

第十六届 D2 前端技术论坛

V8 JS AOT化的探索与实践

喻世江

Contents

目录

- 01 背景
- 02 技术选型
- 03 方案设计
- 04 优化效果
- 05 展望



D2 前端技术论坛
D2 FRONTEND TECHNOLOGY FORUM

精心

01

背景

U4内核V8引擎

快

- ✓ Disk Code Cache
- ✓ UC LLVM Compiler

强

- ✓ JS 卡死检测
- ✓ OOM 定位信息
- ✓ JS API 扩展
- ✓ JSI

稳

- ✓ 疑难崩溃攻克
- ✓ 安全漏洞修复

JavaScript



- 运行时在线编译
- 每次重新编译

Native



- 打包时 PC 离线编译
- 直接运行

跨平台

动态化

性能

JS AOT

AOT : Ahead of Time , 提前编译。

目标

让 JavaScript 具备动态化特性的同时，运行性能也可能与 Native 对标，尤其是首次启动。



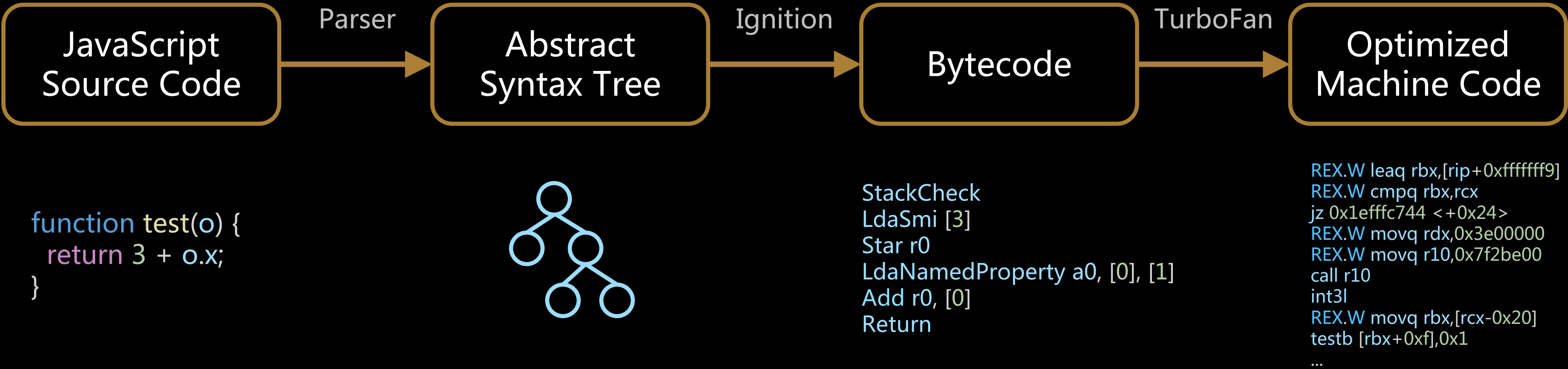
D2 前端技术论坛
D2 FRONTEND TECHNOLOGY FORUM

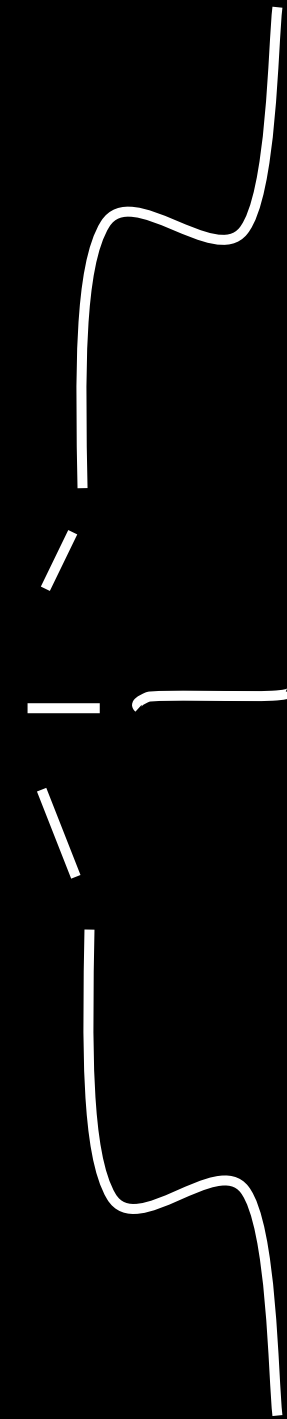
精心

02

技术选型

V8 运行 JavaScript 的流水线





本地代码

(Machine Code , 汇编)

全字节码缓存

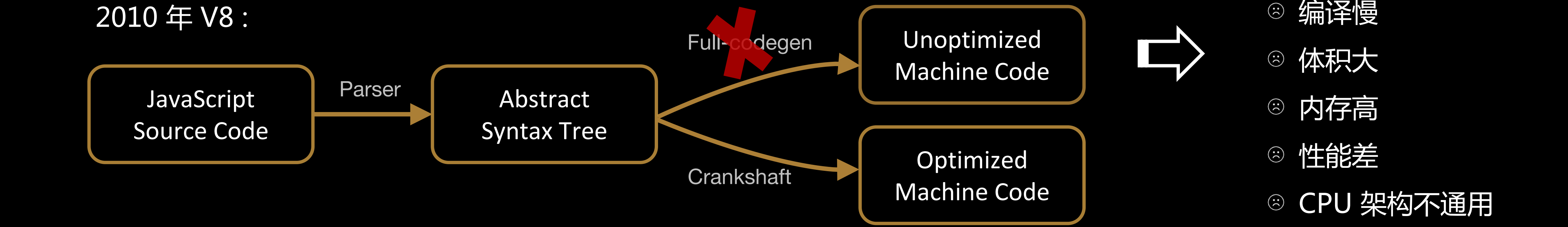
(Full Code Cache)

部分字节码缓存

(Code Cache)

调研一：使用本地代码（Machine Code，汇编）

2010 年 V8：



12.7.3 The **addition** operator (+)

NOTE The **addition** operator concatenates or numeric **addition**.

12.7.3.1 Runtime Semantics: Evaluation

AdditiveExpression : AdditiveExpression + MultiplicativeExpression

- Let *ref* be the result of evaluating *AdditiveExpression*.
- Let *real* be GetValue(*ref*).
- ReturnIfAbrupt(*real*).
- Let *rref* be the result of evaluating *MultiplicativeExpression*.
- Let *rreal* be GetValue(*rref*).
- ReturnIfAbrupt(*rreal*).
- Let *lprim* be ToPrimitive(*real*).
- ReturnIfAbrupt(*lprim*).
- Let *rprim* be ToPrimitive(*rreal*).
- ReturnIfAbrupt(*rprim*).
- If Type(*lprim*) is String or Type(*rprim*) is String, then
 - Let *lstr* be ToString(*lprim*).
 - Return ToString(*str*).
 - Let *str* be ToString(*rprim*).
 - Return Concatenate(*lstr*, *str*).
- Return the String that is the result of concatenating *lstr* and *rstr*.
- Let *lnum* be ToNumber(*lprim*).
- ReturnIfAbrupt(*lnum*).
- Let *rnum* be ToNumber(*rprim*).
- ReturnIfAbrupt(*rnum*).
- Return the result of applying the **addition** operation to *lnum* and *rnum*. See the Note below 12.7.5.

NOTE 1 No hint is provided in the calls to ToPrimitive in steps 7 and 9. All standard objects except the

6.2.3.1 GetValue (V)

- ReturnIfAbrupt(V).
- If Type(V) is not Reference, return V.
- Let *base* be GetBase(V).
- If IsUnresolvableReference(V), throw a ReferenceError exception.
- If IsPropertyReference(V), then
 - If HasPrimitiveBase(V) is true, then
 - Assert: In this case, *base* will never be null or undefined.
 - Let *base* be ToObject(*base*).
 - Return base.[[Get]](GetReferencedName(V), GetThisValue(V)).
- Else *base* must be an Environment Record,
 - Return base.GetBindingValue(GetReferencedName(V), IsStrictReference(V)) (see 8.1.1).

NOTE The object that may be created in step 5.a.ii is not accessible outside of the above abstract operation and the ordinary object [[Get]] internal method. An implementation might choose to avoid the actual creation of the object.

7.1.12.1 ToString (argument)

The abstract operation ToString converts argument to a value of type String according to Table 12:

Argument Type	Result
Completion Record	If <i>argument</i> is an abrupt completion, return <i>argument</i> . Otherwise return ToString(argument.[[value]]).
Undefined	Return "undefined".
Null	Return "null".
Boolean	If <i>argument</i> is true, return "true". If <i>argument</i> is false, return "false".
Number	See 7.1.12.1.
String	Return <i>argument</i> .
Symbol	Throw a TypeError exception.
Object	Apply the following steps: <ol style="list-style-type: none">Let <i>primValue</i> be ToPrimitive(<i>argument</i>, hint "string").Return ToString(<i>primValue</i>).

7.1.12.1 ToString Applied to the Number Type

The abstract operation ToString converts a Number *m* to String format as follows:

- If *m* is NaN, return the String "NaN".
- If *m* is +0 or -0, return the String "0".
- If *m* is less than zero, return the String concatenation of the String "-" and ToString(-*m*).
- If *m* is +∞, return the String "Infinity".
- Otherwise, let *n*, *k*, and *s* be integers such that $k \geq 1$, $10^{k-1} \leq s < 10^k$, the Number value for $s \times 10^{-k}$ is *m*, and *k* is as small as possible. Note that *k* is the number of digits in the decimal representation of *s*, that *s* is not divisible by 10, and that the least significant digit of *s* is not necessarily uniquely determined by these criteria.
- If $k \leq n \leq 21$, return the String consisting of the code units of the decimal representation of *s*, followed by the code unit 0x002E (FULL STOP), followed by the code units of the remaining *k-n* digits of the decimal representation of *s*.
- If $-6 < n \leq 0$, return the String consisting of the code unit 0x0030 (DIGIT ZERO), followed by the code unit 0x002E (FULL STOP), followed by -*n* occurrences of the code unit 0x0030 (DIGIT ZERO), followed by the code units of the *k* digits of the decimal representation of *s*.
- Otherwise, if *k* = 1, return the String consisting of the code unit of the single digit of *s*, followed by code unit 0x0065 (LATIN SMALL LETTER E), followed by the code unit 0x002B (PLUS SIGN) or the code unit 0x002D (HYPHEN-MINUS) according to whether *n*-1 is positive or negative, followed by the code units of the decimal representation of the integer abs(*n*-1) (with no leading zeroes).
- Return the String consisting of the code units of the most significant digit of the decimal representation of *s*, followed by code unit 0x002E (FULL STOP), followed by the code units of the remaining *k*-1 digits of the decimal representation of *s*, followed by code unit 0x0065 (LATIN SMALL LETTER E), followed by code unit 0x002B (PLUS SIGN) or the code unit 0x002D (HYPHEN-MINUS) according to whether *n*-1 is positive or negative, followed by the code units of the decimal representation of the integer abs(*n*-1) (with no leading zeroes).

NOTE 1 The following observations may be useful as guidelines for implementations, but are not

7.1.12.2 ToNumber (argument)

The abstract operation ToNumber converts argument to a value of type Number according to Table 11:

Argument Type	Result
Completion Record	If <i>argument</i> is an abrupt completion, return <i>argument</i> . Otherwise return ToNumber(argument.[[value]]).
Undefined	Return NaN.
Null	Return +0.
Boolean	Return 1 if <i>argument</i> is true. Return +0 if <i>argument</i> is false.
Number	Return <i>argument</i> (no conversion).
String	See grammar and conversion algorithm below.
Symbol	Throw a TypeError exception.
Object	Apply the following steps: <ol style="list-style-type: none">Let <i>primValue</i> be ToPrimitive(<i>argument</i>, hint "number").Return ToNumber(<i>primValue</i>).

7.1.3.1 ToNumber Applied to the String Type

ToNumber applied to Strings applies the following grammar to the input String interpreted as a sequence of UTF-16 encoded code points (6.1.4). If the grammar cannot interpret the String as an expansion of *StringNumericLiteral*, then the result of ToNumber is NaN.

NOTE 1 The terminal symbols of this grammar are all composed of Unicode BMP code points so the result will be NaN if the string contains the UTF-16 encoding of any supplementary code points or any unpaired surrogate code points.

Syntax

```
StringNumericLiteral ::  
  StrWhiteSpaceOpt  
  StrWhiteSpaceOpt StrNumericLiteral StrWhiteSpaceOpt  
  
StrWhiteSpaceOpt ::  
  Whitespace  
  LineTerminator  
  
StrNumericLiteral ::  
  StrDecimalLiteral  
  BinaryIntegerLiteral  
  OctalIntegerLiteral  
  HexIntegerLiteral  
  
StrDecimalLiteral ::  
  StrUnsignedDecimalLiteral  
  + StrUnsignedDecimalLiteral  
  - StrUnsignedDecimalLiteral  
  
StrUnsignedDecimalLiteral ::  
  Infinity  
  DecimalDigits , DecimalDigitsOpt ExponentPartOpt
```

7.1.1 ToPrimitive (input, PreferredType)

The abstract operation ToPrimitive converts argument to a value of type Number according to Table 11: Conversion occurs according to Table 9.

Input Type	Result
Completion Record	If <i>input</i> is an abrupt completion, return <i>input</i> . Otherwise return ToPrimitive(<i>input</i> , [[value]]) also passing the optional hint <i>PreferredType</i> .
Undefined	Return <i>input</i> .
Null	Return <i>input</i> .
Boolean	Return <i>input</i> .
Number	Return <i>input</i> .
String	Return <i>input</i> .
Symbol	Return <i>input</i> .
Object	Perform the steps following this table.

When Type(*input*) is Object, the following steps are taken:

- If *PreferredType* was not passed, let *hint* be "default".
- Else if *PreferredType* is hint Number, let *hint* be "number".
- Else *PreferredType* is hint String, let *hint* be "string".
- Let *excoticPrim* be GetMethod(*input*, @@toPrimitive).
- ReturnIfAbrupt(*excoticPrim*).
- If *excoticPrim* is not null, then
 - Let *result* be Call(*excoticPrim*, *input*, *hint*).
 - ReturnIfAbrupt(*result*).
 - If Type(*result*) is not Object, return *result*.
 - Throw a TypeError exception.
- If *hint* is "default", let *hint* be "number".
- Return OrdinaryToPrimitive(*input*, *hint*).

When the abstract operation OrdinaryToPrimitive is called with arguments *O* and *hint*, the following steps are taken:

- Assert: Type(*O*) is Object
- Assert: Type(*hint*) is String and its value is either "string" or "number".

7.1.12 Call (F, V, argumentsList)

The abstract operation Call is used to invoke the [[Call]] method of a function object. The operation is called with arguments *F*, *V*, and *argumentsList* where *F* is the function object, *V* is an ECMAScript language value that is the this value, and *argumentsList* is the value passed to the corresponding argument of the internal method. If *argumentsList* is not present, an empty List is used as its value. This abstract operation performs the following steps:

- ReturnIfAbrupt(F).
- If *argumentsList* was not passed, let *argumentsList* be a new empty List.
- If IsCallable(F) is false, throw a TypeError exception.
- Return F.[[Call]](V, *argumentsList*).

NOTE 1 However, objects may override the behaviour of defining a [[Call]] internal method. Of the objects defined in this specification only Date objects (see 20.3.4.45) and Symbol objects (see 19.4.3.4) override the default ToPrimitive behaviour. Date objects treat no hint as if the hint were String.

7.3.9 GetMethod (O, P)

The abstract operation GetMethod is used to get the value of a specific property of an object when the value of the property is a function object. The operation is performed using a wrapper object appropriate for the type of the value. The operation takes two arguments *O* and *P* where *O* is the object and *P* is the property name. The steps are as follows:

- Let *func* be GetValue(*O*, *P*).
- ReturnIfAbrupt(*func*).
- If *func* is not a function object, return undefined.
- Return *func*.

7.3.2 GetV (V, P)

The abstract operation GetV is used to retrieve the value of a specific property of an ECMAScript language value. If the value is not a function object, the operation is performed using a wrapper object appropriate for the type of the value. The operation takes two arguments *V* and *P* where *V* is the value and *P* is the property name. The steps are as follows:

- Assert: IsPropertyKey(*P*) is true.
- Let *O* be ToObject(*V*).
- Return O.[[Get]](*P*).

7.1.13 ToObject (argument)

The abstract operation ToObject converts argument to a value of type Object according to Table 13:

Argument Type	Result
Completion Record	ToObject(argument.[[value]]).
Undefined	Throw a TypeError exception.
Null	Throw a TypeError exception.
Boolean	Return a new Boolean object whose [[BooleanData]] internal slot is set to the value of <i>argument</i> . See 19.3 for a description of Boolean objects.
Number	Return a new Number object whose [[NumberData]] internal slot is set to the value of <i>argument</i> . See 20.1 for a description of Number objects.
String	Return a new String object whose [[StringData]] internal slot is set to the value of <i>argument</i> . See 21.1 for a description of String objects.
Symbol	Return a new Symbol object whose [[SymbolData]] internal slot is set to the value of <i>argument</i> . See 19.4 for a description of Symbol objects.
Object	Return <i>argument</i> .

- 编译慢
- 体积大
- 内存高
- 性能差
- CPU 架构不通用

调研二：缓存全部字节码（Full Code Cache）

- JS 函数运行覆盖度低（43%）

$$= \frac{\text{运行函数个数}}{\text{总函数个数}}$$

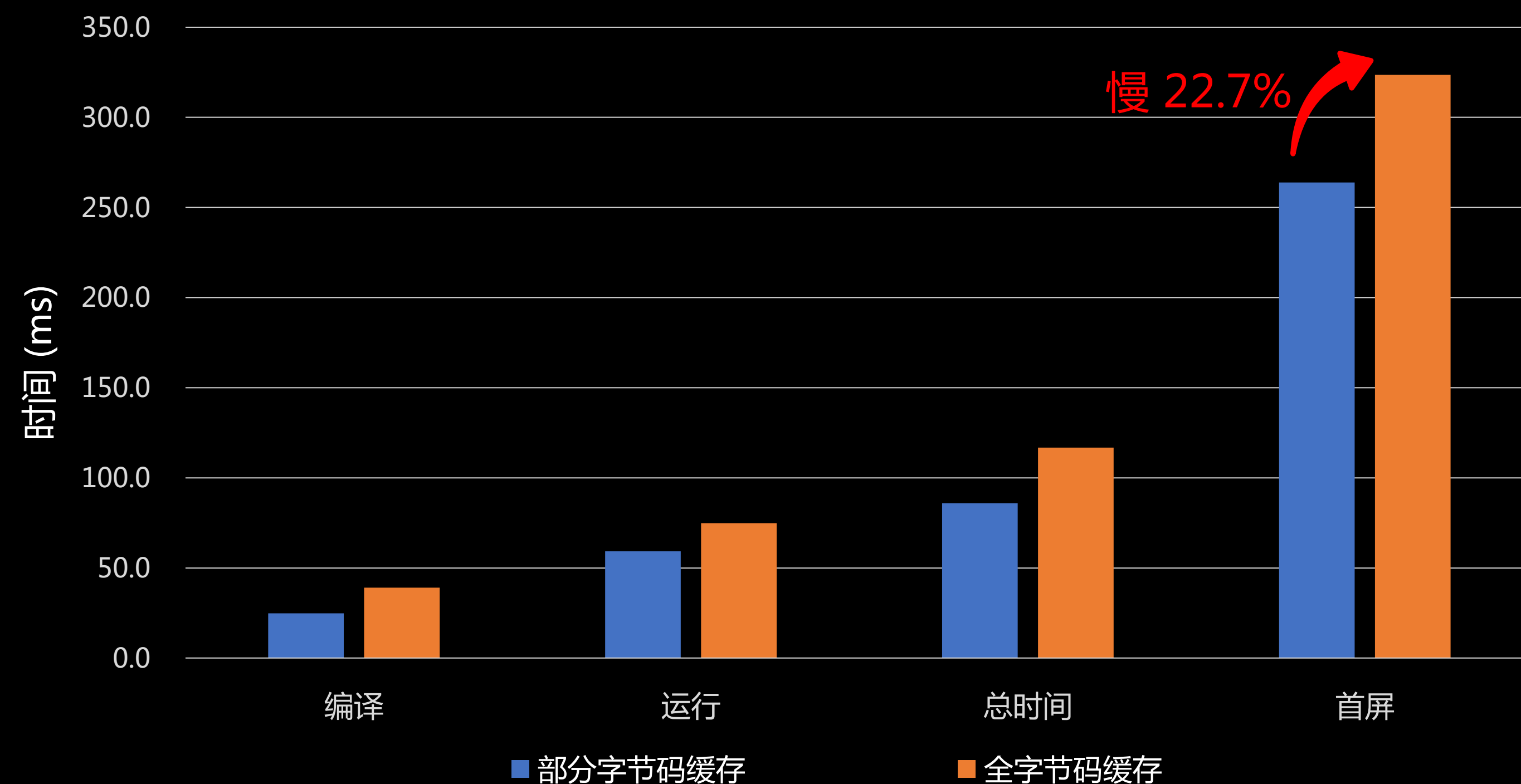
- 代码膨胀严重（2.6 x）

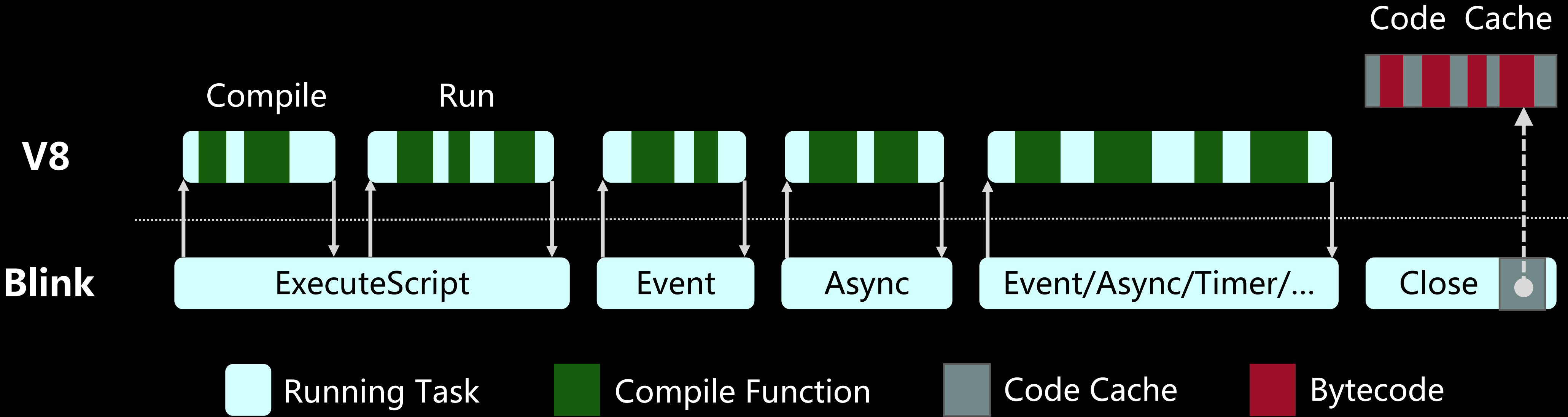
$$= \frac{\text{全字节码大小}}{\text{JS源码大小}}$$

- 加载 & 反序列化消耗大

全字节码缓存 VS. 部分字节码缓存

（越小越好）





- ⊖ V8 (U4 内核) 版本的碎片化
- ⊖ CPU 架构通用性
- ⊖ 机型通用性

面临的主要难题

有效性

兼容性



D2 前端技术论坛
D2 FRONTEND TECHNOLOGY FORUM

精心

03

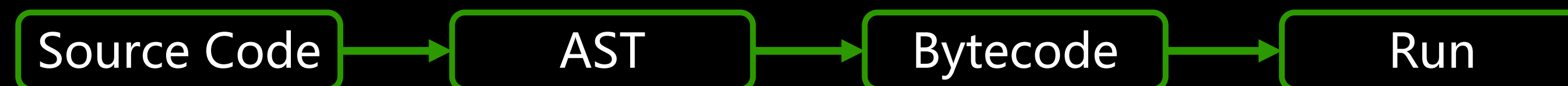
方案设计

保证 AOT 的
有效性

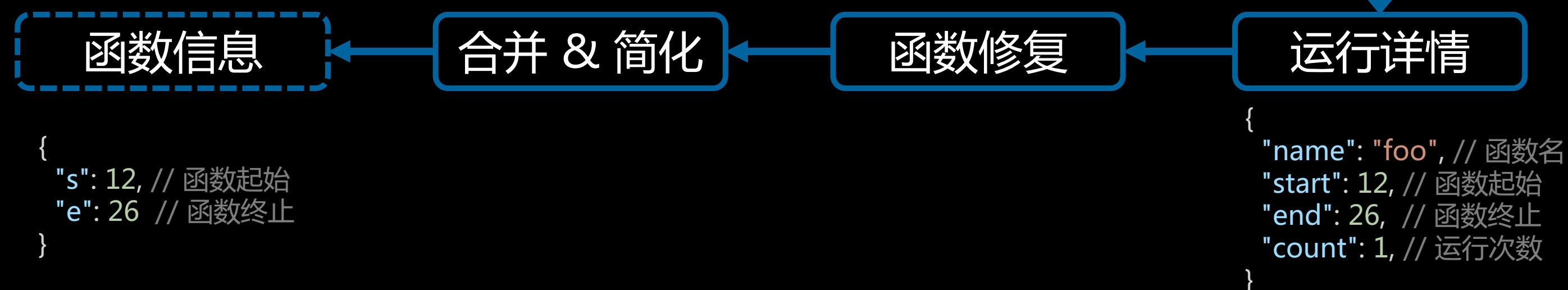
策略一：追求极致的性能 —— PGO（Profile-guided optimization）

仅为需要被执行的函数生成代码。

函数信息收集
(移动端预发布)



函数信息处理
(服务端)



AOT 生成



策略二：追求便捷的使用 —— 先验规则

预测需要被执行的函数。

```
function foo() { /* ... */ }
function bar() {
  function inner() {
    return (x, y) => {
      return x + y;
    }
  }
  return inner();
}
var Test = {
  "a": function() { /* ... */ },
  "b": function() {
    return () => {
      return "hello";
    }
  },
}
```

特征

嵌套越深，使用率越低

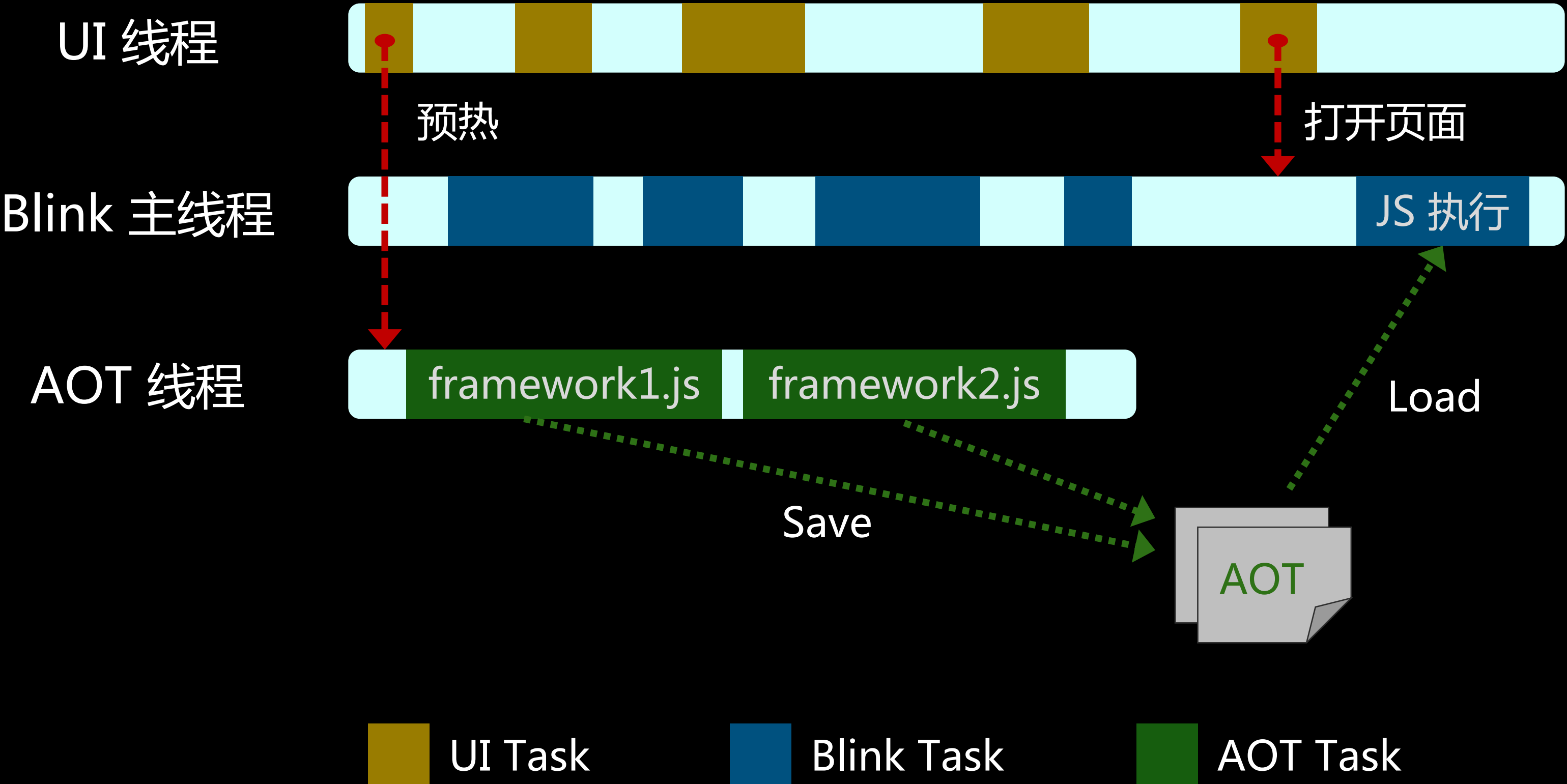
JS 越小，运行覆盖度越高

策略

- 小型 JS：不生成
- 中型 JS：全字节码缓存（覆盖度 > 80%）
- 大型 JS：Top 3 层
- 运行后增量更新

保证 AOT 的
兼容性

策略一：在线生成（空闲时预热）



生成时机：

- APP 空闲时
- 后台线程

影响或不足：

- 资源浪费（磁盘 & CPU）

适用场景：

- 框架 JS，不经常变动

策略一：在线生成（访问时生成）



生成时机：

- 页面打开时
- 后台线程

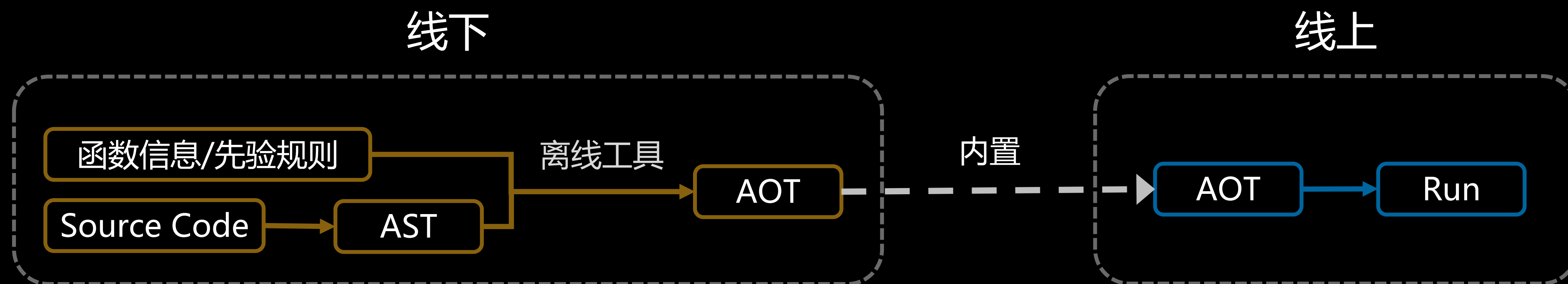
影响或不足：

- 可能来不及

适用场景：

- 业务 JS，已经离线到本地

策略二：离线生成



适用场景：

- APP冷启动时执行（没有预热时机）
- 不经常变动
- 不需要动态更新

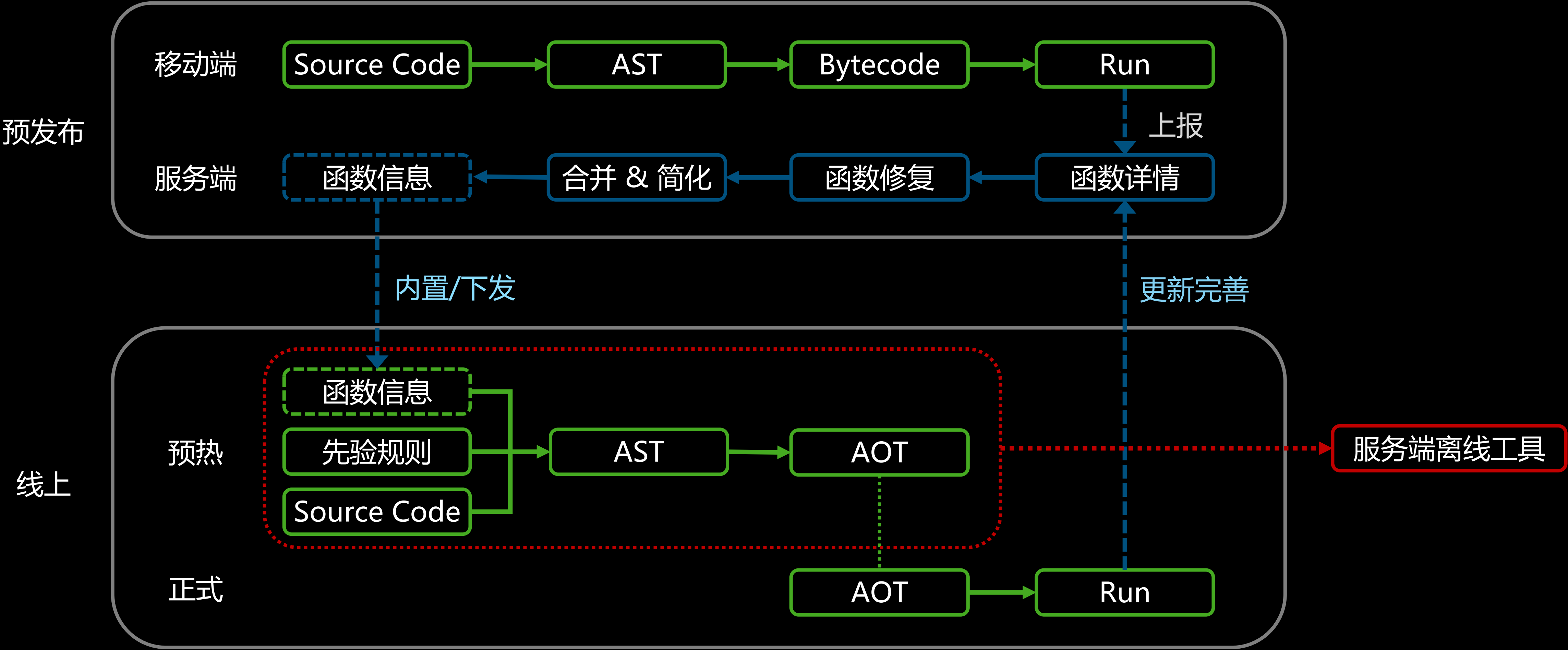
影响或不足：

- 更新 U4 内核后，可能需同步更新 AOT



丢弃 JS 源码？

- Full Code Cache
- Function.prototype.toString()

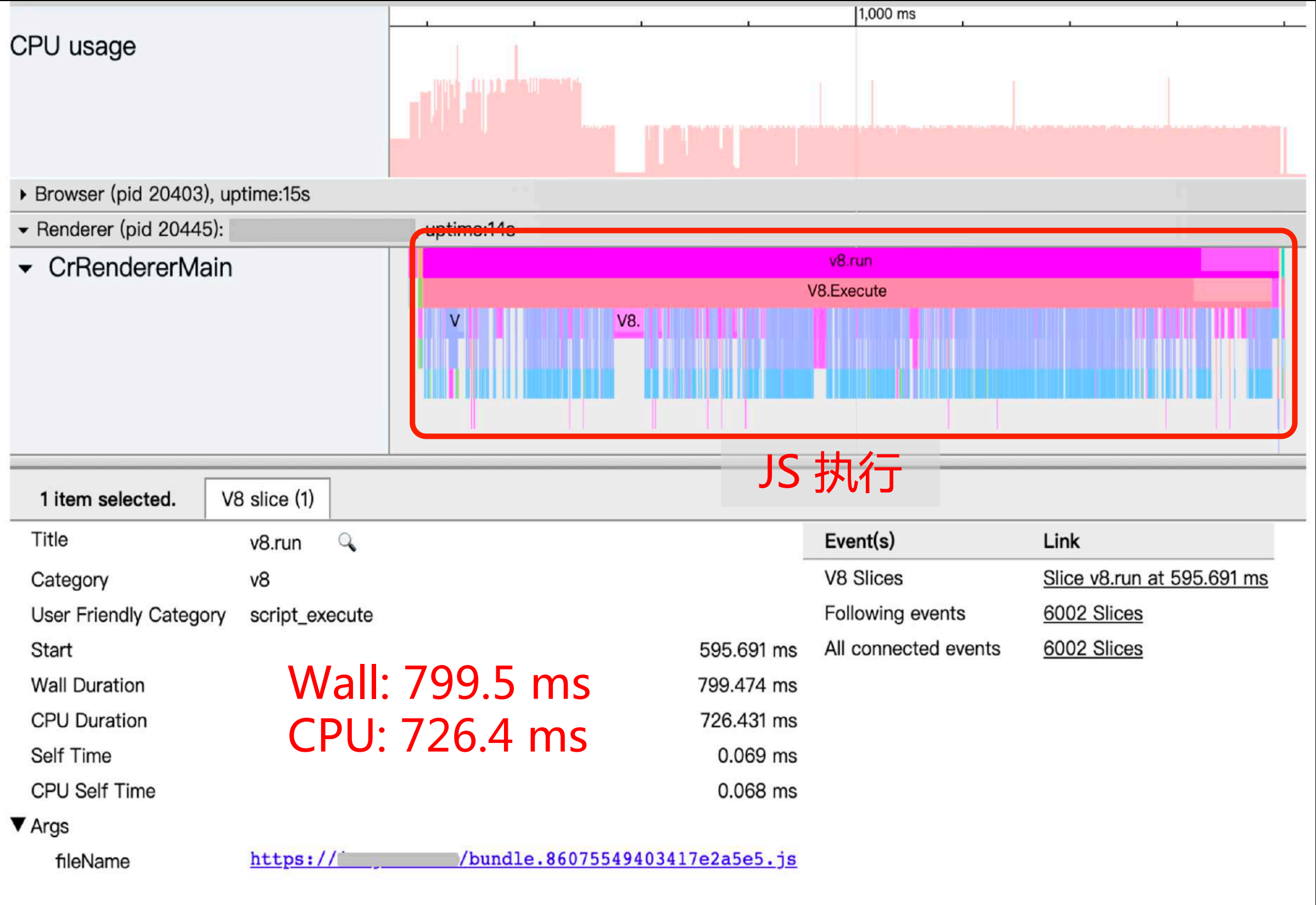




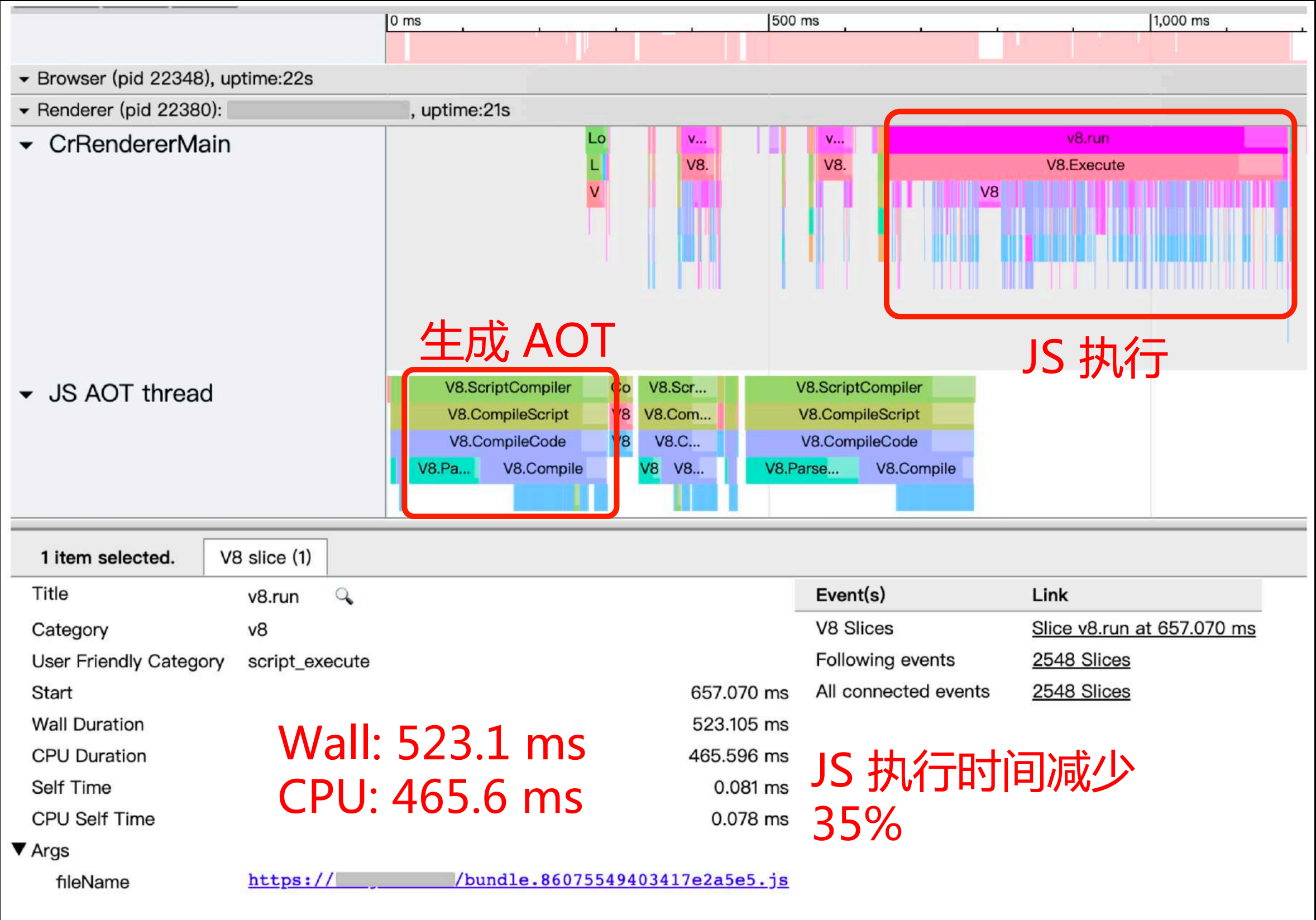
04

优化效果

正常访问（无 AOT）



打开时预热生成 AOT（先验规则）



PGO

49.0% ↑

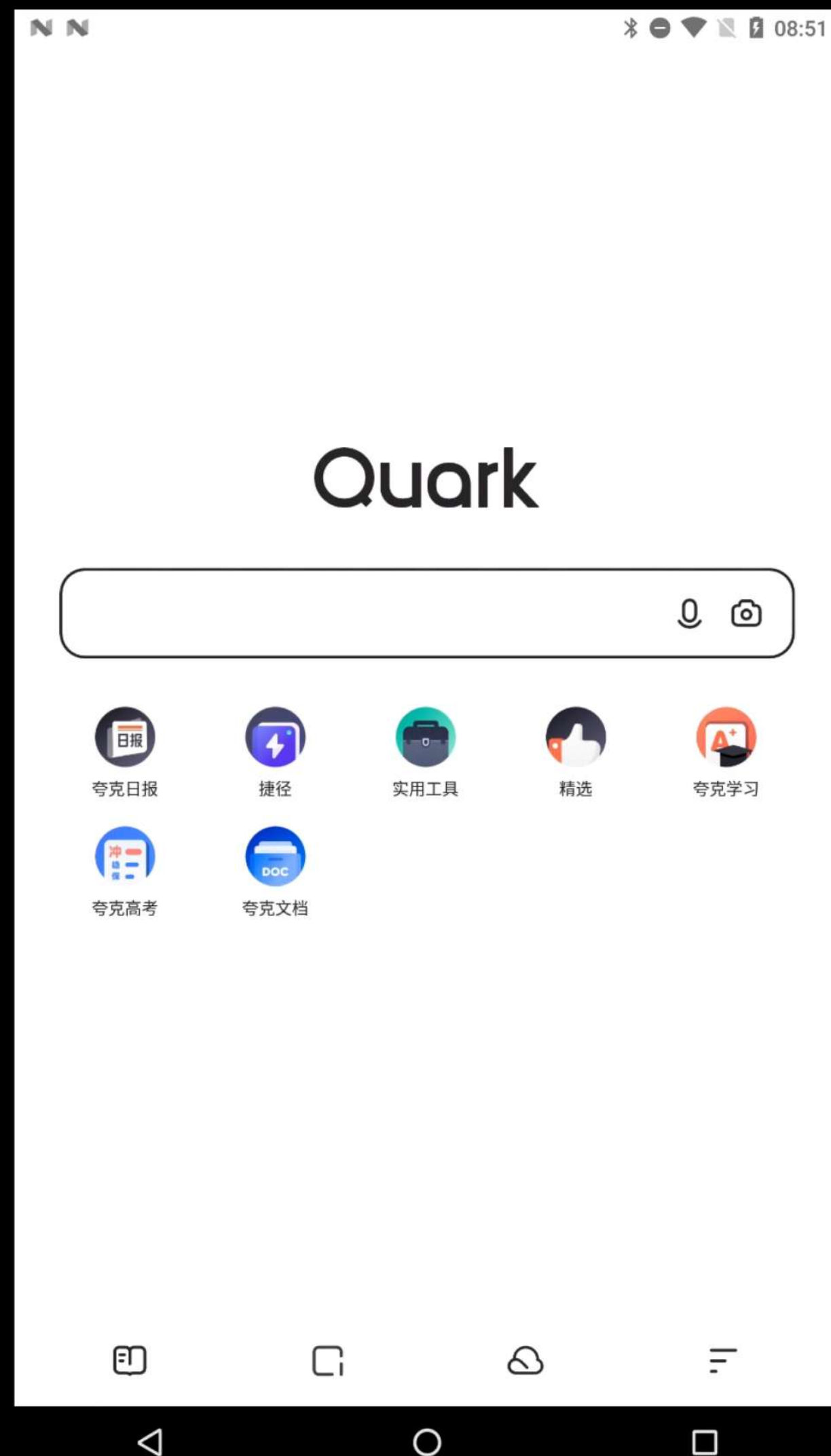
先验规则

33.9% ↑

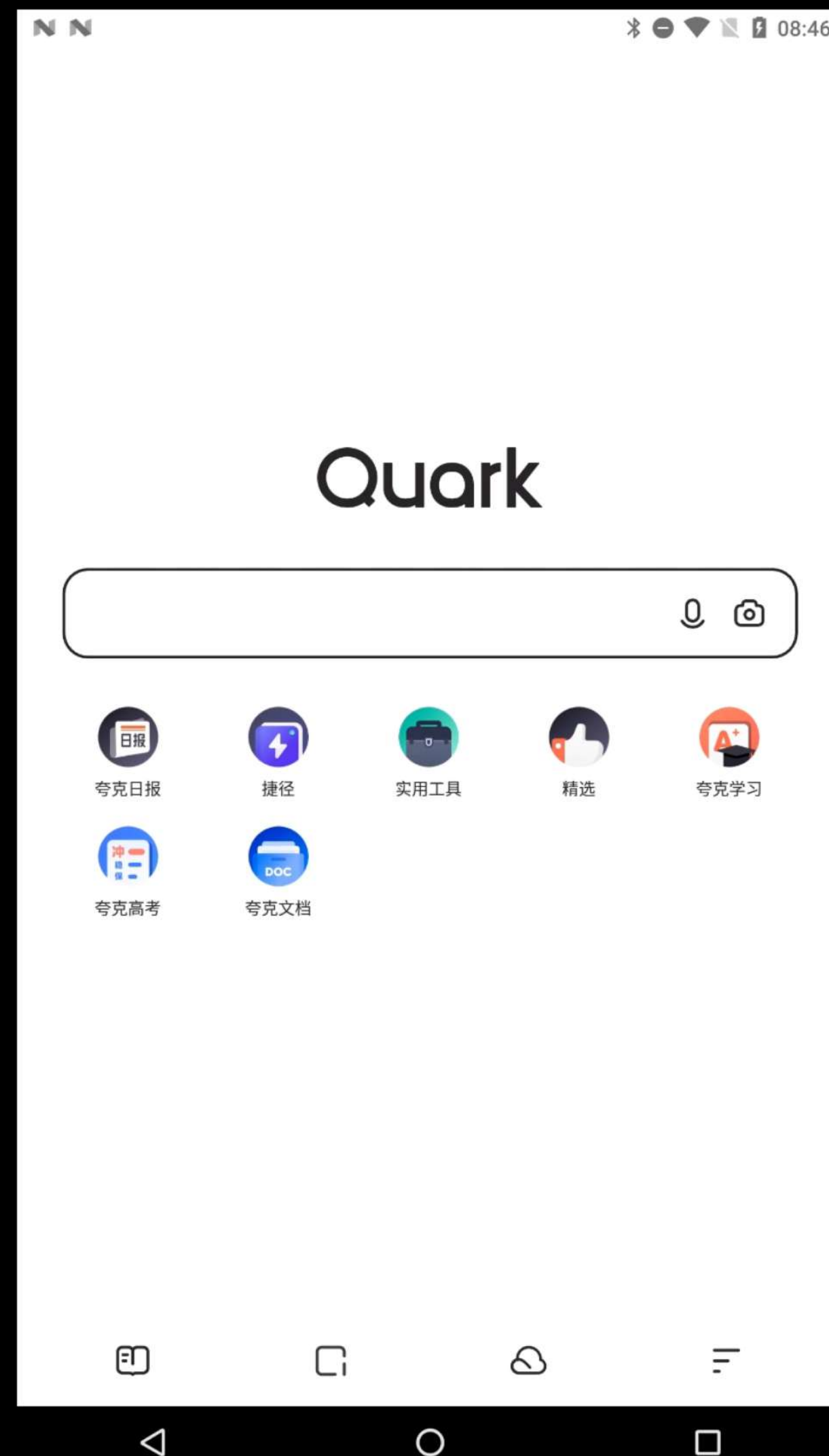
(TOP 30+ 站点平均)

夸克高考首屏性能提升 17.6%

无 AOT



使用 AOT





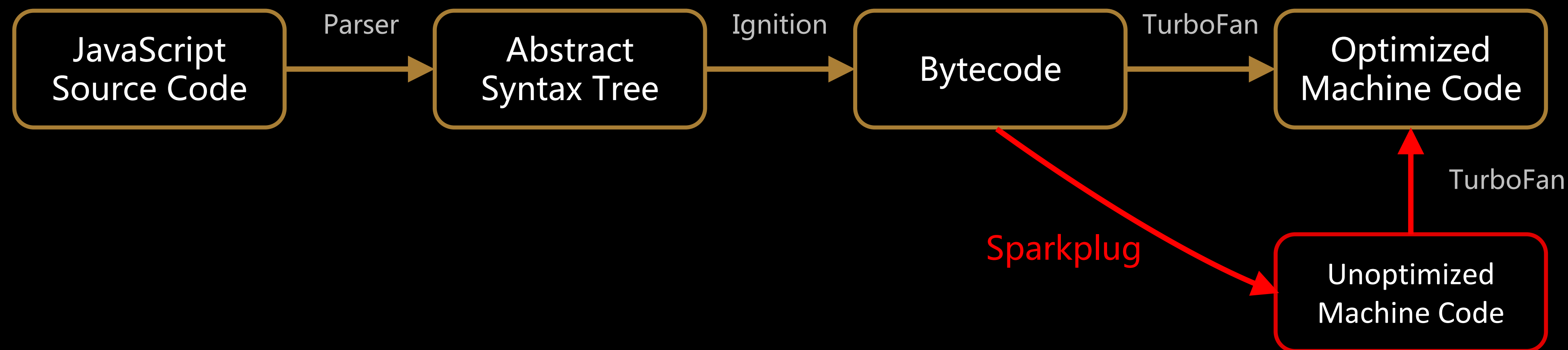
D2 前端技术论坛
D2 FRONTEND TECHNOLOGY FORUM

精心

05

展望

U4 3.0 & 4.0 ➡ U4 5.0



JS 性能 ~20% ↑

Thanks



“U4 内核技术” 公众号