



开放原子开源基金会 OpenHarmony开发者大会 2023

OpenHarmony TypeScript语言编译和运行优化



吴哲锋

OpenHarmony方舟编译器架构师

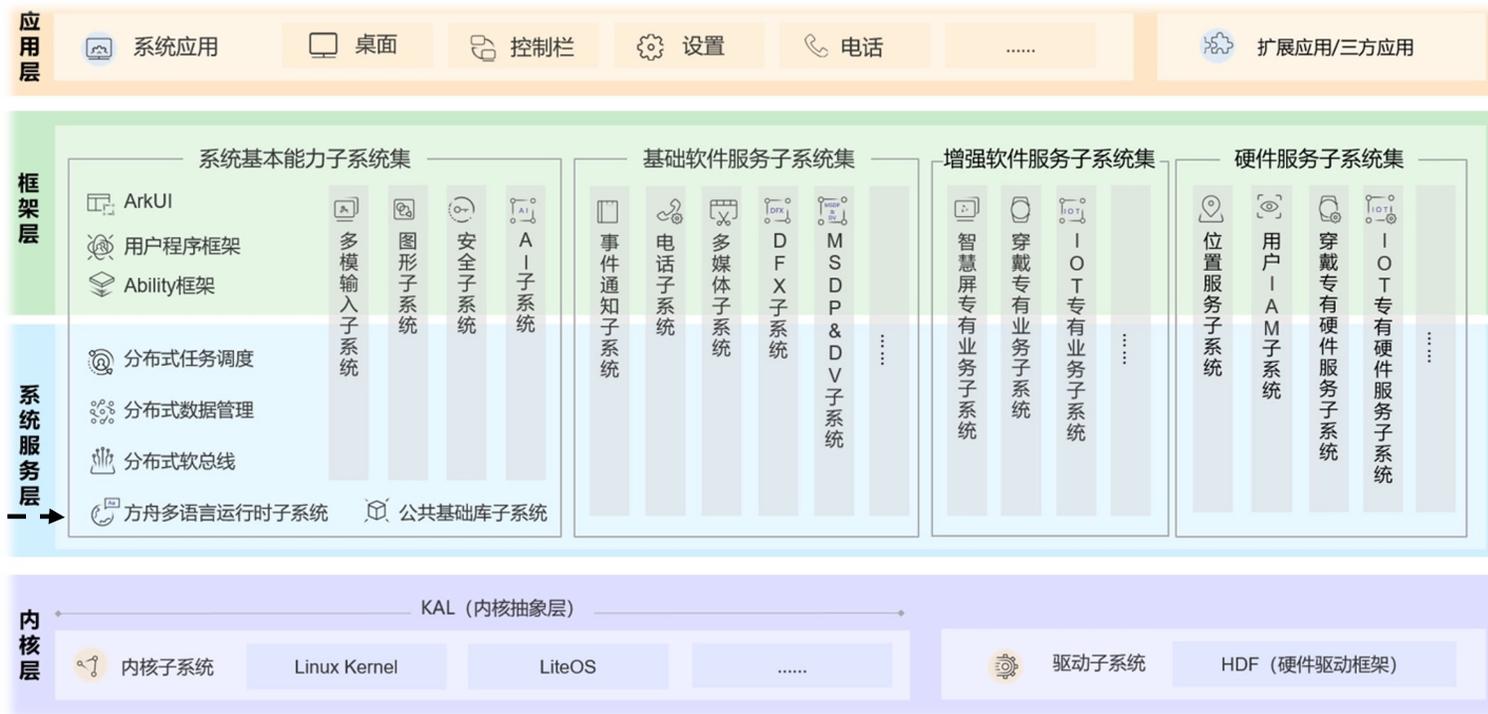
目录

01 OpenHarmony方舟编译器介绍

02 TypeScript语言编译和运行优化

03 未来路线图

OpenHarmony方舟编译器介绍



OpenHarmony架构

- 方舟编译器ArkTS运行时是OpenHarmony上默认的语言运行时，支持TypeScript-4.2和JavaScript-2021语言规范
- 提供高效container容器库，完备的C++交互ArkTS能力、高性能的解释器、静态编译器和垃圾回收器等核心组件
- 作为应用执行底座，驱动着万物互联时代的OpenHarmony应用程序

OpenHarmony方舟编译器介绍

gitee 开源软件 企业版 高校版 私有云 博客 我的

OpenHarmony / arkcompiler_ets_frontend

Watch 26 Star 31 Forked 118

代码 Issues 12 Pull Requests 23 Wiki 统计 流水线 服务

master 分支 53 标签 23

+ Pull Request + Issue 文件 Web IDE 克隆/下载

openharmony_ci	!994Update ohos-typescript's integrity by...	e5d89c0	1天前	1036次提交
es2panda	Fix runtime error when CallExpression's callee is TSNonNullExpression		13天前	
figures	Add images for readme		7个月前	
legacy_bin	Update the ts2abc of API 8 to fix the path length limit issue on the W...		29天前	
merge_abc	Correct the logic for host_out_path in ark_standalone_build		6天前	
test262	Enable abc bytecode optimizer		20天前	
testTs	Modify ts2abc type extraction for methods in class		8天前	
test_ecma_bcopt	fix code and copyright		1年前	
ts2panda	Update ohos-typescript's integrity by third_party_typescript's PR69		2天前	
.gitattributes	add ark ts2abc		2年前	
.gitignore	Descriptor: Modify for ark standalone build		5个月前	
BUILD.gn	Descriptor: Modify for ark standalone build		5个月前	
LICENSE	add ark ts2abc		2年前	
OAT.xml	Use api8 ark compiler when set compatible for api8 on api9 SDK		7个月前	
README.md	Descriptor: 部件化整改 ark -> arkcompiler		6个月前	
README_zh.md	Descriptor: 部件化整改 ark -> arkcompiler		6个月前	
bundle.json	Descriptor: 部件化整改 ark -> arkcompiler		6个月前	
ets_frontend_config.gni	Descriptor: Modify for ark standalone build		5个月前	

简介

ts2abc组件是方舟平台的一个组件，其作为方舟编译器中JavaScript语言的前端工具，支持将JavaScript文件转换为方舟字节码文件

暂无标签

JavaScript 等 6 种语言

Apache-2.0

发行版

暂无发行版

OpenHarmony

Gitee 指数 100

代码活跃度 100 (>99%)

影响力 100 (>99%)

社区活跃度 100 (>99%)

流行趋势 100 (>99%)

团队健康 100 (>99%)

贡献者 (63)

全部

近期动态

gitee 开源软件 企业版 高校版 私有云 博客 我的

OpenHarmony / arkcompiler_ets_runtime

Watching 22 Starred 76 Forked 290

代码 Issues 62 Pull Requests 34 Wiki 统计 流水线 服务 管理

master 分支 57 标签 23

+ Pull Request + Issue 文件 Web IDE 克隆/下载

openharmony_ci	!3875refactor early elimination	1f994c5	3小时前	5491次提交
.gitee	refactor stl container		1年前	
docs	doc 鸿蒙修改为OpenHarmony		13天前	
ecmascript	!3875refactor early elimination		3小时前	
script	AOT and ASM-Interpreter share builtins stub with call and new		5个月前	
test	Description: refactor early elimination		7天前	
.gitignore	Add TS Aot Test Framework		12个月前	
BUILD.gn	Support typeinfer test on es2abc		10天前	
LICENSE	add ark js_runtime		2年前	
OAT.xml	modify arkcompiler_ets_runtime path		9个月前	
README.md	Update ArkCompiler's docs issue		5个月前	
README_zh.md	modify docs		2个月前	
bundle.json	Descriptor: Modify some absolute path in arkcompiler		3个月前	
js_runtime_config.gni	Fix Log print error in intl		25天前	
libark_jsruntime.map	Re-provide Return Stack API for DFX		1个月前	

简介

方舟ArkTS运行时 (ARK TypeScript Runtime) 是OpenHarmony上ArkTS应用使用的运行时。包含ArkTS/JS对象的分配器以及垃圾回收器 (GC)、符合ECMAScript规范的标准库、用于运行ARK前端组件生成的方舟字节码 (ARK Bytecode简称abc) 的解释器、用于加速的内联缓存、静态类型编译器、运行时的C++/C函数接口 (NAPI) 等模块。

暂无标签

TypeScript 等 5 种语言

Apache-2.0

发行版

暂无发行版, 创建

OpenHarmony

Gitee 指数 100

代码活跃度 100 (>99%)

影响力 100 (>99%)

社区活跃度 100 (>99%)

流行趋势 100 (>99%)

团队健康 100 (>99%)

方舟编译器最核心的两个gitee仓库：

- https://gitee.com/openharmony/arkcompiler_ets_frontend
- https://gitee.com/openharmony/arkcompiler_ets_runtime

目录

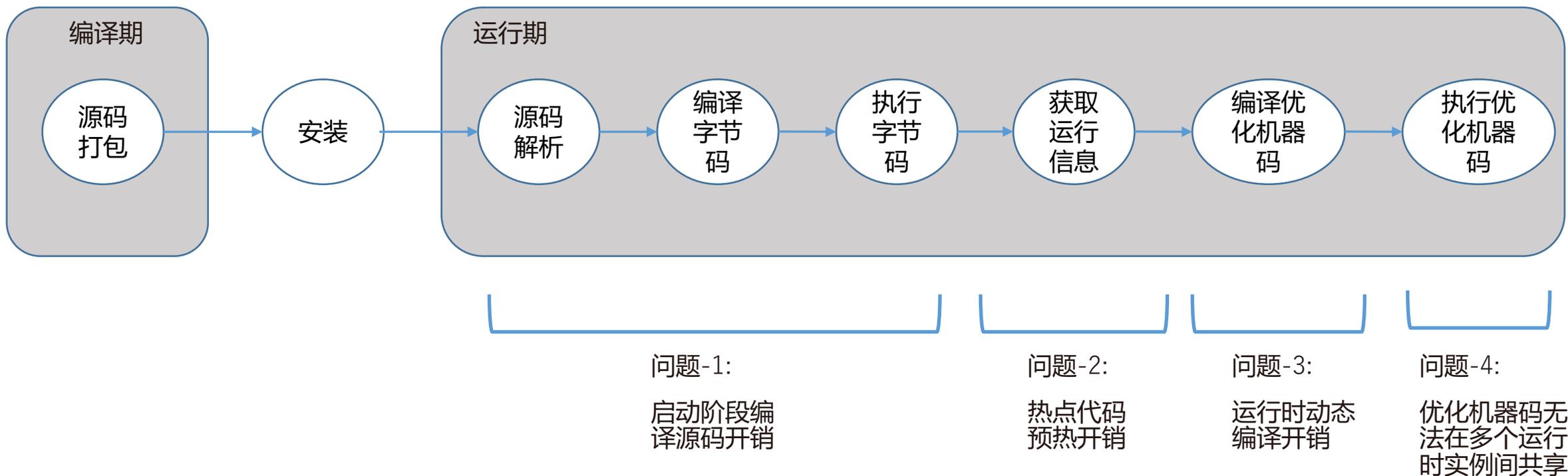
01 OpenHarmony方舟编译器介绍

02 TypeScript语言编译和运行优化

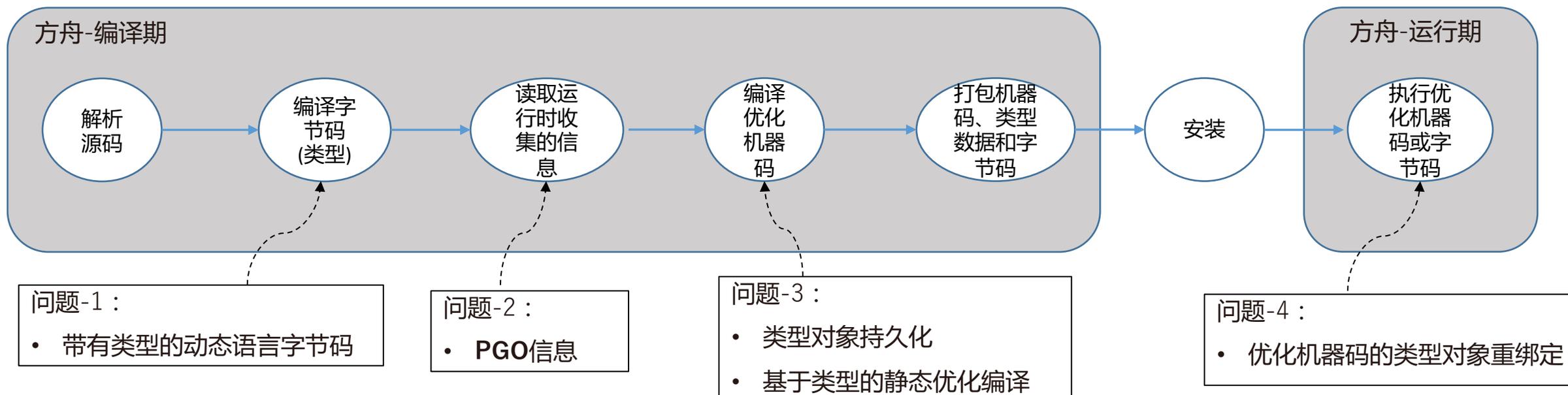
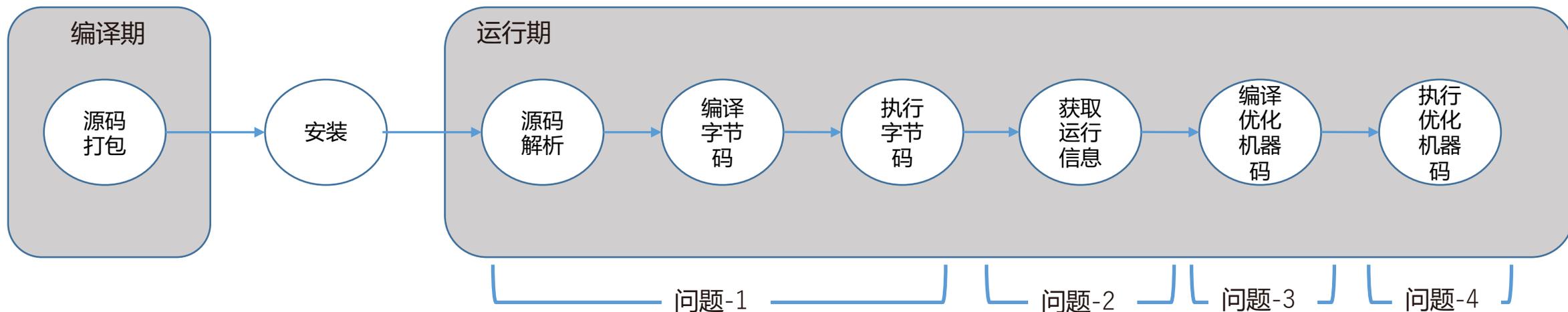
03 未来路线图

TypeScript和JavaScript语言的业界实现

尽管TypeScript和JavaScript拥有世界上最庞大的开发者社区，完善的生态系统，但是业界的语言实现上还有不足之处：



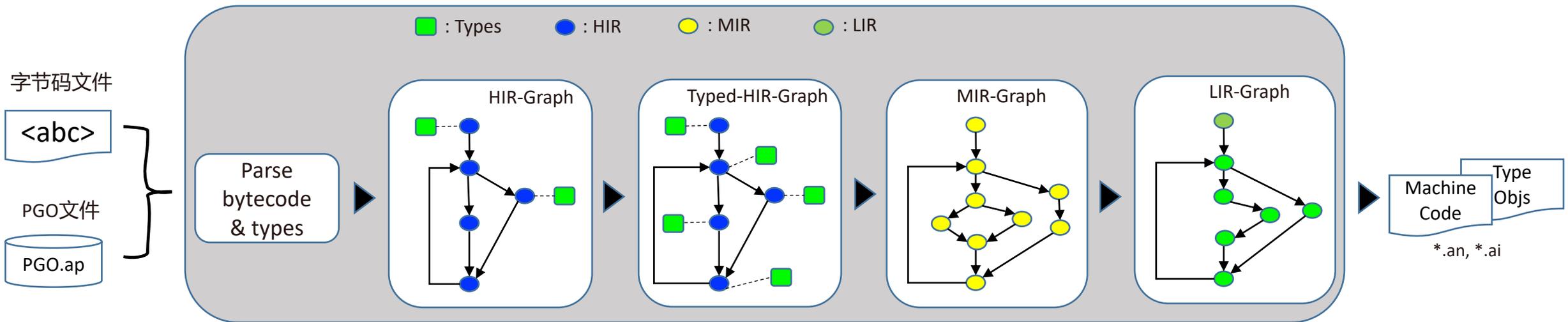
TypeScript和JavaScript语言的方舟实现



编译流水线

设计思路：利用TS类型信息和Profile信息进行静态优化编译并生成优化代码

静态优化编译



类型字节码文件

- TS类型信息
- 运行时类型信息

类型分析

- 解析字节码
- 解析类型信息
- 解析运行时信息

CFG流图构建

- 构建CFG流图
- 关联类型信息

类型推导

- 推导HIR上类型
- HIR关联类型

类型指引lowering

- HIR分解为MIR
- MIR进行必要优化

类型指引lowering

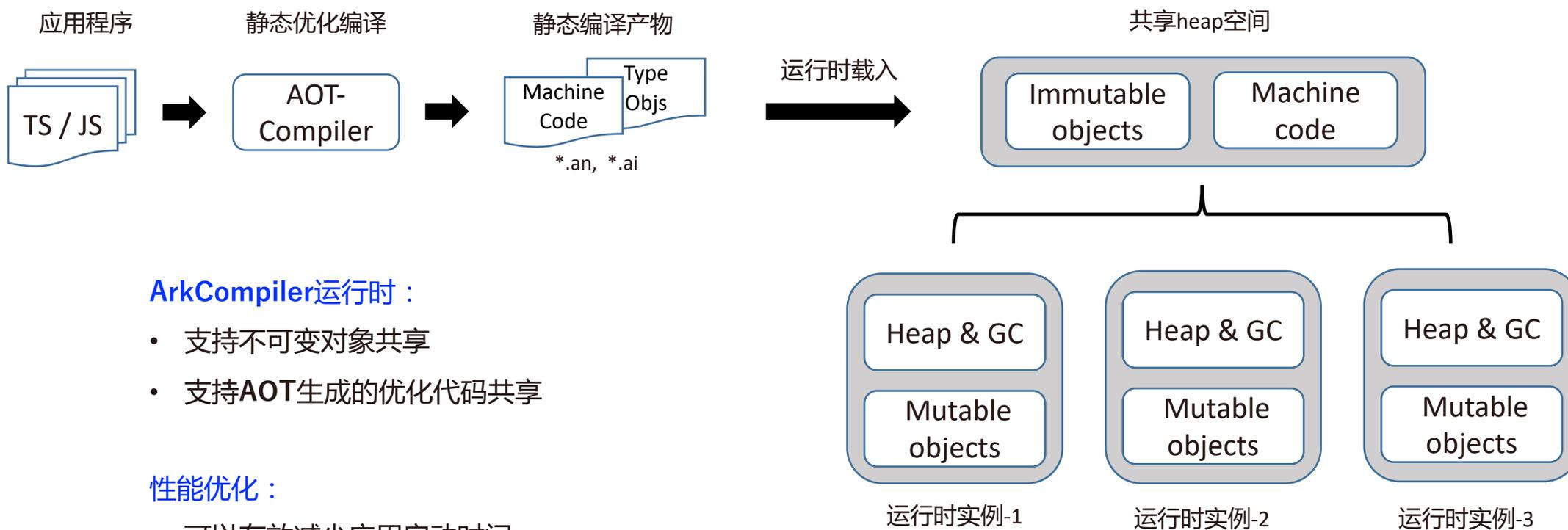
- MIR继续lowering到LIR层并继续优化和代码生成

编译结果导出

- 生成的机器码和类型表示分别保存到an和ai文件中

运行时优化

设计思路：尽量共享不可变对象（比如字节码和优化机器码）



ArkCompiler运行时：

- 支持不可变对象共享
- 支持AOT生成的优化代码共享

性能优化：

- 可以有效减少应用启动时间
- 可以有效降低应用启动内存

代码编译示例

point.ts

```
class Point {
  x: number
  y: number
  constructor(v1: number, v2: number) {
    this.x = v1
    this.y = v2
  }
}
let p = new Point(1.0, 2.0)
```

前端编译



point.abc (字节码信息)

```
.function any test.#1#Point(any a0, any a1, any a2, any a3, any a4) {
  ....
  stobjbyname 0, "x", v5
  lda v2
  sta v7
  mov v5, v7
  lda v4
  stobjbyname 2, "y", v5
  ....
  return
}

.function any test.func_main_0(any a0, any a1, any a2) {
  ....
  mov v6, v8
  defineclasswithbuffer 0, point.#1#Point, test_4, 2, v6
  ....
  newobjrange 1, 3, v5
  ....
  returnundefined
}
```

写操作字节码，x赋值为v1

写操作字节码，y赋值为v2

定义类class Point

申请类Point的对象



代码编译示例

point.ts

```
class Point {
  x: number
  y: number
  constructor(v1: number, v2: number) {
    this.x = v1
    this.y = v2
  }
}
let p = new Point(1.0, 2.0)
```

前端编译



point.abc (类型信息)

```
type2Index:
class Point: 2
constructor: 3
variable2Type:
x: 1
y: 1
v1: 1
v2: 1
p: 4
.....
```

+

变量Point类型为class

类成员变量x, y类型编码为1(表示number)

变量v1, v2类型编码为1(表示number)

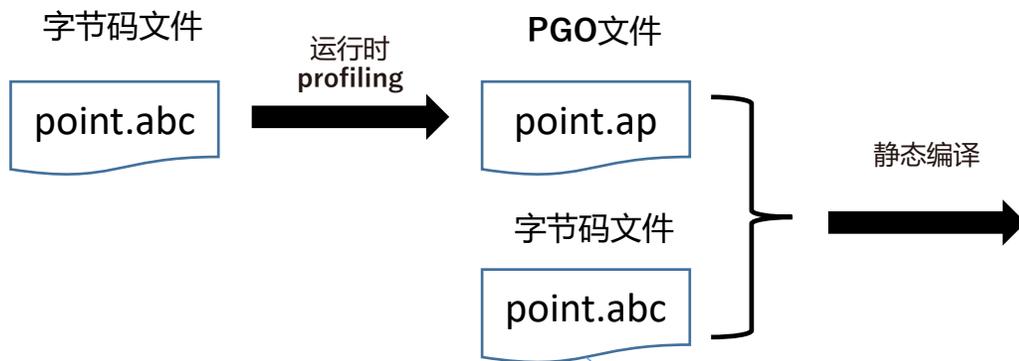
类成员变量constructor为函数

point.abc (字节码信息)

```
.function any test.#1#Point(any a0, any a1, any a2, any a3, any a4) {
  ....
  stobjbyname 0, "x", v5
  lda v2
  sta v7
  mov v5, v7
  lda v4
  stobjbyname 2, "y", v5
  xxxx
  return
}

.function any test.func_main_0(any a0, any a1, any a2) {
  ....
  mov v6, v8
  defineclasswithbuffer 0, test.#1#Point, test_4, 2, v6
  ....
  newobjrange 1, 3, v5
  ....
  returnundefined
}
```

代码编译示例



point.ts

```

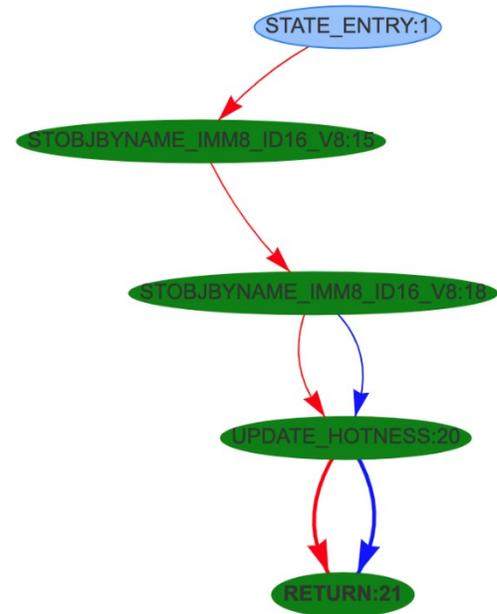
class Point {
  x: number
  y: number
  constructor(v1: number, v2: number) {
    this.x = v1
    this.y = v2
  }
}

let p = new Point(1.0, 2.0)
  
```

```

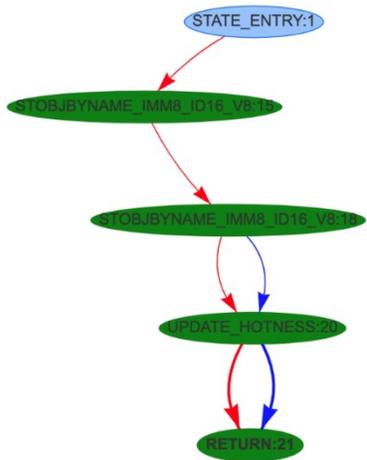
.function any test.#1#Point(any a0, any a1, any a2, ....) {
  ....
  stobjbyname 0, "x", v5
  lda v2
  sta v7
  mov v5, v7
  lda v4
  stobjbyname 2, "y", v5
  xxxx
  return
}
  
```

HIR表示的控制流图



- 从字节码翻译为编译器中间表示 (Circuit-IR)
- Circuit-IR是一种Sea-of-node类型的中间表示
- 红色蓝色表示控制流和依赖流

代码编译示例



HIR表示的控制流图

类型推导和对象布局



根据关联静态类型：

class Point: [x: number, y: number]



Hidden Class

key	offet
x	0
y	1

<HC0>



Lowering(STOBJBYNAME (obj, x))

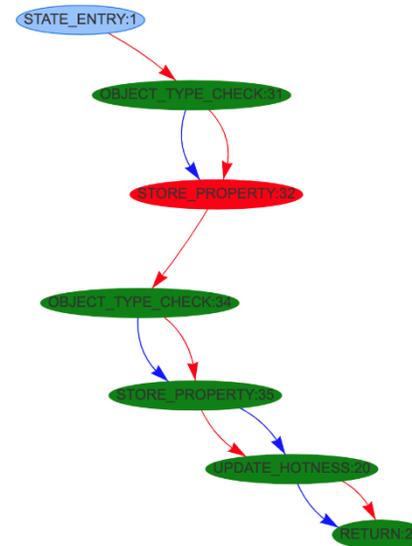
=> OBJECT_TYPE_CHECK(HC0)

STORE_PROPERTY(obj, x, 0)

Lowering(STOBJBYNAME (obj, y))

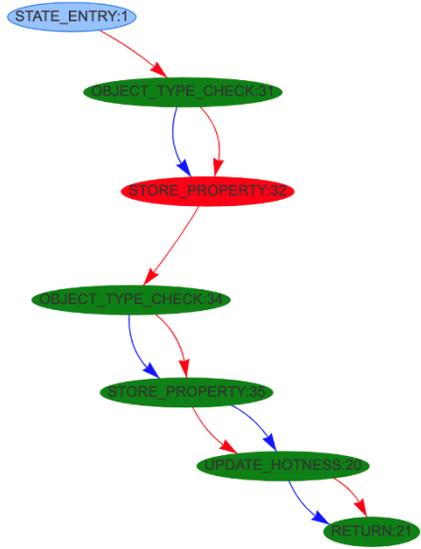
=>

根据类型信息
lowering到MIR



MIR表示的控制流图

代码编译示例

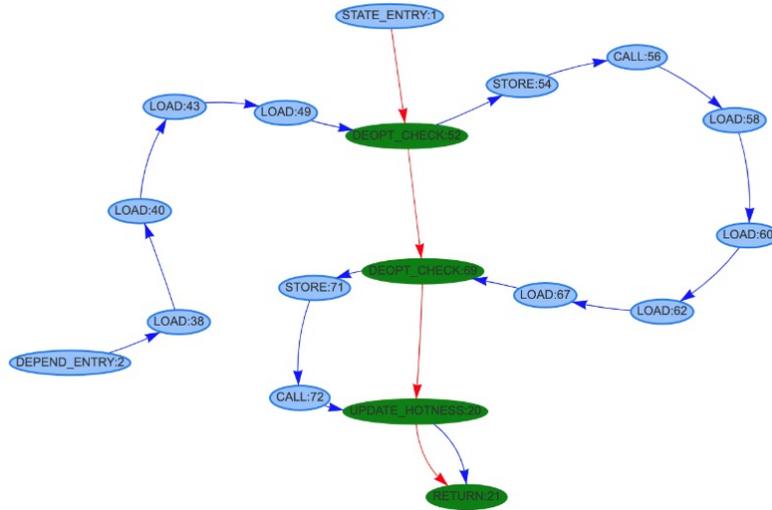


MIR表示的控制流图

生成必要类型check-bailout点和进行check删除等



基本编译优化后 lowering到LIR



LIR表示的控制流图

代码生成



```

00000350:e5894855 pushq %rbp
00000351:48e58948 movq %rsp, %rbp
00000354:10ec8348 subq $16, %rsp
00000358:56415741 pushq %r15
....
0000038a:08528b48 movq 8(%rdx), %rdx
0000038e:30423b48 cmpq 48(%rdx), %rax
00000392:00a7850f jne 0x43f
00000398:204b8948 movq %rcx, 32(%rbx)
0000039c:000020ba movl $32, %edx
....
    
```

point.an

HCO point.ai

目录

01 OpenHarmony方舟编译器介绍

02 TypeScript语言编译和运行优化

03 未来路线图

未来路线图

OpenHarmony-3.2版本

- 支持TS的Host端AOT编译器
- 利用TS类型信息进行基础的优化编译

OpenHarmony 后续版本

- 实现更多的优化编译算法
- 提供设备侧的Device-AOT-Compiler，实现基于PGO的端侧AOT编译功能
- 跨平台部署AOT编译结果的能力

THANK YOU



扫描二维码 关注官方公众号

【官网网址】 www.openharmony.cn