Ztest: A C++ Unit Testing Framework

High-level Language Programming Project

郑辰阳　叶穗华　吴泓庆　齐彦松　王瑞箐

未来技术学院
数据科学与大数据技术

2025 年 6 月 6 日

系统背景与动机

- 面向对象设计与工程实践是软件工程教育的核心
- 现有测试框架 (GTest) 存在不足：
  - 上手难度高
  - 并发支持有限
  - 报告系统简单
  - 可扩展性一般
- 我们需要一个：
  - 灵活高效的测试工具
  - 支持 GUI 界面
  - 多种测试类型
  - 详细测试报告

框架对比

表: 主流测试框架对比

框架	GUI 支持	并发测试	报告系统	扩展性	数据驱动
Google Test	无	有限	基础	中等	不支持
JUnit	Eclipse 插件	支持	HTML/XML	高	支持
PyTest	第三方工具	优秀	丰富	优秀	支持
Catch2	无	一般	简洁	中等	不支持
Ztest*	精美	优秀	丰富	高	支持

系统功能

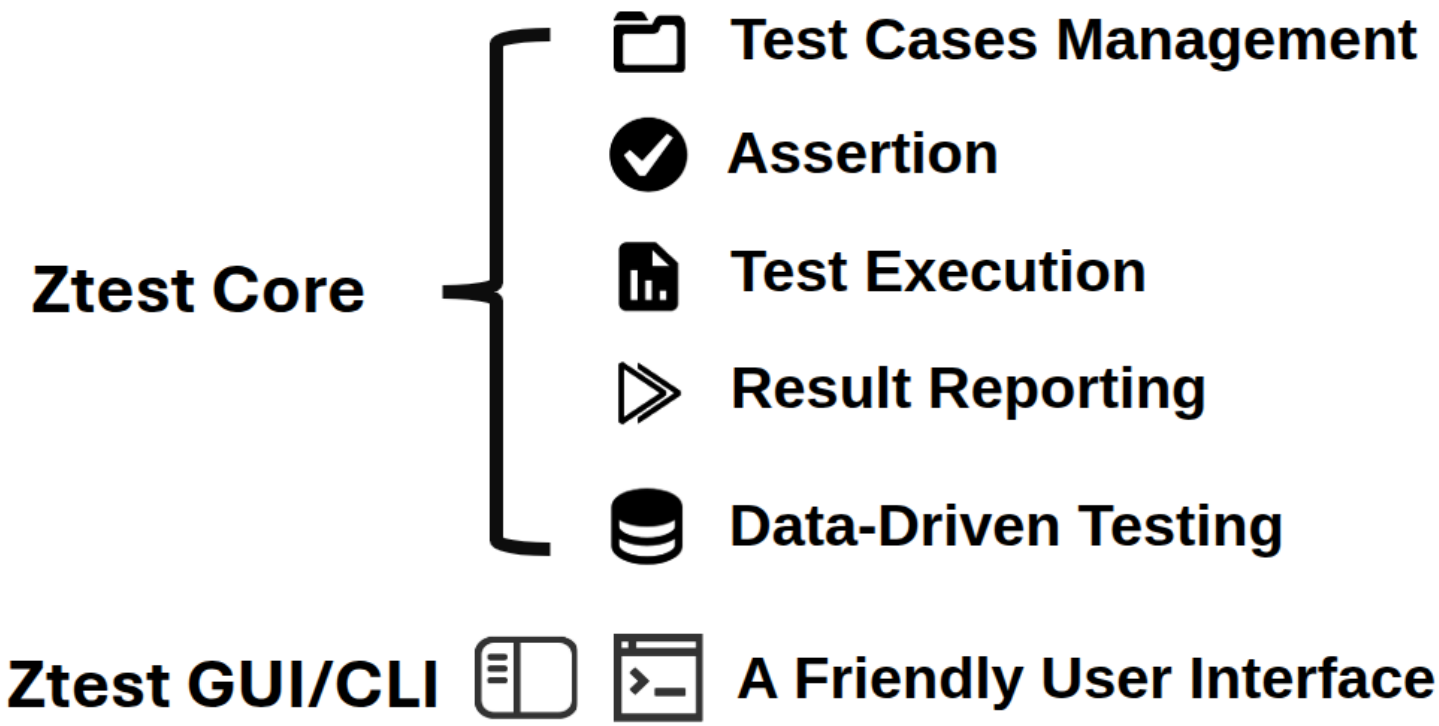


图: Ztest 功能架构

测试类型

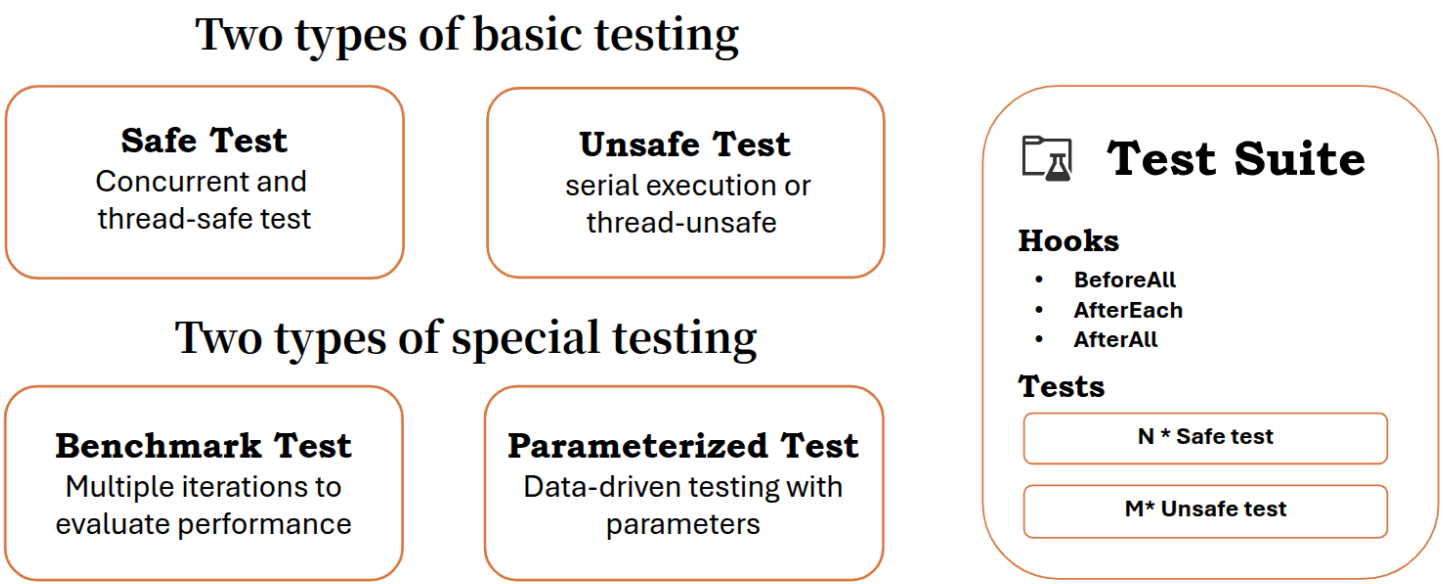


图: 支持的测试类型

核心架构

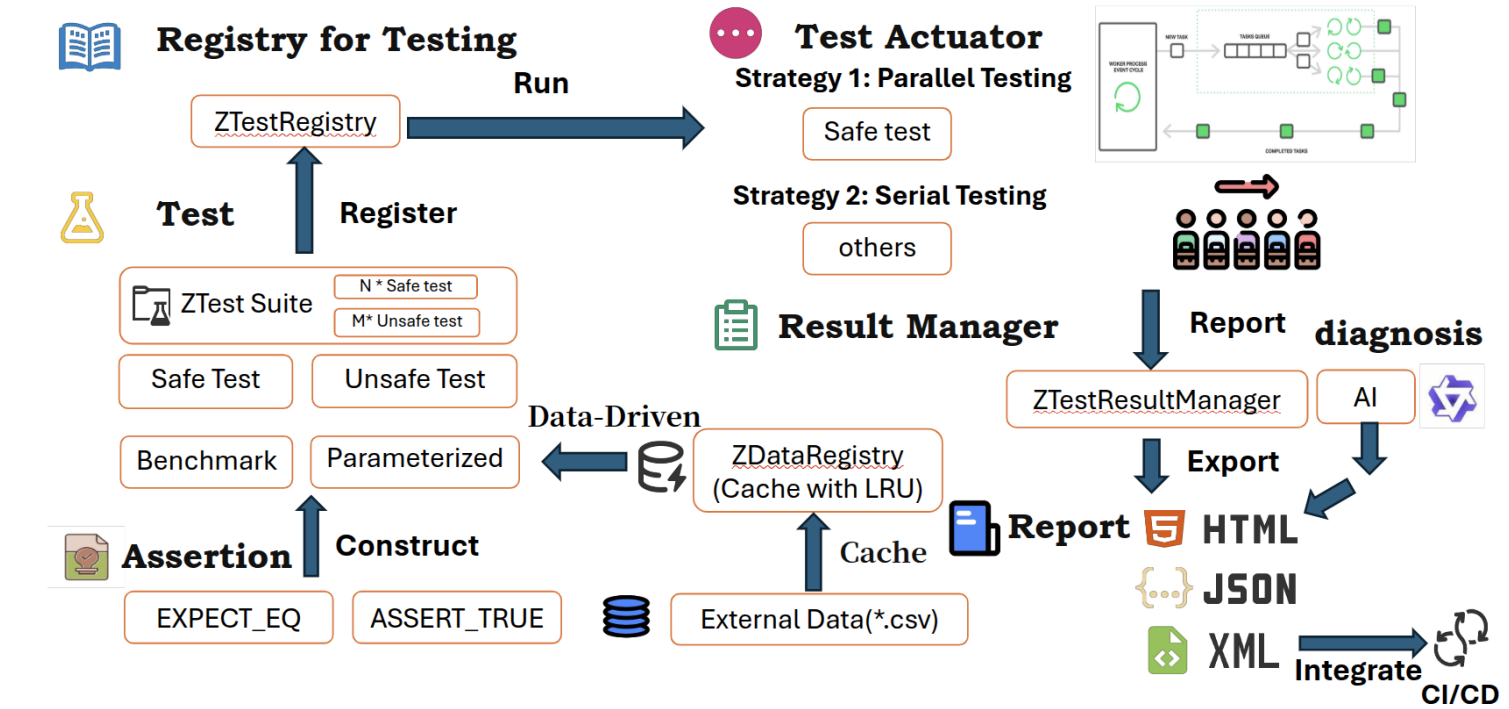


图: Ztest 核心架构

GUI 架构

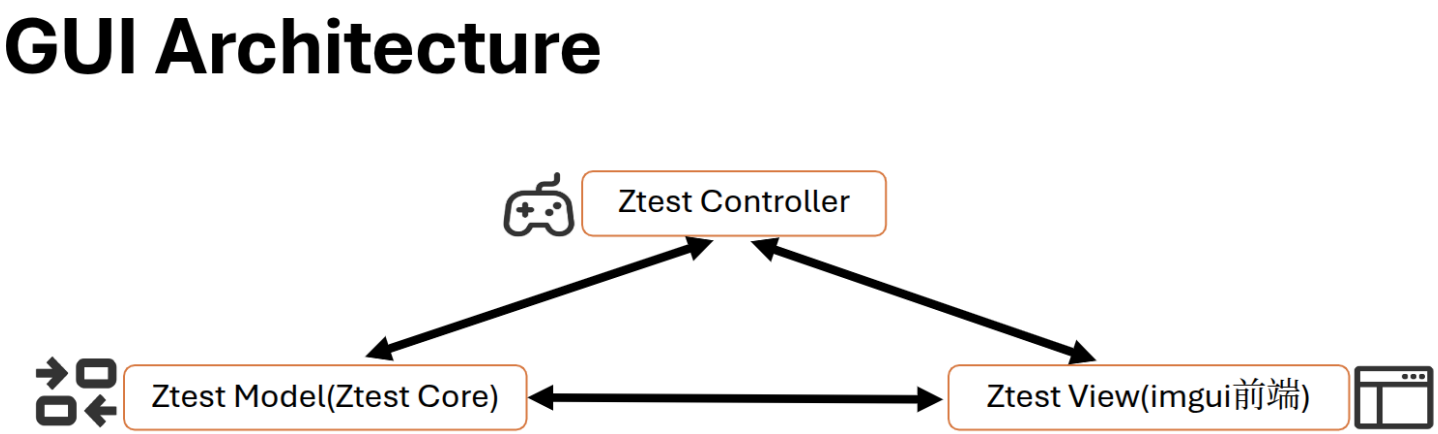


图: GUI 架构设计

类图设计

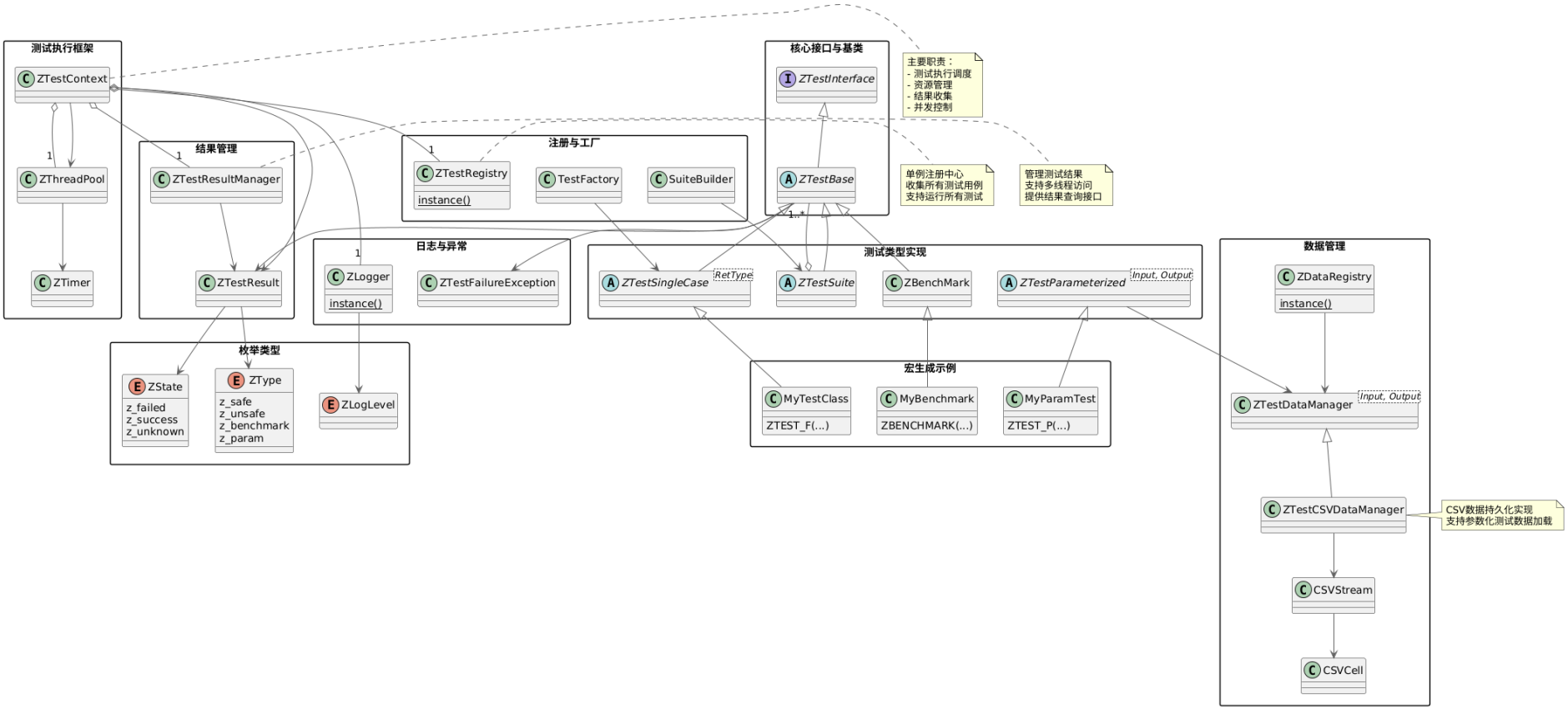

测试执行框架
ZTestContext
ZThreadPool
ZTimer
结果管理
ZTestResultManager
ZTestResult
注册与工厂
ZTestRegistry
instance()
TestFactory
SuiteBuilder
核心接口与基类
ZTestInterface
ZTestBase
主要职责：
- 测试执行调度
- 资源管理
- 结果收集
- 并发控制
单例注册中心
收集所有测试用例
支持运行所有测试
管理测试结果
支持多线程访问
提供结果查询接口
日志与异常
ZLogger
instance()
ZTestFailureException
测试类型实现
ZTestSingleCase
RetType
ZTestSuite
ZBenchMark
ZTestParameterized
Input, Output
枚举类型
ZState
z_failed
z_success
z_unknown
ZType
z_safe
z_unsafe
z_benchmark
z_param
ZLogLevel
宏生成示例
MyTestClass
ZTEST_F(...)
MyBenchmark
ZBENCHMARK(...)
MyParamTest
ZTEST_P(...)
数据管理
ZDataRegistry
instance()
ZTestDataManager
Input, Output
ZTestCSVDataManager
CSV数据持久化实现
支持参数化测试数据加载
CSVStream
CSVCell

图: 系统类图

测试定义语法

测试定义宏

[basicstyle=]
$\texttt{ZTEST}_F(SuiteName, TestName, safe/unsafe)...ZBENCHMARK(SuiteName, TestNam$

链式定义

[basicstyle=] $\texttt{auto test}_case = TestFactory :: createTest(”Add”, ZType :: Z_SAFE, ””, add, 2, 3).setExpectedOutput(5).beforeAll([]()logger.info(”Init”);).afterEa$

断言机制

- **EXPECT_EQ**: 验证两个值是否相等
- **ASSERT_TRUE**: 验证条件是否为真

使用示例

[basicstyle=] $\texttt{EXPECT}_E Q(5, add(2,3)); ASSERT_T RUE(6 == add(2,3));$

异常处理

断言失败抛出 ZTestFailureException，可自定义异常处理

数据驱动测试

- 参数化测试支持
- CSV 数据导入
- 数据缓存优化：
  - 懒加载机制
  - LRU 缓存淘汰策略

示例代码

[basicstyle=] `ZTestDataManager<vector<int>, int> sum_test_data =`
$1, 2, 3, -1, 1, 0, 10, 20, 30; ZTEST_P(ArithmeticSuite, SumTest, sum_test_data) auto[inpu$

测试执行器

- 多线程并行执行
- 线程池管理
- 智能调度策略：
  - 并行运行 safe 测试
  - 串行运行 unsafe 测试
  - 串行运行 Benchmark
  - 串行运行 Parameterized 测试

```
[2025-06-06 11:44:04] [DEBUG] [Unsafe] Starting unsafe tests execution
[2025-06-06 11:44:04] [DEBUG] [Unsafe] Running test: RUN.unsafe_test_single_case1
[2025-06-06 11:44:05] [INFO] [Unsafe] Test succeeded: RUN.unsafe_test_single_case1 (1000.182633ms)
[2025-06-06 11:44:05] [DEBUG] [Unsafe] Running test: RUN.unsafe_test_single_case2
[2025-06-06 11:44:07] [INFO] [Unsafe] Test succeeded: RUN.unsafe_test_single_case2 (2000.185993ms)
[2025-06-06 11:44:07] [DEBUG] [Unsafe] Running test: RUN.unsafe_test_single_case3
[2025-06-06 11:44:10] [INFO] [Unsafe] Test succeeded: RUN.unsafe_test_single_case3 (3000.079462ms)
[2025-06-06 11:44:10] [DEBUG] [Unsafe] Execution completed - Total: 3 | Succeeded: 3 | Failed: 0
[2025-06-06 11:44:10] [DEBUG] Worker 0 started (TID: 132352662693568)
[2025-06-06 11:44:10] [DEBUG] Worker 3 started (TID: 132352294512320)
[2025-06-06 11:44:10] [DEBUG] Worker 2 started (TID: 132352302905024)
[2025-06-06 11:44:10] [INFO] [Safe] Starting parallel execution of 3 safe tests using 8 workers
[2025-06-06 11:44:10] [DEBUG] Worker 5 started (TID: 132352286119616)
[2025-06-06 11:44:10] [DEBUG] Worker 1 started (TID: 132352654300864)
[2025-06-06 11:44:10] [DEBUG] New task enqueued (Total pending: 1)
[2025-06-06 11:44:10] [DEBUG] New task enqueued (Total pending: 2)
[2025-06-06 11:44:10] [DEBUG] Worker 4 started (TID: 132352149812928)
[2025-06-06 11:44:10] [DEBUG] New task enqueued (Total pending: 3)
[2025-06-06 11:44:10] [DEBUG] Worker 6 started (TID: 132352277726912)
[2025-06-06 11:44:10] [DEBUG] Starting test [RUN.safe_test_single_case1] on thread: 132352277726912
[2025-06-06 11:44:10] [DEBUG] Starting test [RUN.safe_test_single_case2] on thread: 132352662693568
[2025-06-06 11:44:10] [DEBUG] Starting test [RUN.safe_test_single_case3] on thread: 132352286119616
[2025-06-06 11:44:10] [DEBUG] Worker 7 started (TID: 132352269334208)
[2025-06-06 11:44:11] [DEBUG] Finished test [RUN.safe_test_single_case2] on thread: 132352662693568
[2025-06-06 11:44:11] [DEBUG] Task completed in 1000ms (Remaining: 0)
```

GUI 界面

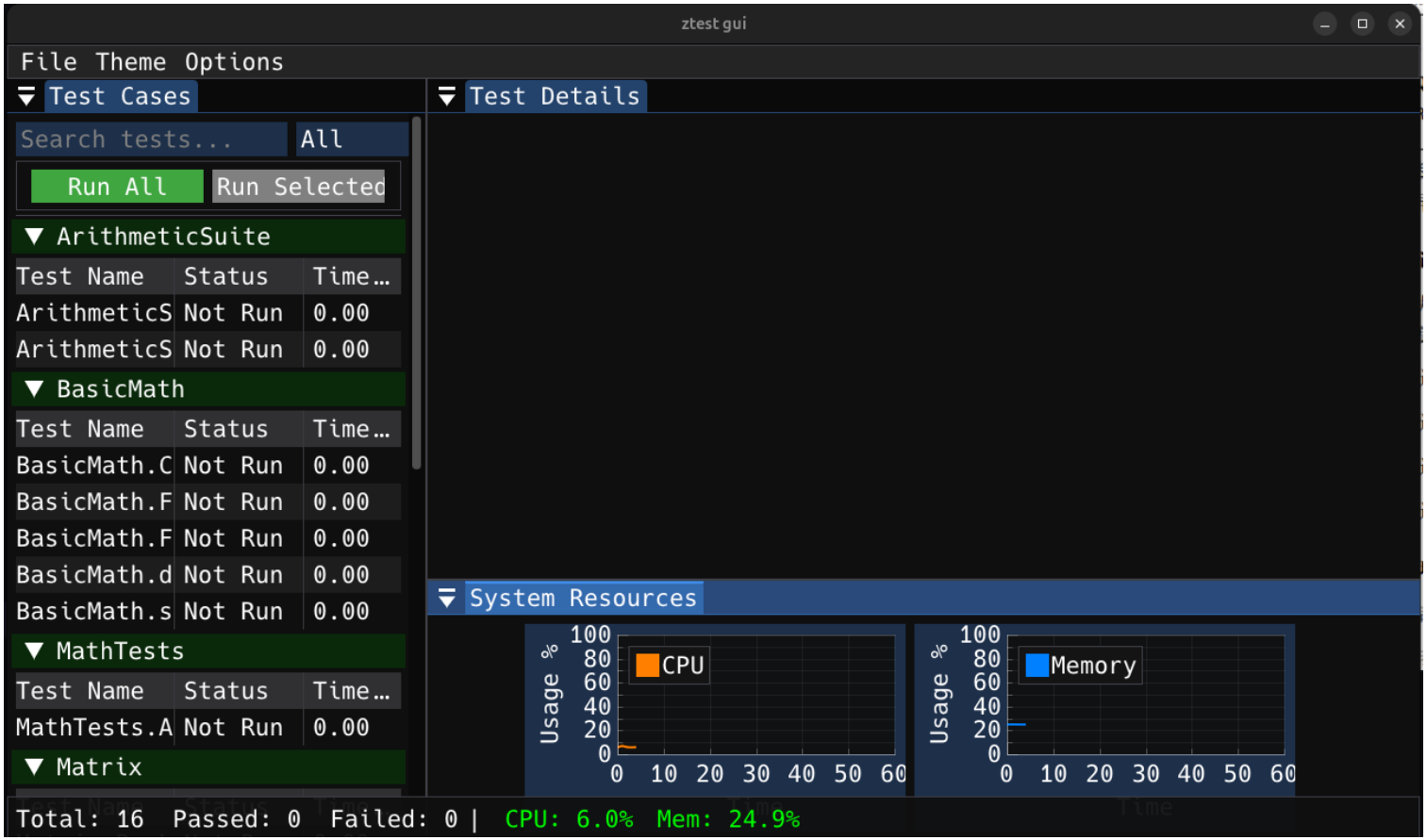
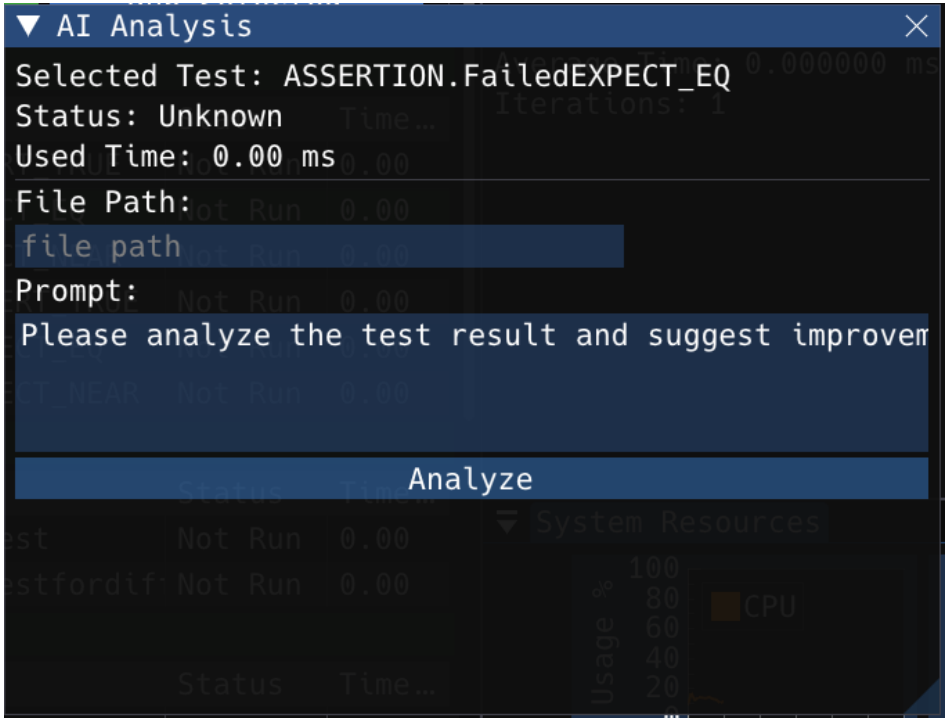


图: GUI 界面设计

AI 智能诊断

- 基于 qwen3 接口
- 功能：
  - 识别失败原因
  - 提供修复建议
  - 风险评估
  - 覆盖率分析
  - 稳定性建议



功能测试

- 断言功能测试
- 测试管理测试
- 执行器测试
- 数据驱动测试
- Benchmark 测试

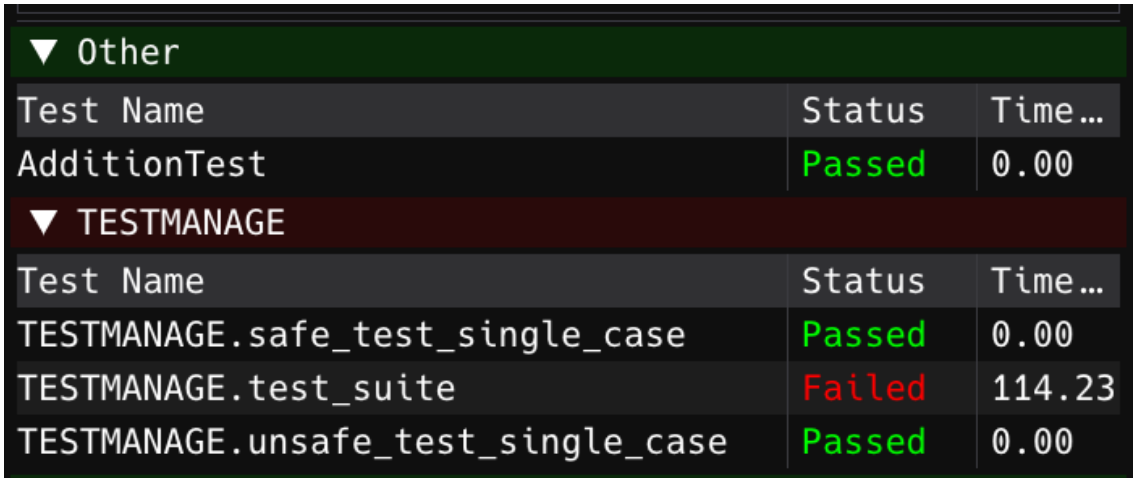
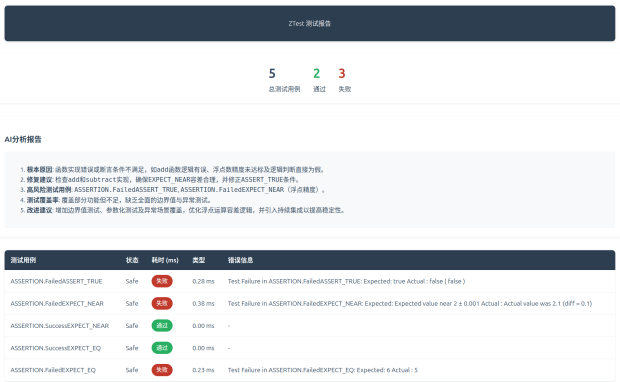
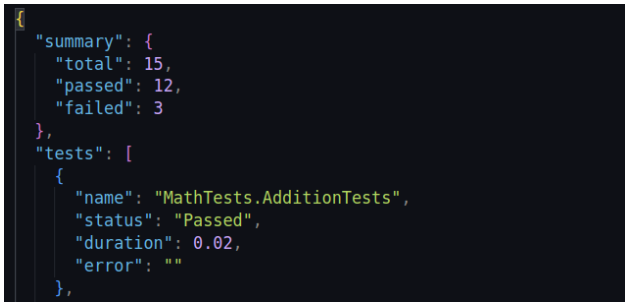


图: 测试管理功能验证

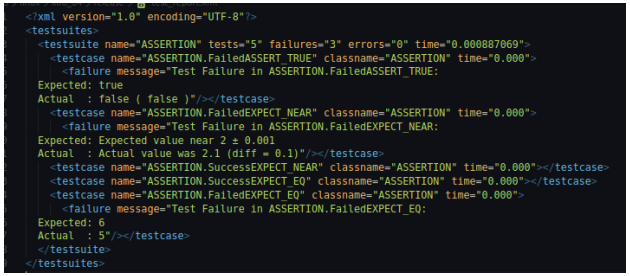
报告生成



HTML 报告



JSON 报告



XML 报告

开发进度

阶段	时间
确定主题	2024.04.26-05.03
核心代码实现	2024.05.03-05.05
线程池优化	2024.05.05-05.08
报告输出实现	2024.05.08-05.11
GUI 改进	2024.05.11-05.13
CLI 接口	2024.05.13-05.15
Benchmark 测试	2024.05.15-05.24
参数化测试	2024.05.25-05.28
AI 诊断	2024.06.02-06.04
总结工作	2024.06.04-06.05

团队分工

成员	职责
郑辰阳	架构设计，核心代码，报告
叶穗华	GUI 改进，测试逻辑
吴泓庆	GUI 改进
齐彦松	GUI 改进
王瑞箐	Windows 移植

心得体会

- 郑辰阳：深刻体会模块化设计和分层架构重要性
- 叶穗华：前端设计需考虑跨平台、库版本等因素
- 齐彦松：认识到优秀软件需兼顾技术深度和用户体验
- 王瑞箐：实践中学习，收获众多知识
- 吴泓庆：掌握 MVC 架构实践运用，提高团队合作能力

Q & A

谢谢聆听！

欢迎提问！