ediyor. Günün birinde bağışlar kesilirse projenin devam edip etmeyeceği meçhul.
- Yeni gelenlere karşı pek yardımsever değiller. CPython bu açıdan pek çok açık kaynak projeye örnek olmalı.

Kaynaklar

- PEP 20 -- The Zen of Python

PyPy

- http://pypy.org/
- http://speed.pypy.org/
- http://doc.pypy.org/
- http://morepypy.blogspot.com/
- http://bugs.pypy.org/
- Makaleler

1. Kenneth Reitz ve Benjamin Peterson'ın sunumlarından alıntı. ?

Source: slides.md
30/30