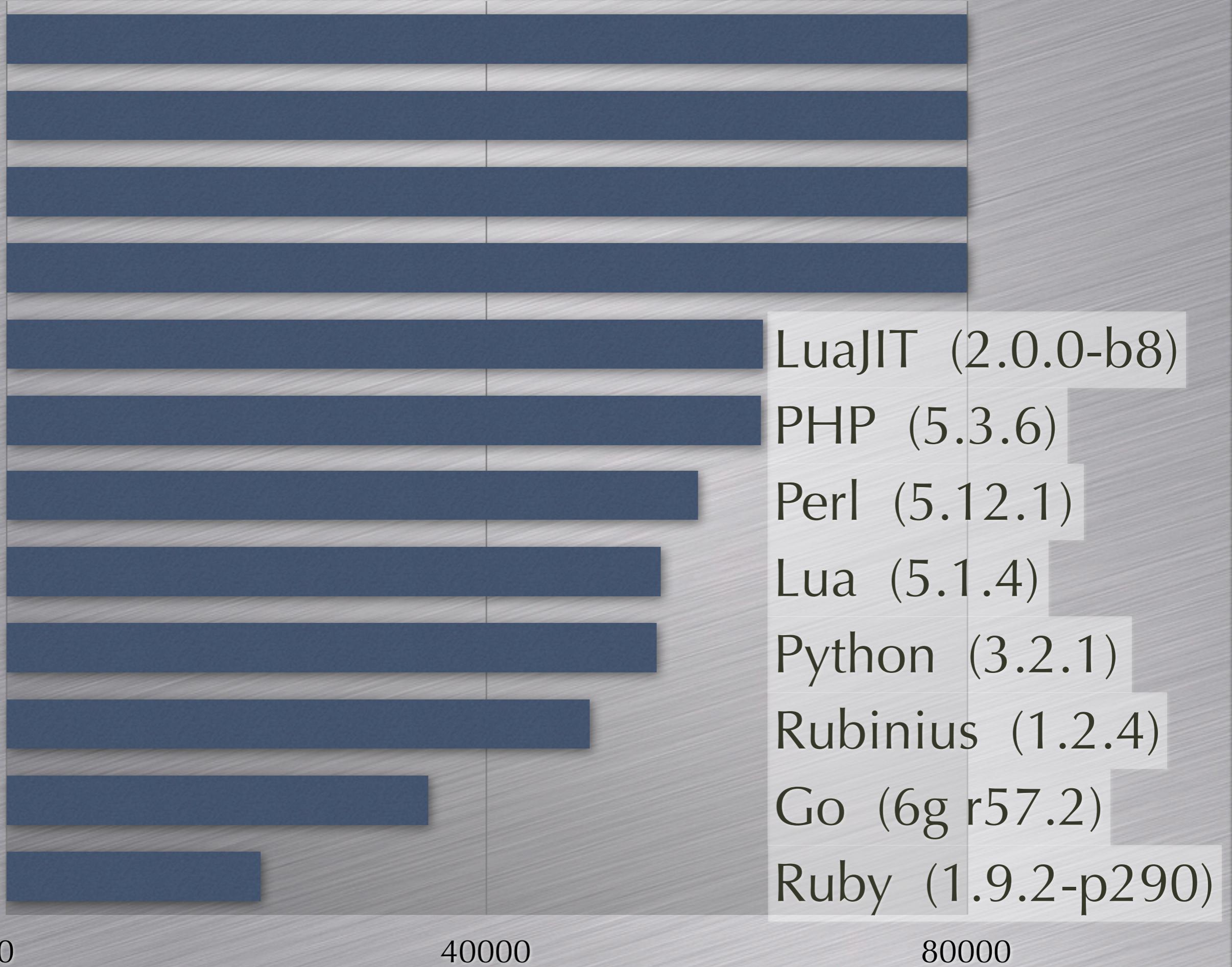
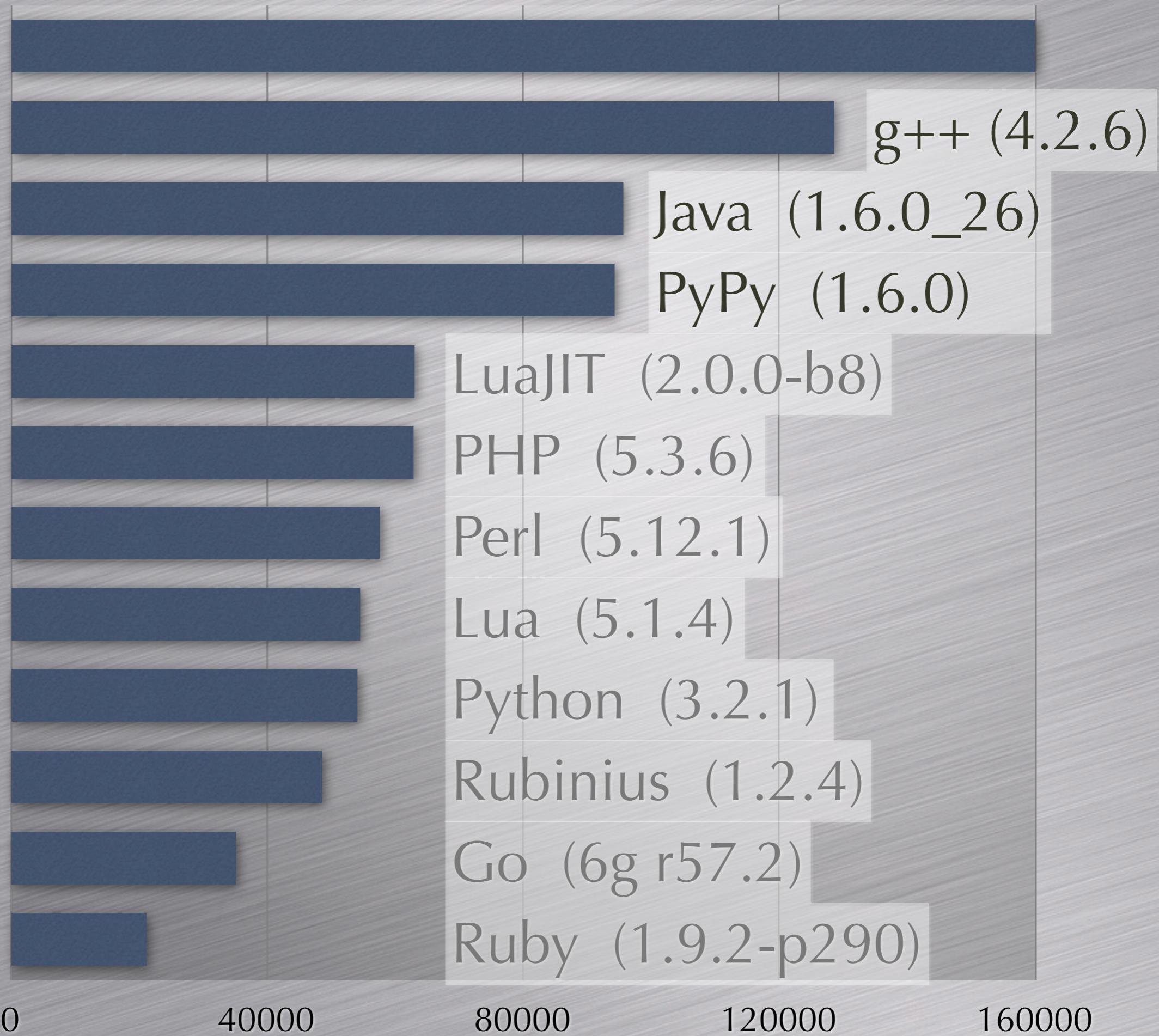
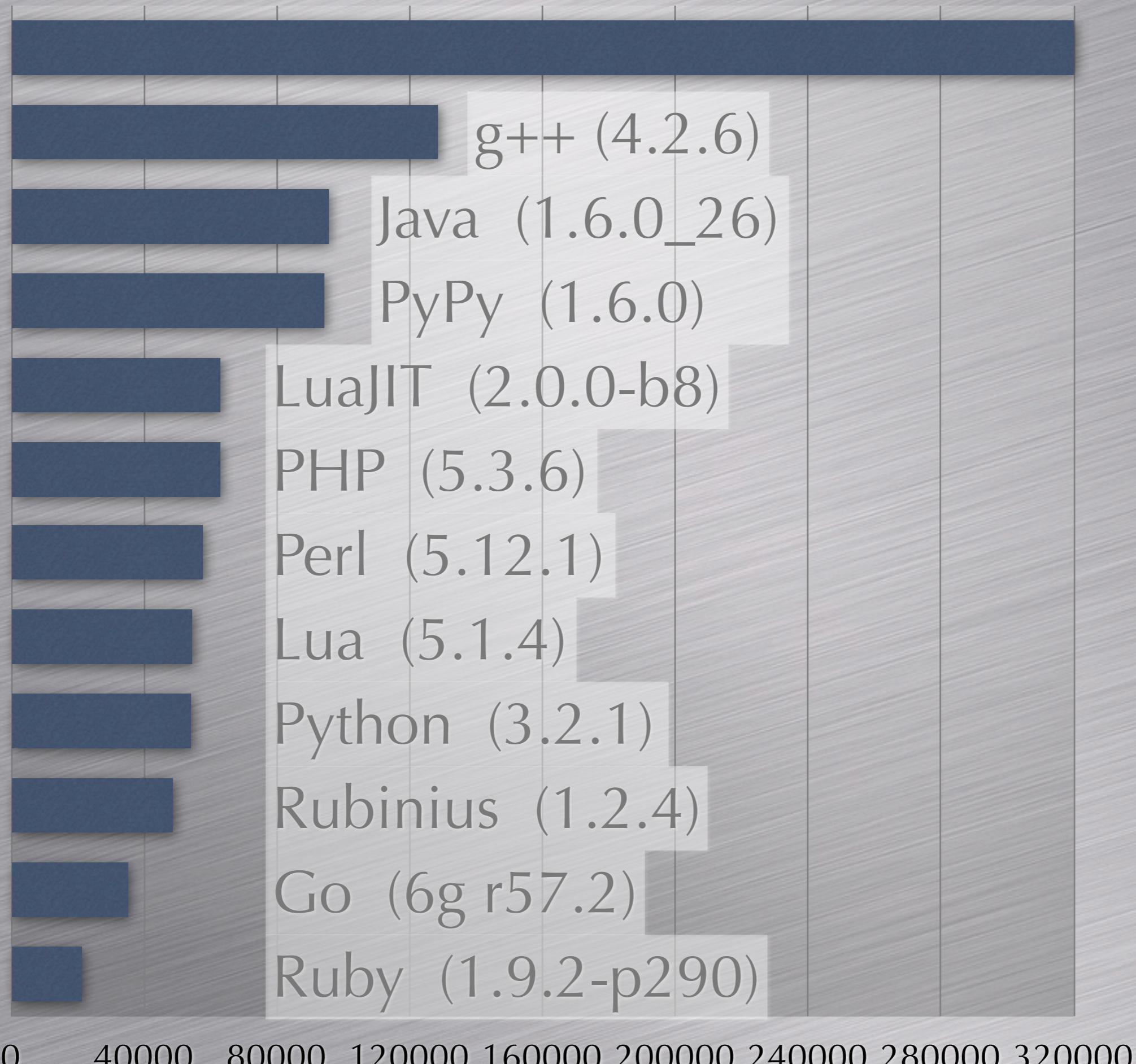


文字列結合の
ベンチマークを
いろんな処理系で
やってみた

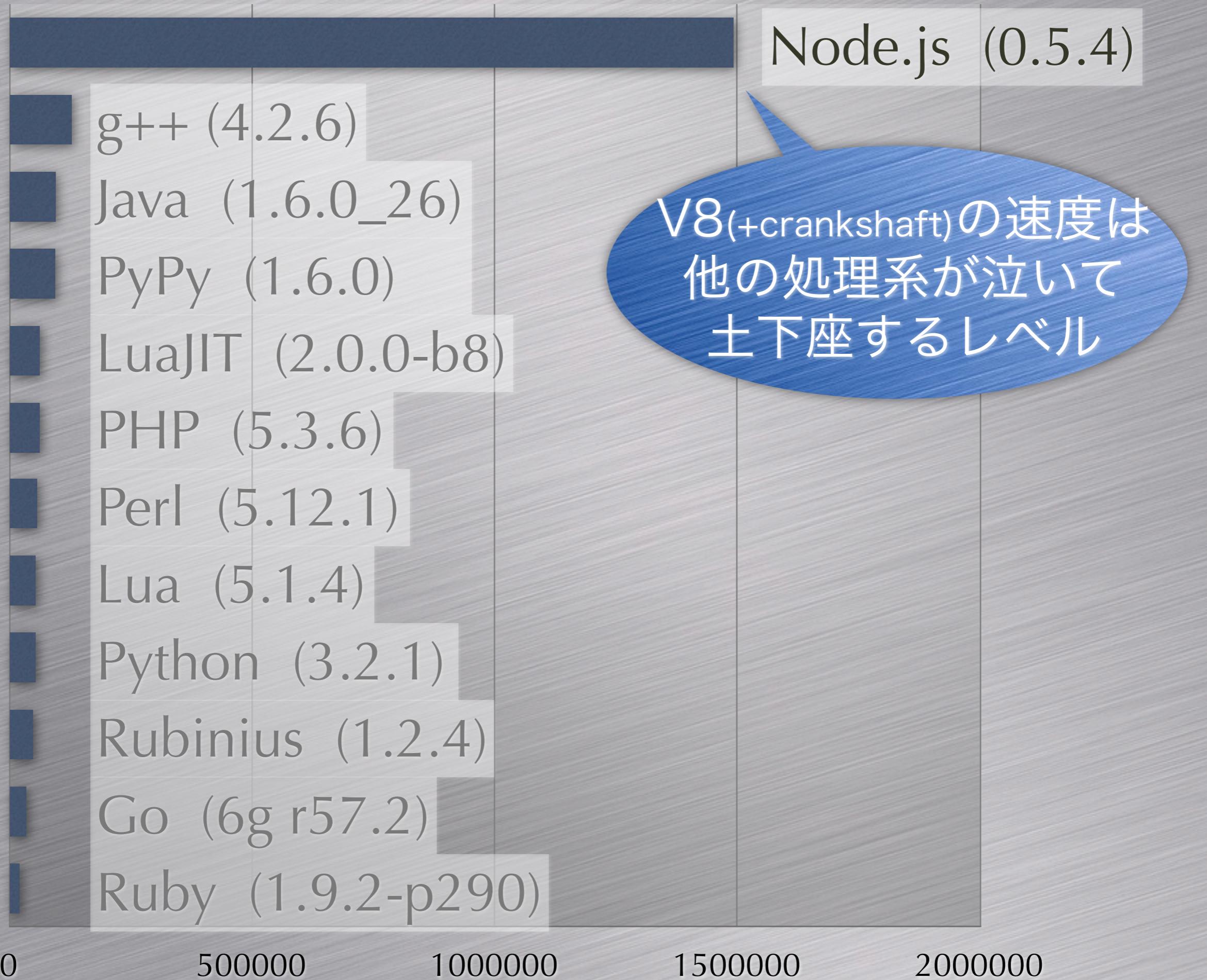
- HTMLテーブルを生成
 - 5項目 × 20行 × 10万回
- ループ時間を取り除いた、正味の時間
- 3回繰り返して中央値を採用
- Mac OS X 10.6, Core2 Duo 2GHz
- <http://bit.ly/pNNmGY>











# cycle=1	actual
push(s,s,s);join('')	0.977
buf+=s;buf+=s;buf+=s	0.471
buf+=s+s+s	0.465

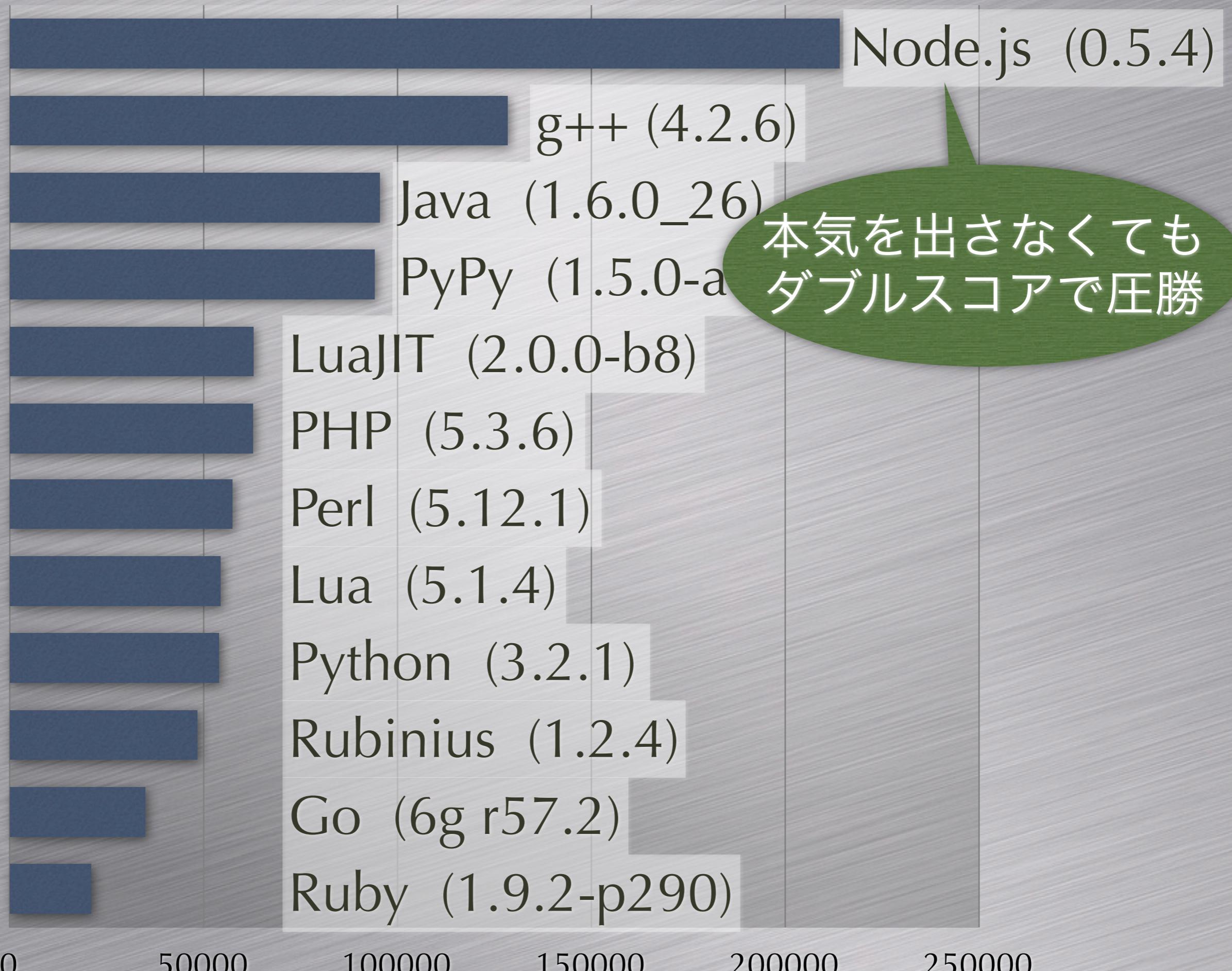
1回目は
まだ本気を
出してない

# cycle=2	actual
push(s,s,s);join('')	1.008
buf+=s;buf+=s;buf+=s	0.479
buf+=s+s+s	0.066

2回目から
本気を出す
(Tracing JIT)

# cycle=3	actual
push(s,s,s);join('')	1.004
buf+=s;buf+=s;buf+=s	0.463
buf+=s+s+s	0.066

Node.js 0.5.1
以降で確認



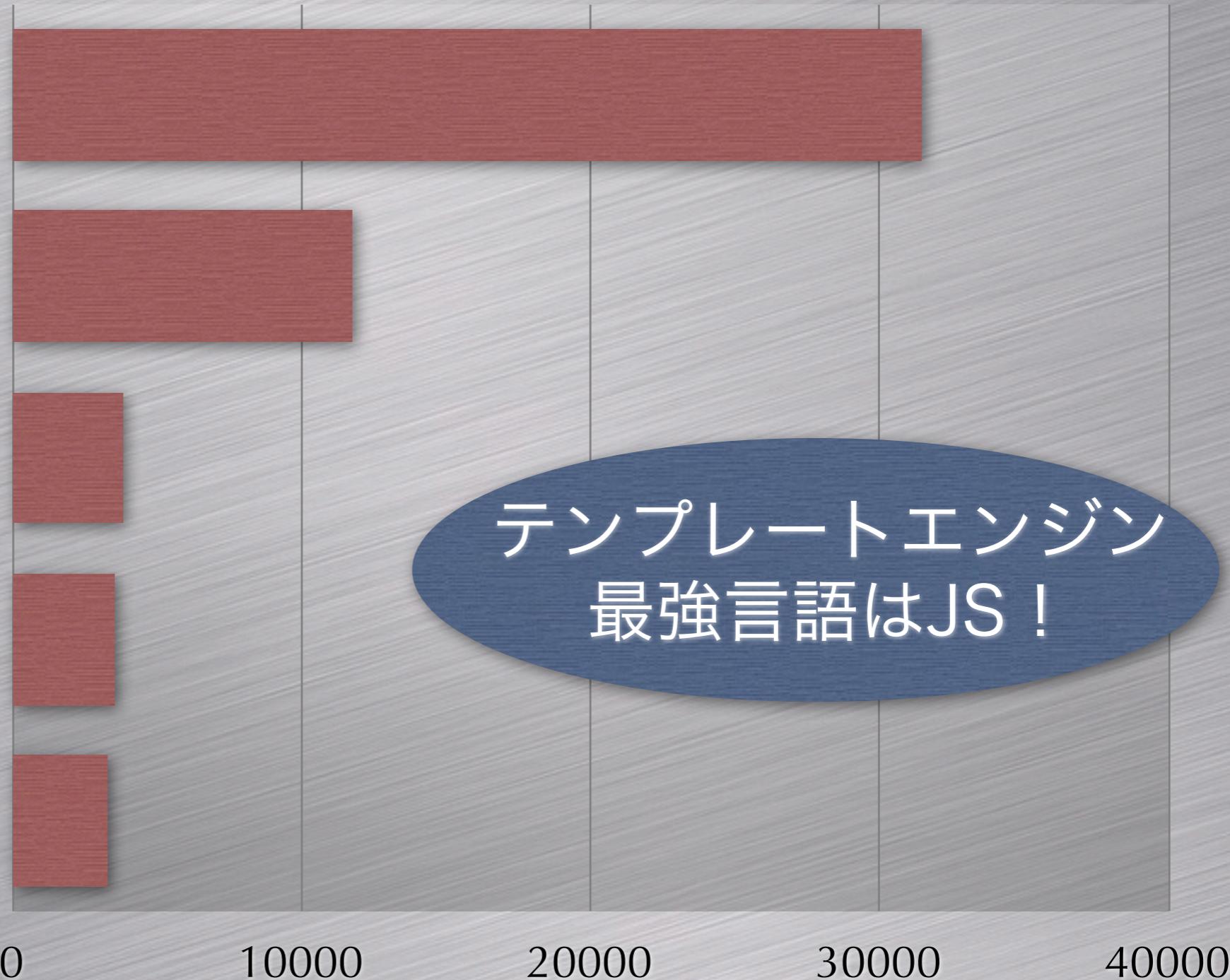
Tenjin (node.js)

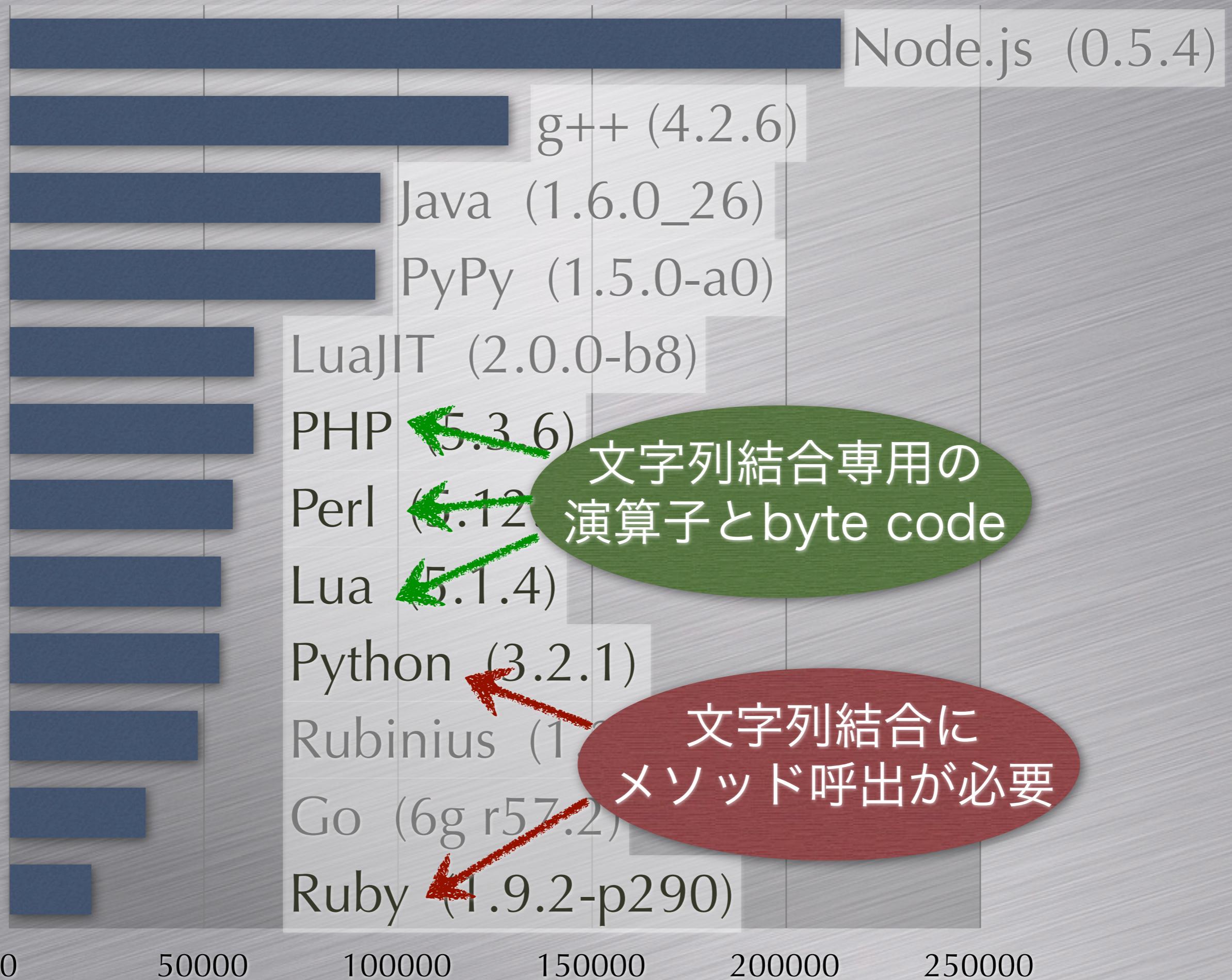
Tenjin (perl)

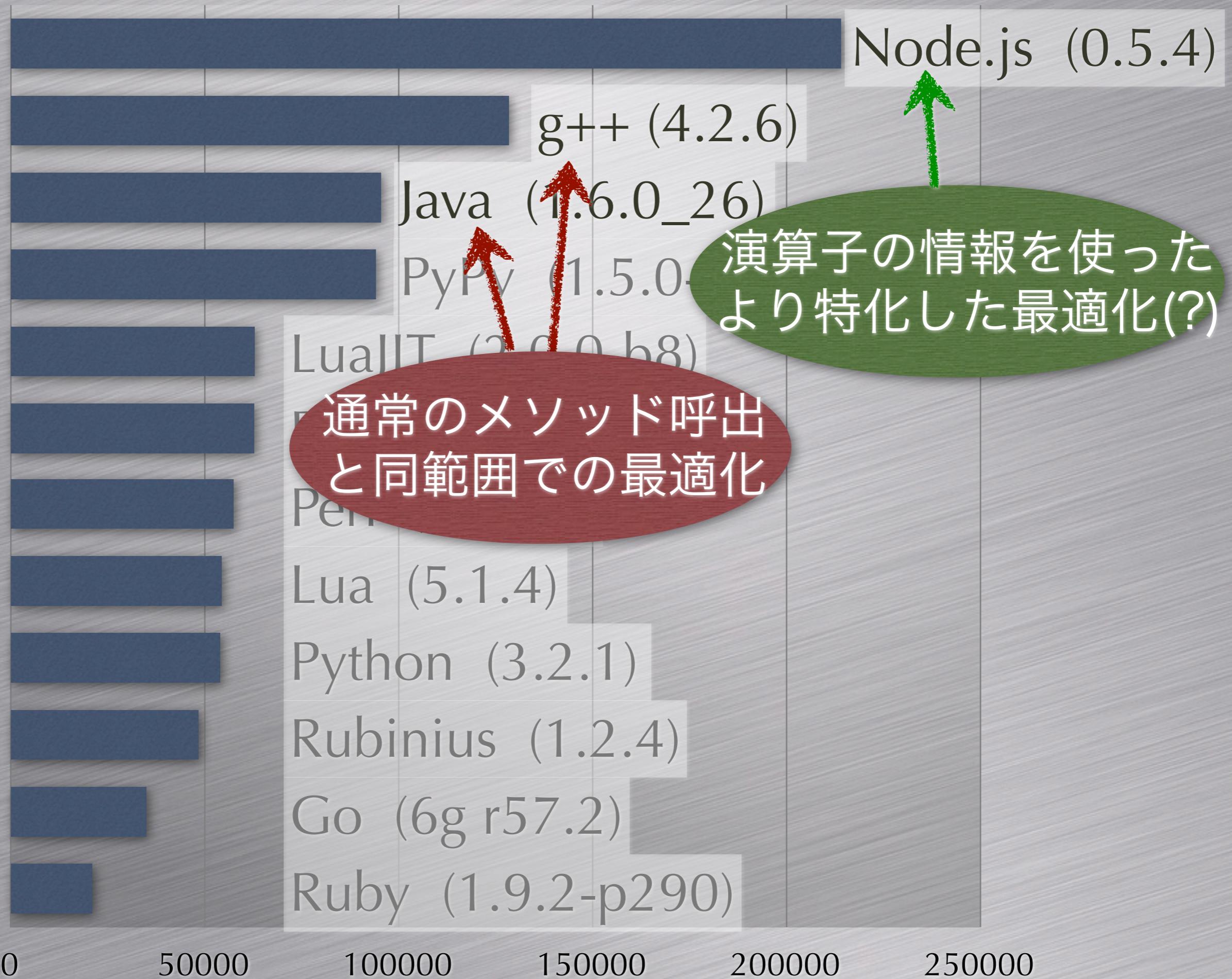
Tenjin (php)

Tenjin (ruby)

Tenjin (python)







g++: std::string

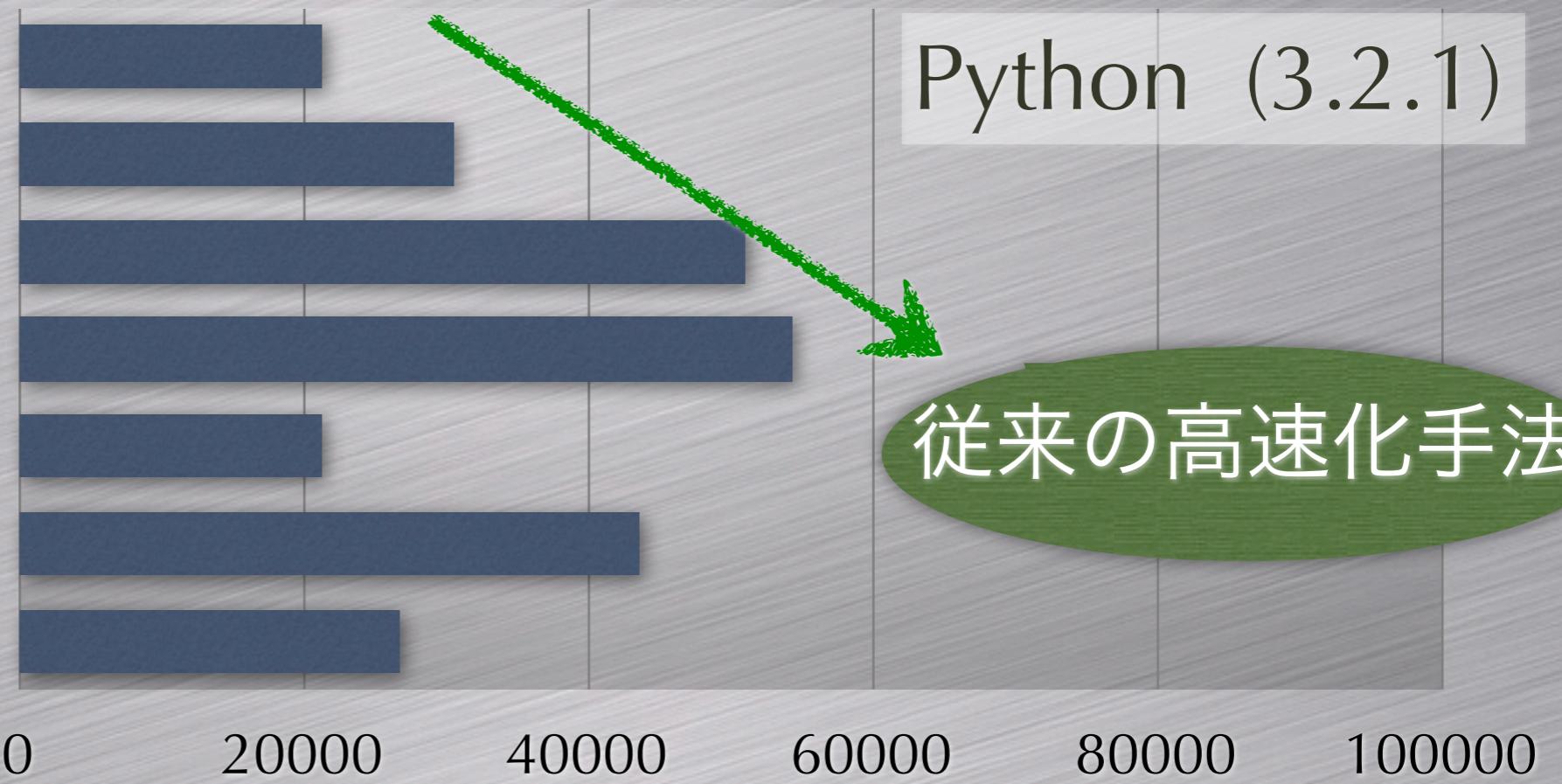
Java: StringBuilder + toString()

StringBuilderから
Stringへのコピーが発生

Java: StringBuilder w/o toString()

toString()を呼び出さ
なければC++に肉薄

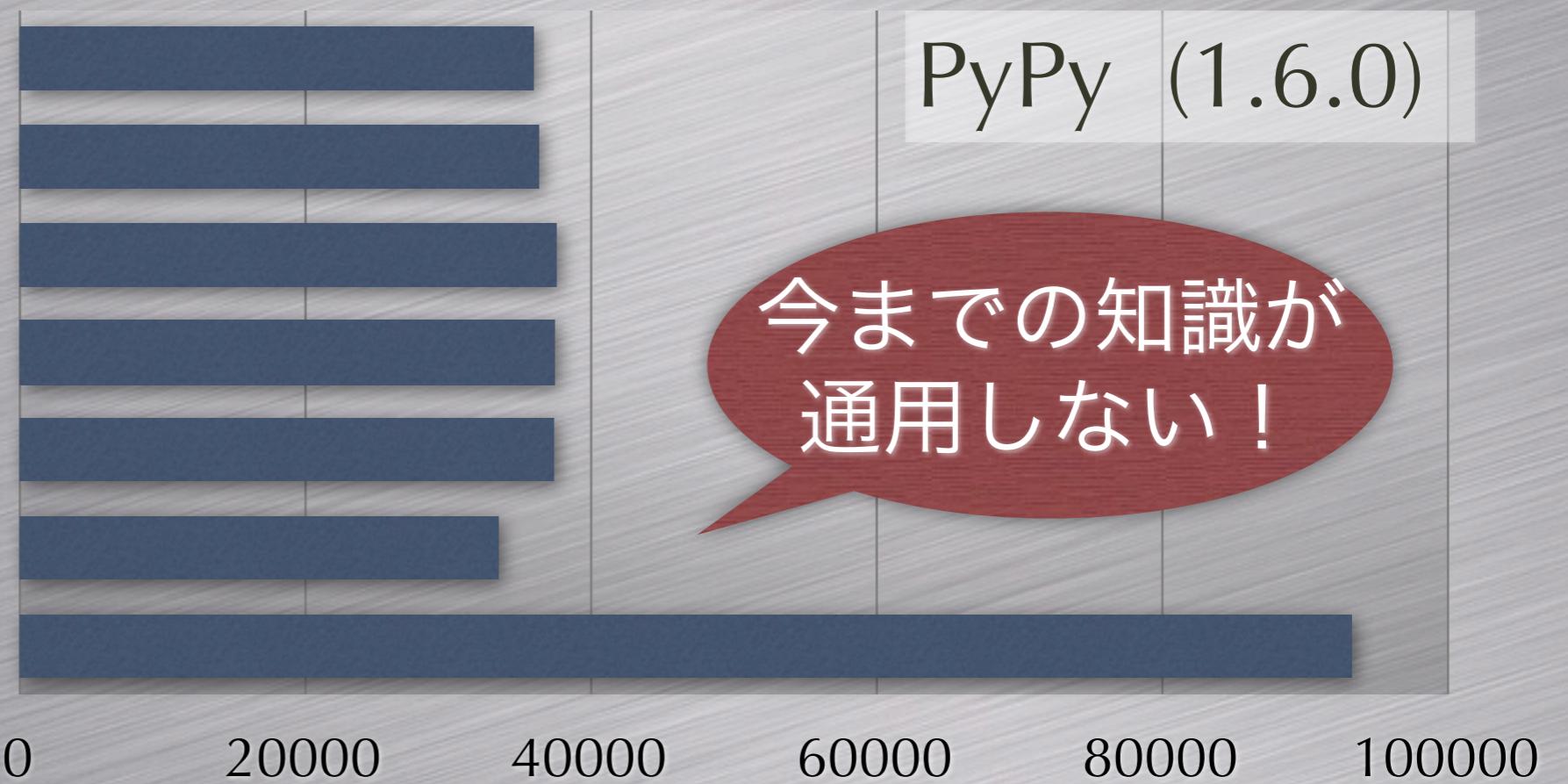
list.append()
append()
list.extend()
extend()
extend() + str()
buf[-1:] = ()
StringIO.write()



Python (3.2.1)

従来の高速化手法

list.append()
append()
list.extend()
extend()
extend() + str()
buf[-1:] = ()
StringIO.write()



PyPy (1.6.0)

今までの知識が
通用しない！

まとめ

- V8 (+ crankshaft) 速すぎワロタ
- 専用の演算子は速い、
メソッド呼び出しあはわりと遅い
- StringBuilder#toString() は重い
- JITの時代では従来の知識が通用しない

おしまい