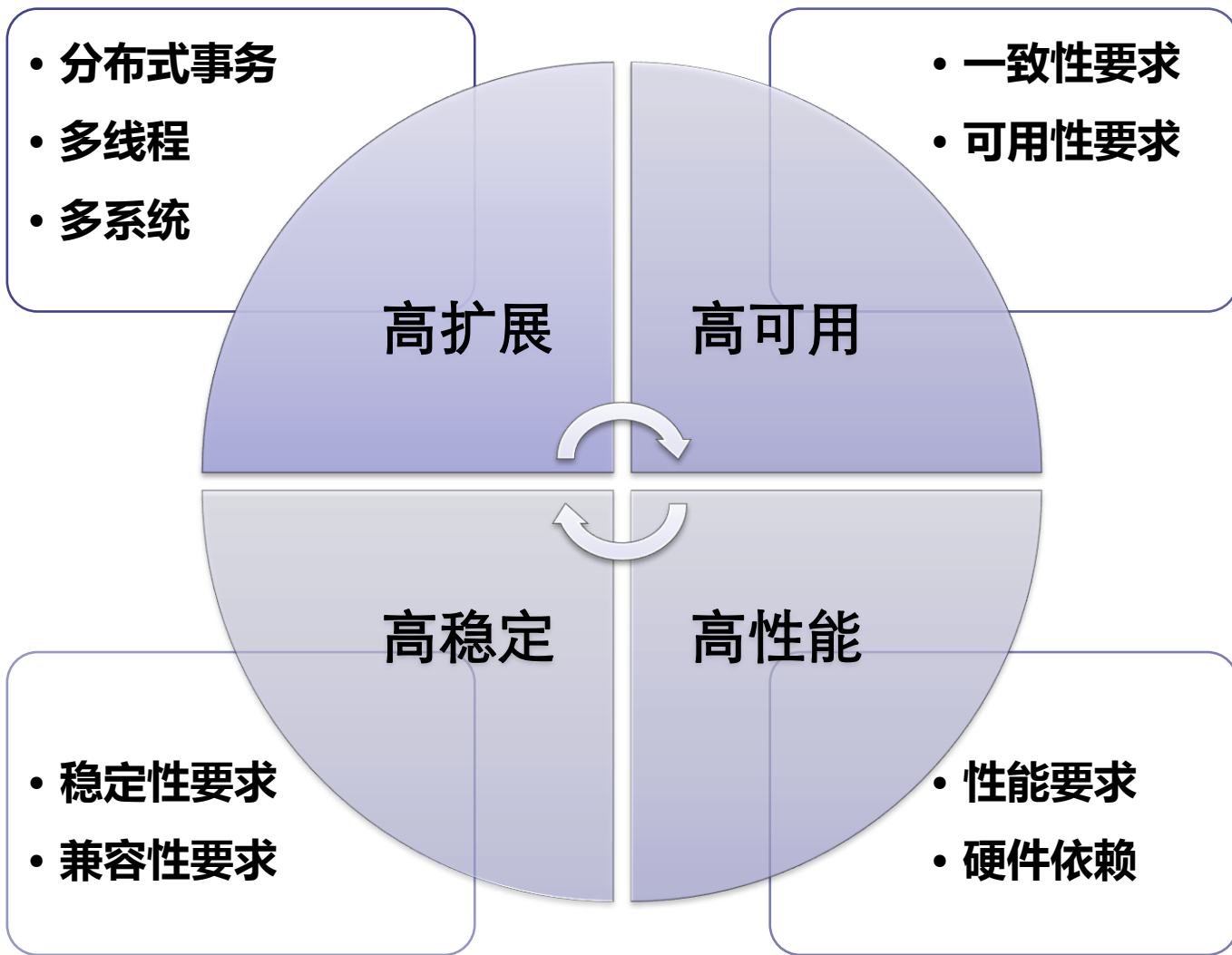


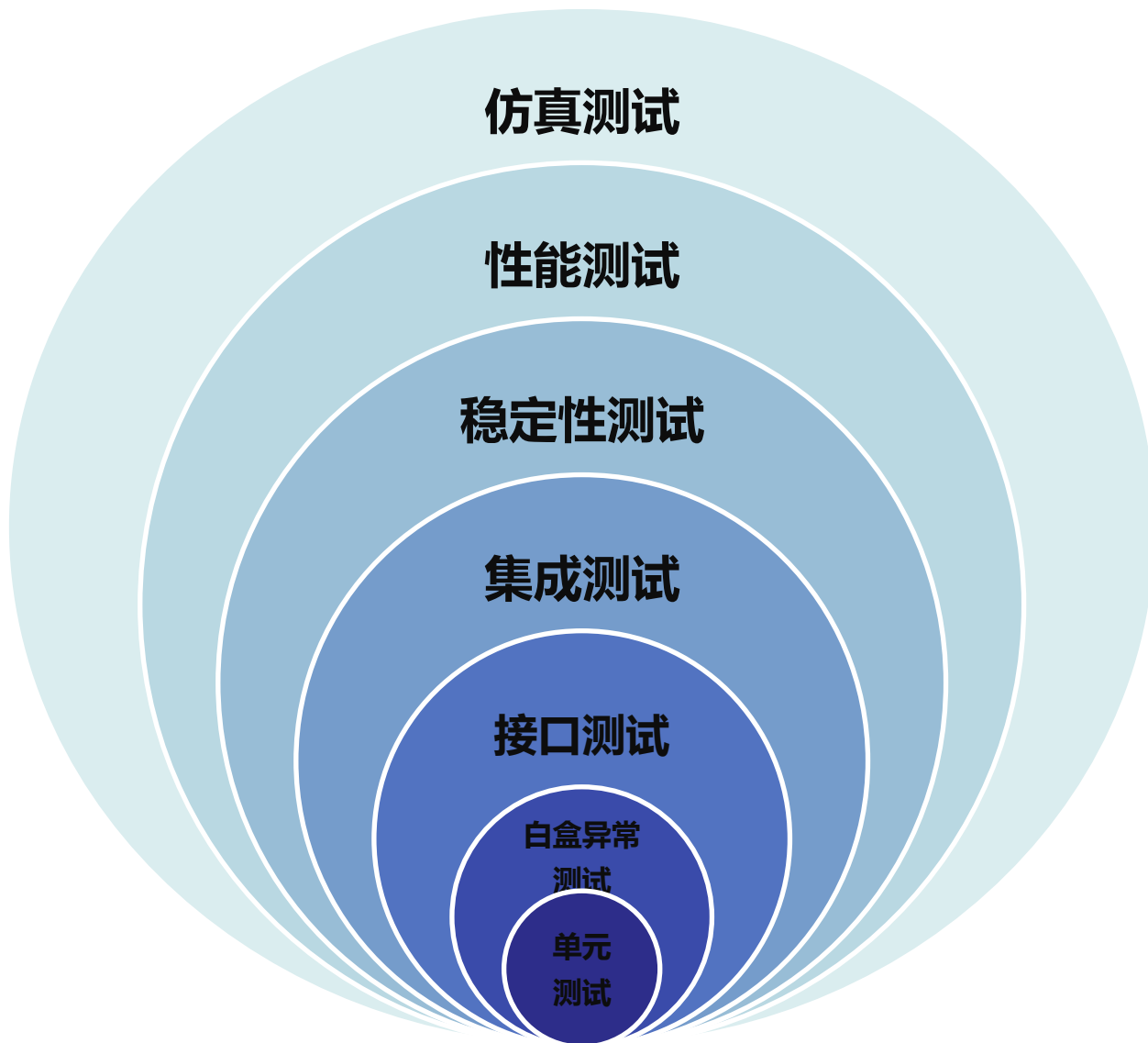
分布式系统测试实践

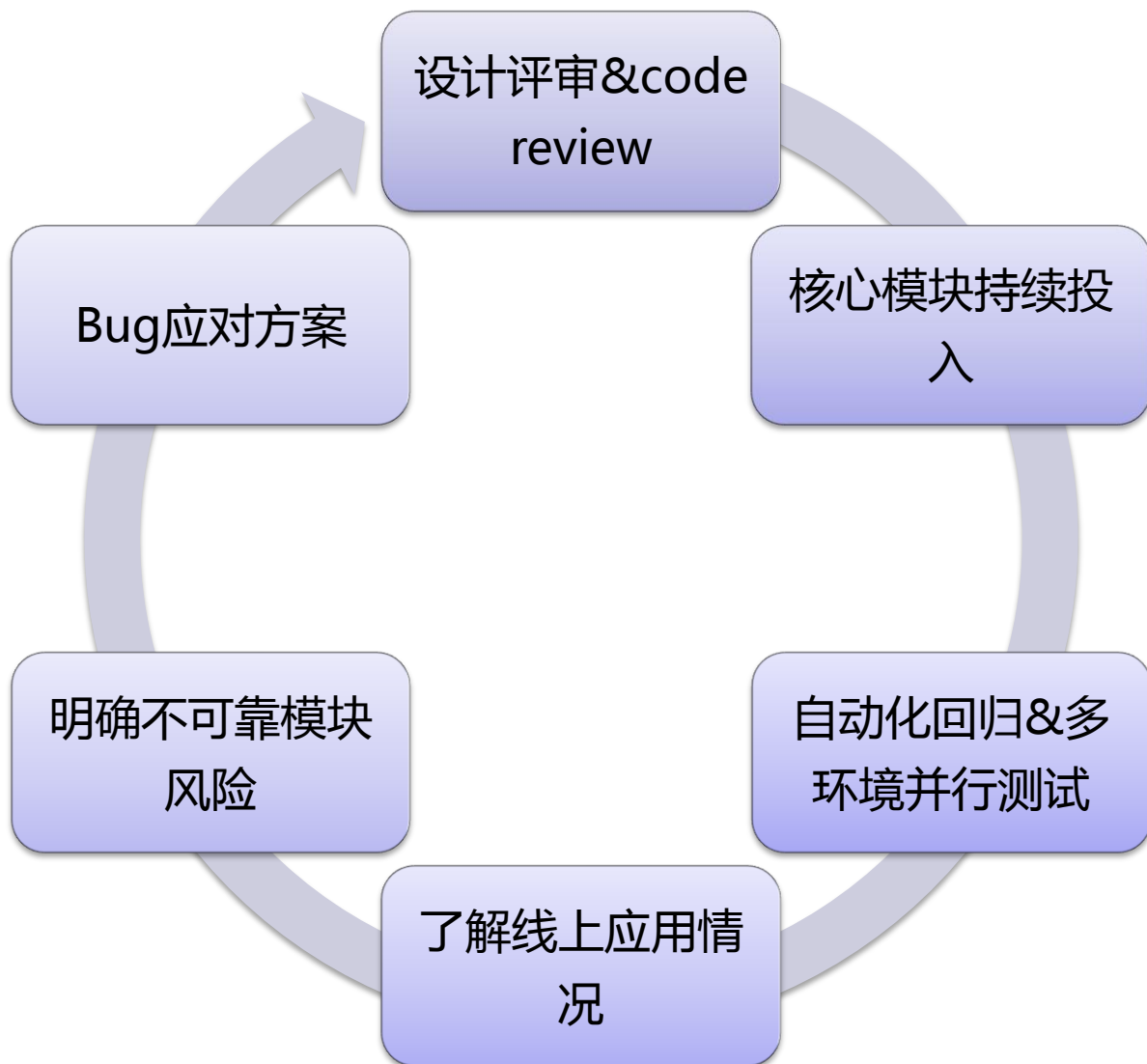
陈鑫/神秀

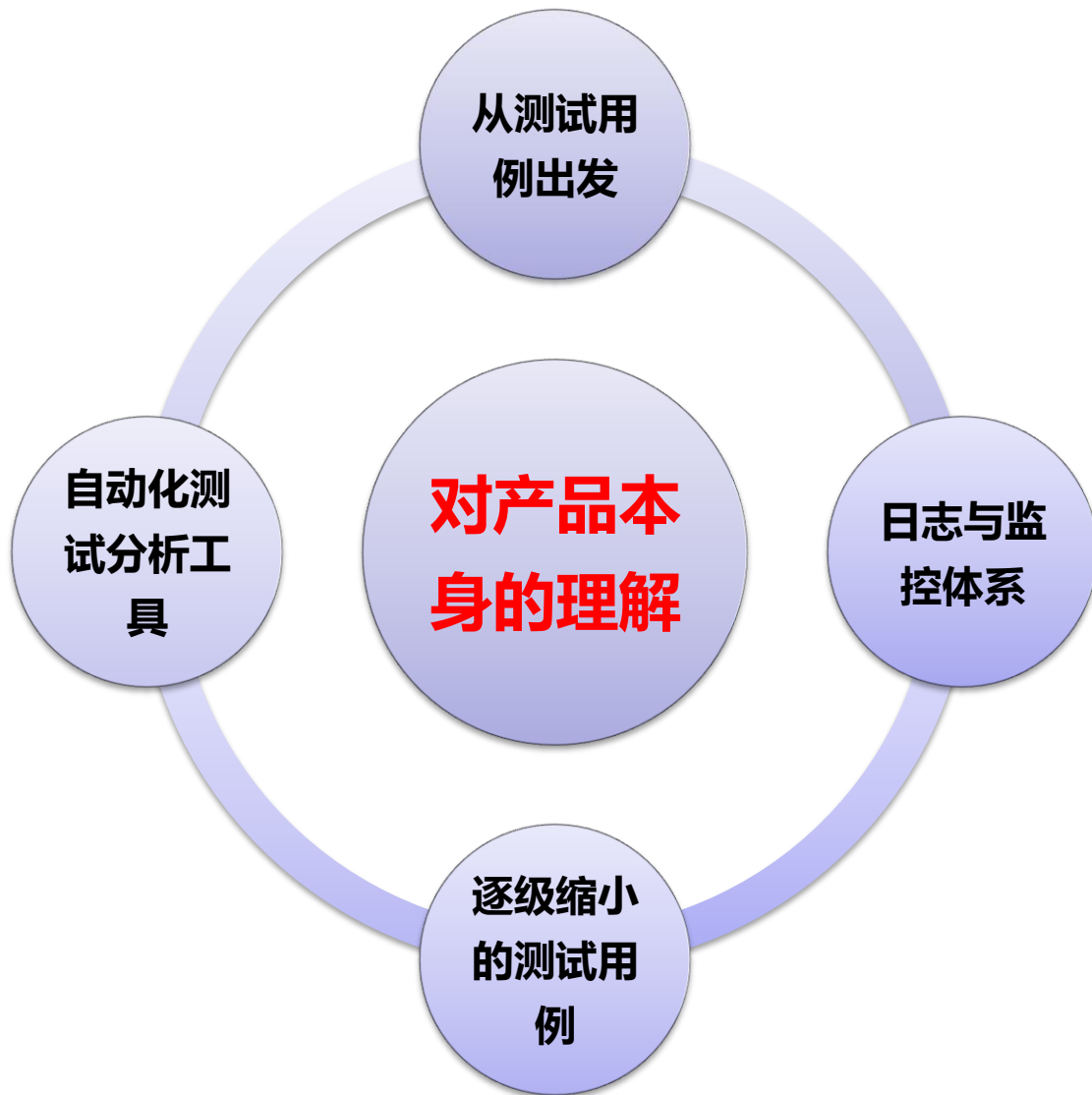


- 分布式系统测试的难题和应对
- 分布式系统测试的自动化
- 实践经验分享









■ 心目中给力的测试工具

- 轻松编写用例
- 数据自动采集分析
- 最大程度复用现成的工具
- 部署、构建、执行自动化，每天看看报告就行



有这样的工具么？

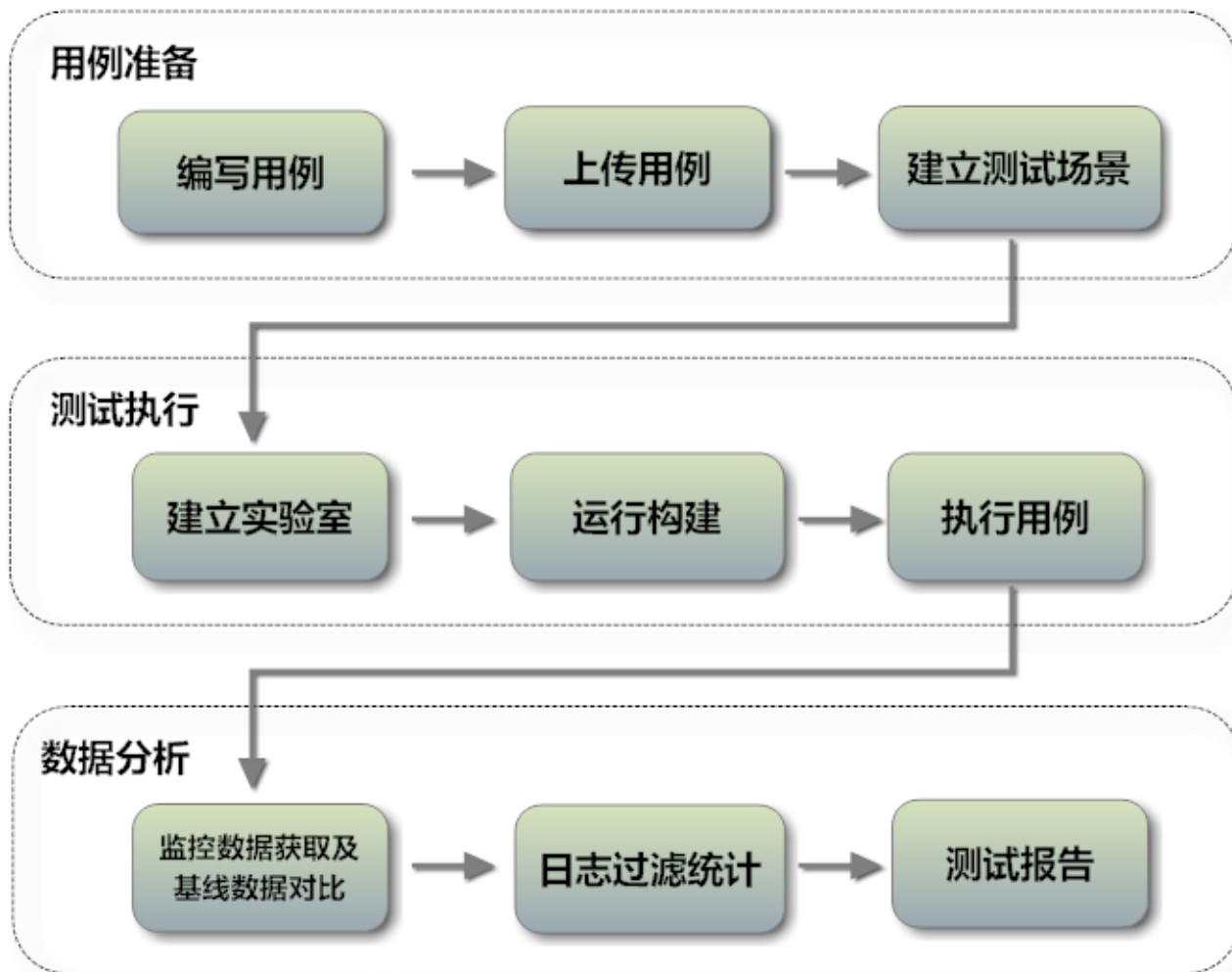
■ DST（分布式系统测试工具）

- 支持编写测试用例实现多机并行测试
- 可集成多种已有的测试工具及用例
- 可配置的监控数据自动收集与展示
- 日志自动分析与查看
- 可扩展的任务执行控制功能
- 性能、功能结果对比
- 测试报告自动生成

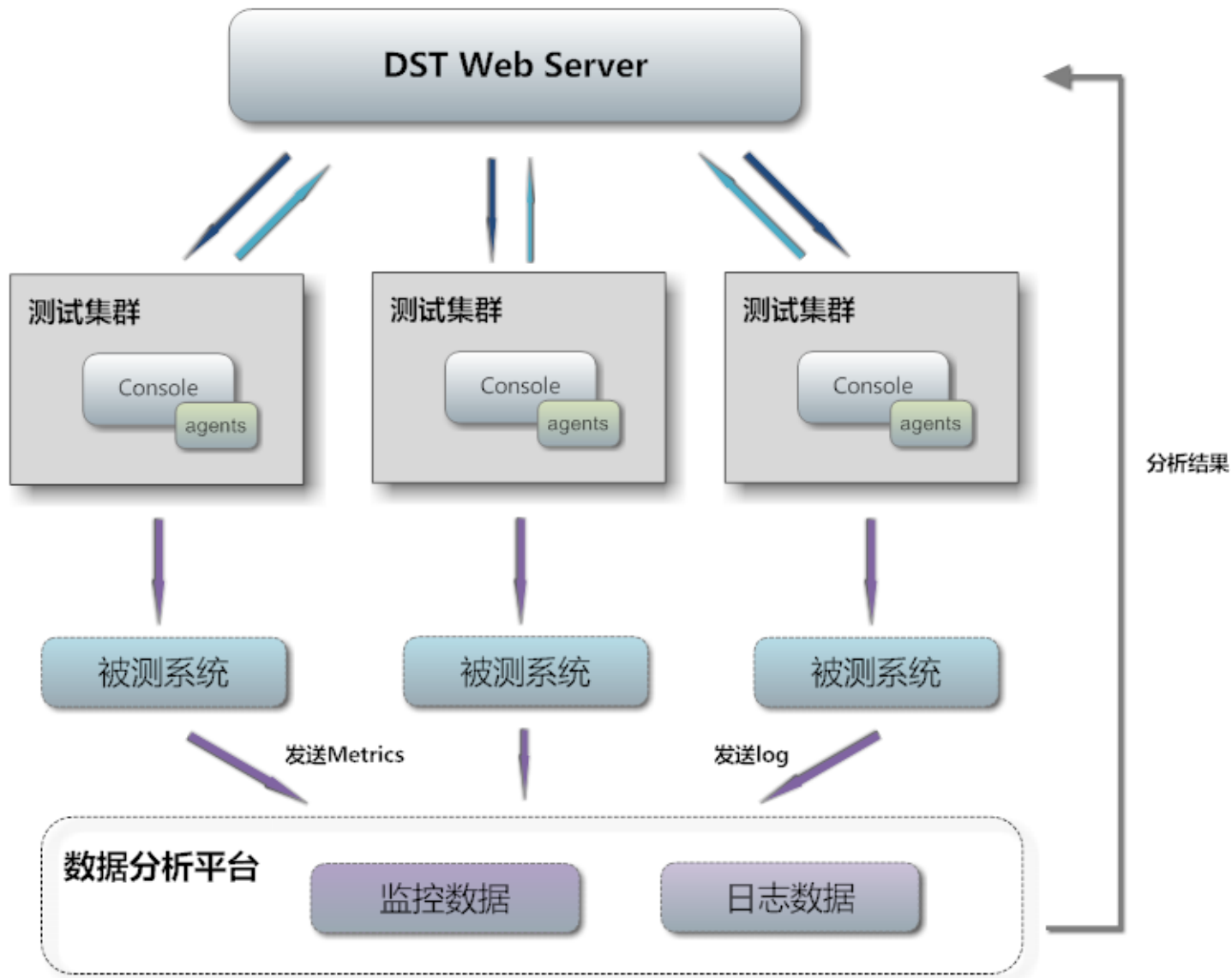
DST



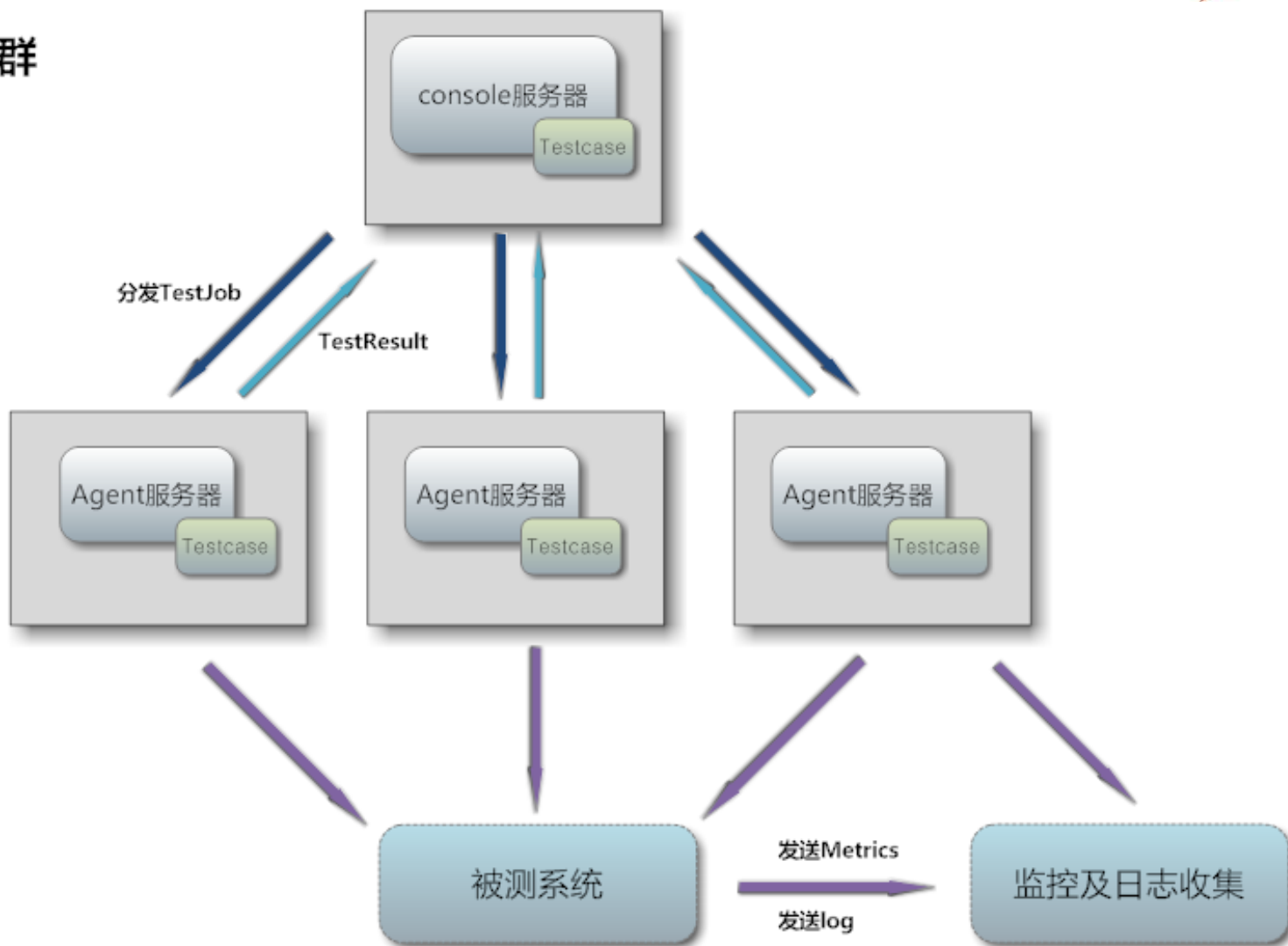
测试过程



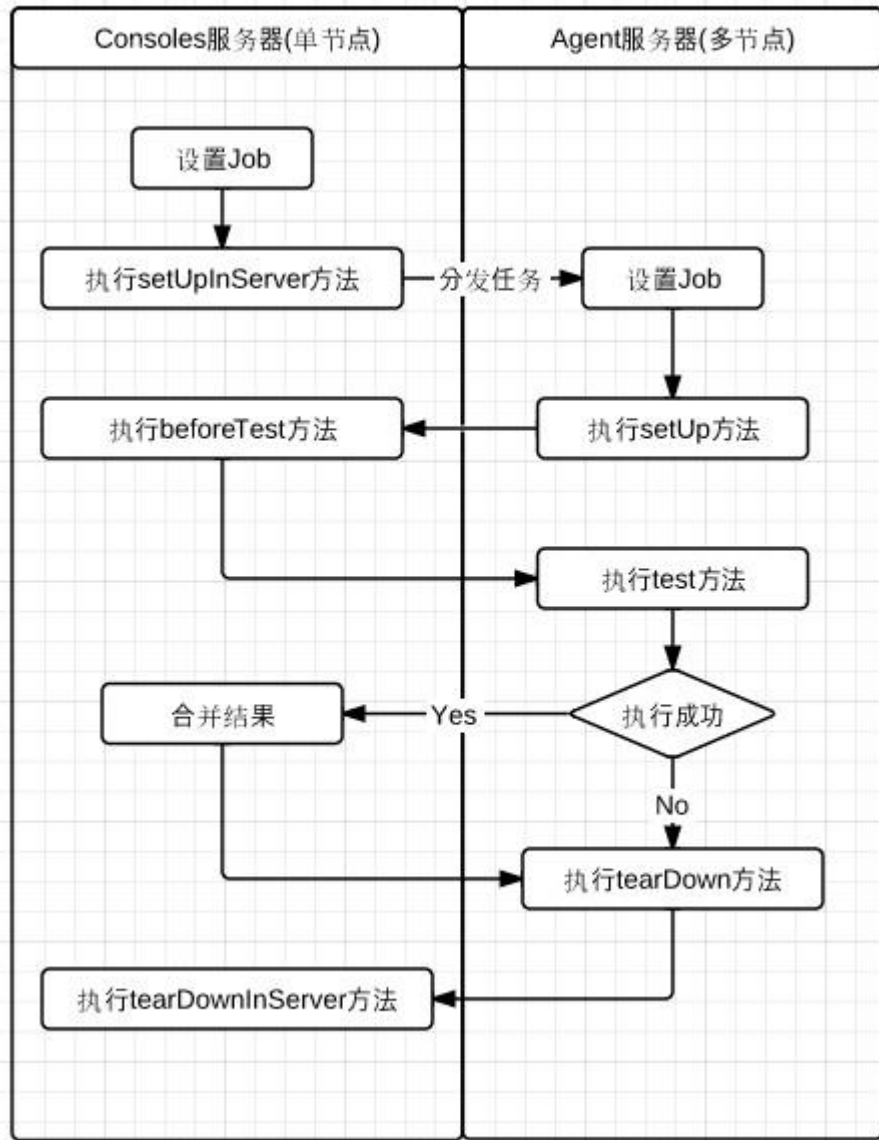
DST



测试集群

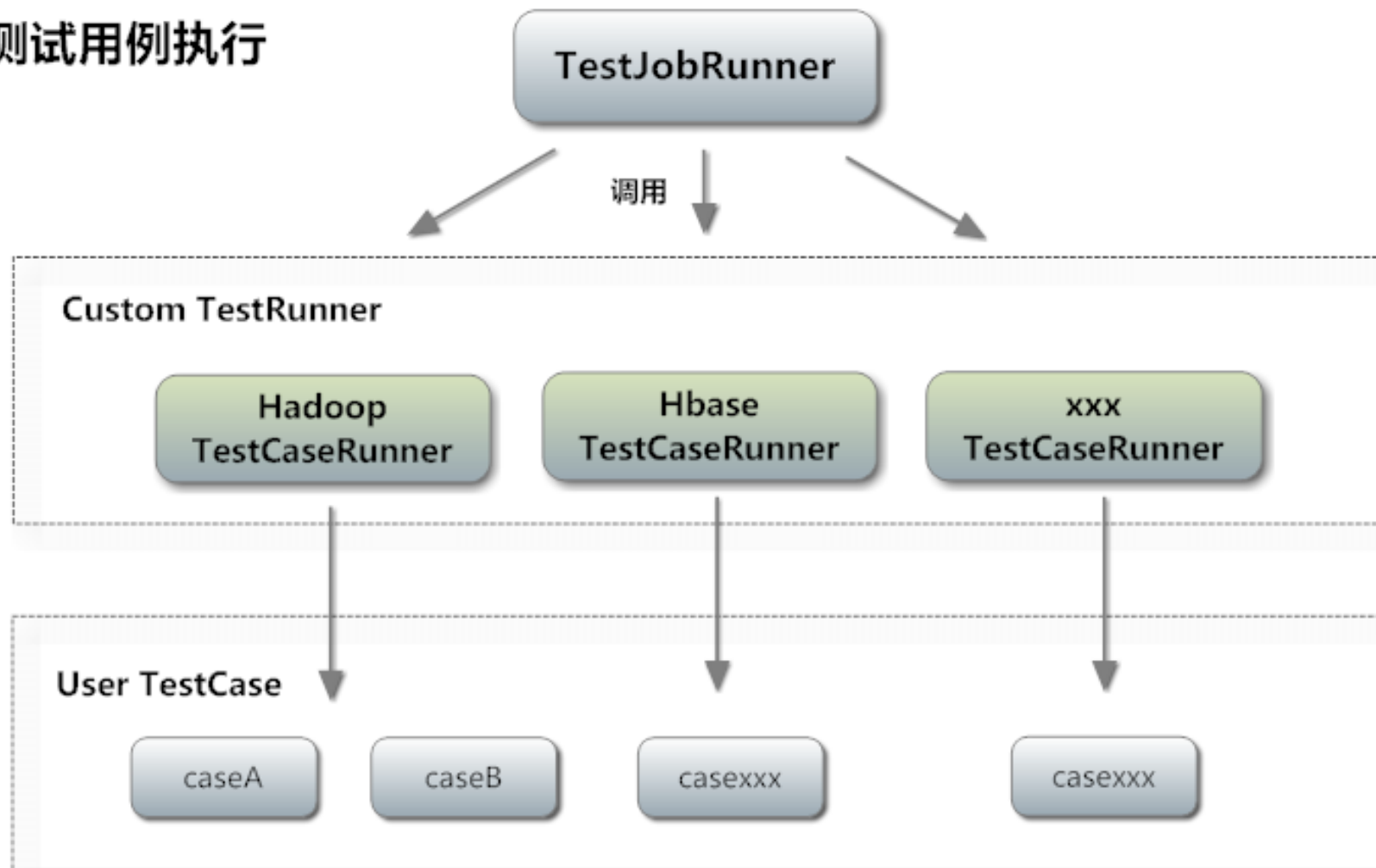


■ 按照单元测试的方式
编写集成、性能测试用例

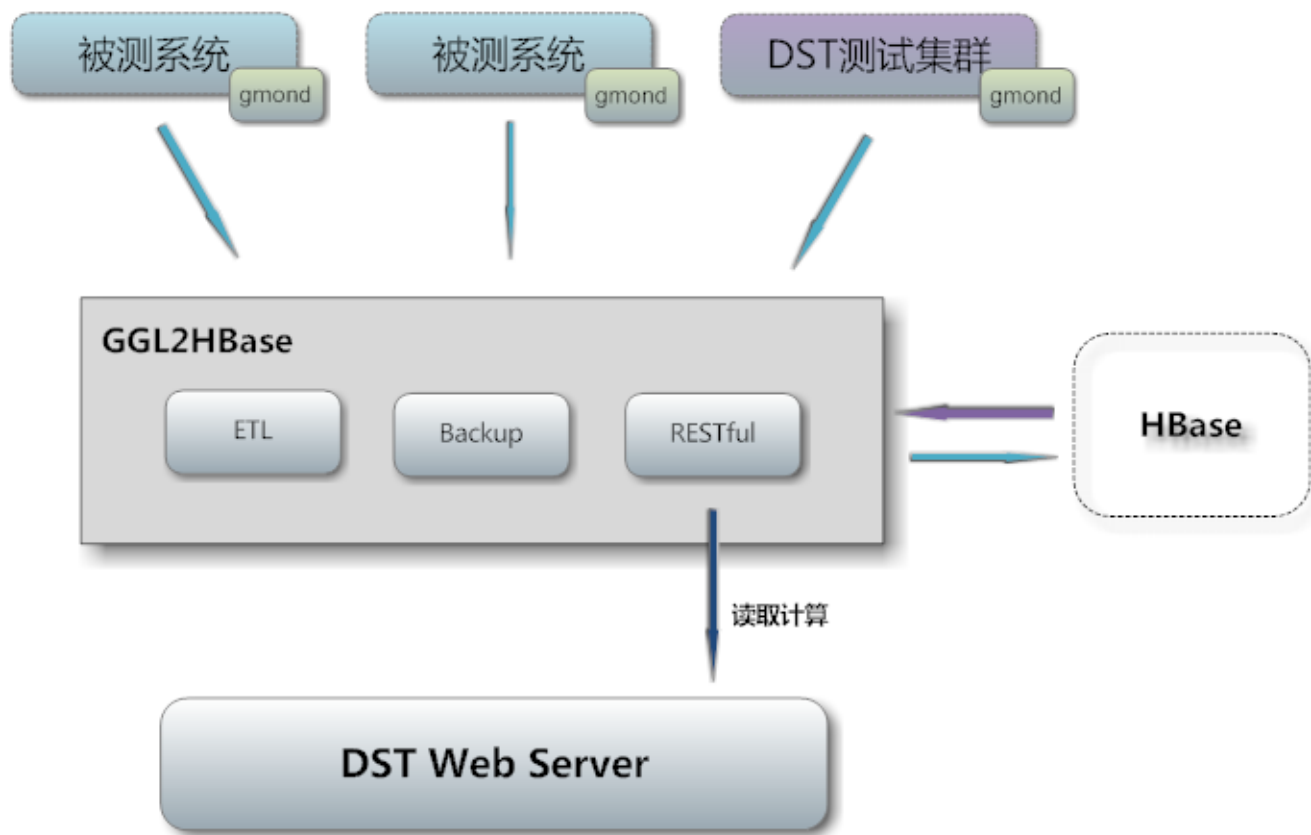


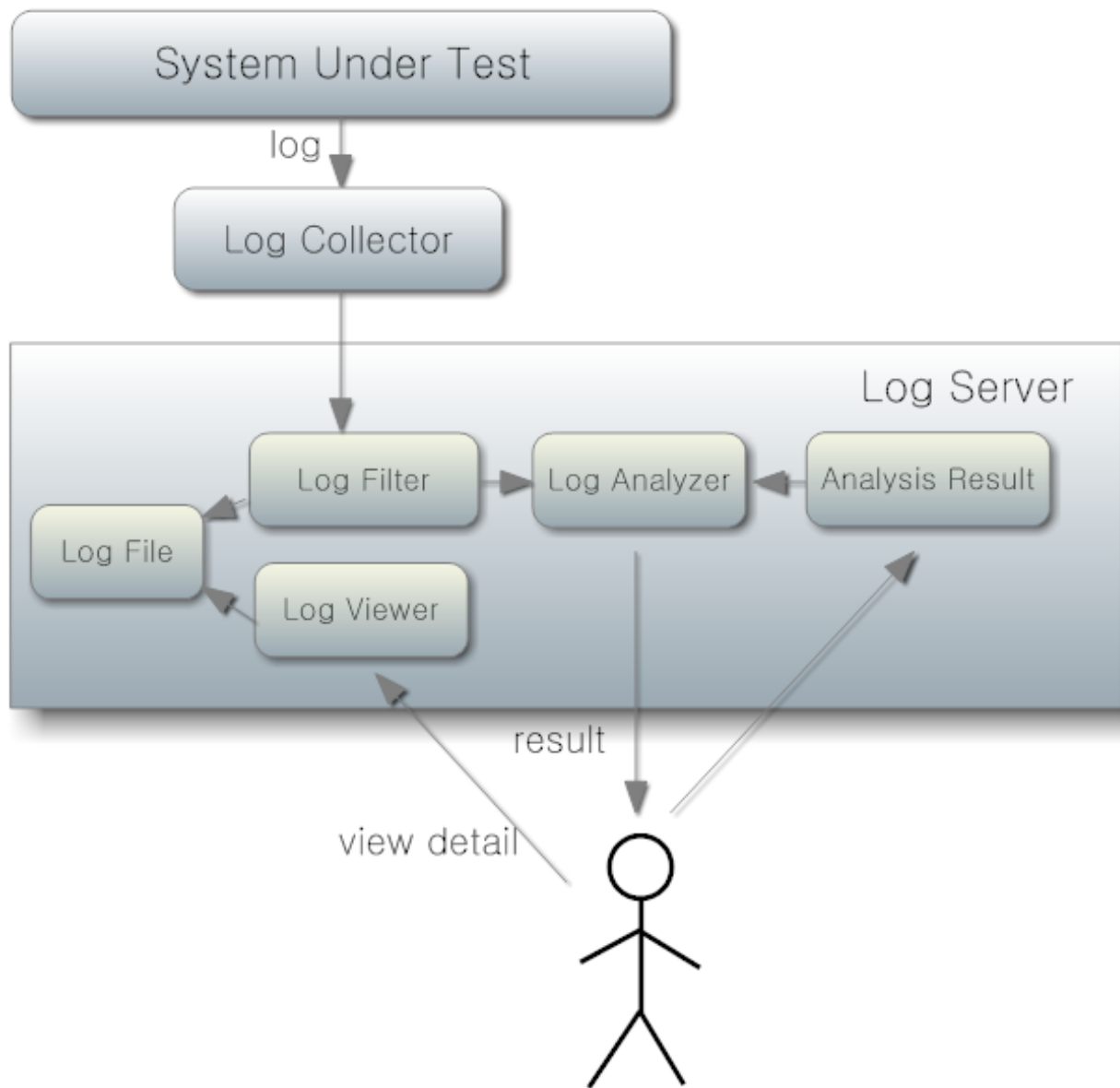
- 通过实现TestRunner来集成多种测试工具

测试用例执行



监控体系





The screenshot displays the Kelude HDFS/ADFS monitoring interface. At the top, there is a search bar with the text "请输入关键字/编号" and a dropdown menu set to "TC". Below this, a navigation sidebar on the left contains icons for "场景管理", "任务管理", "集群管理", "性能监控", and "日志查询". The main content area is divided into two sections: "构建详情" (Build Details) and "测试数据" (Test Data).

构建详情 (Build Details): This section shows a table of job execution records for "perf-dfsio-write-003".

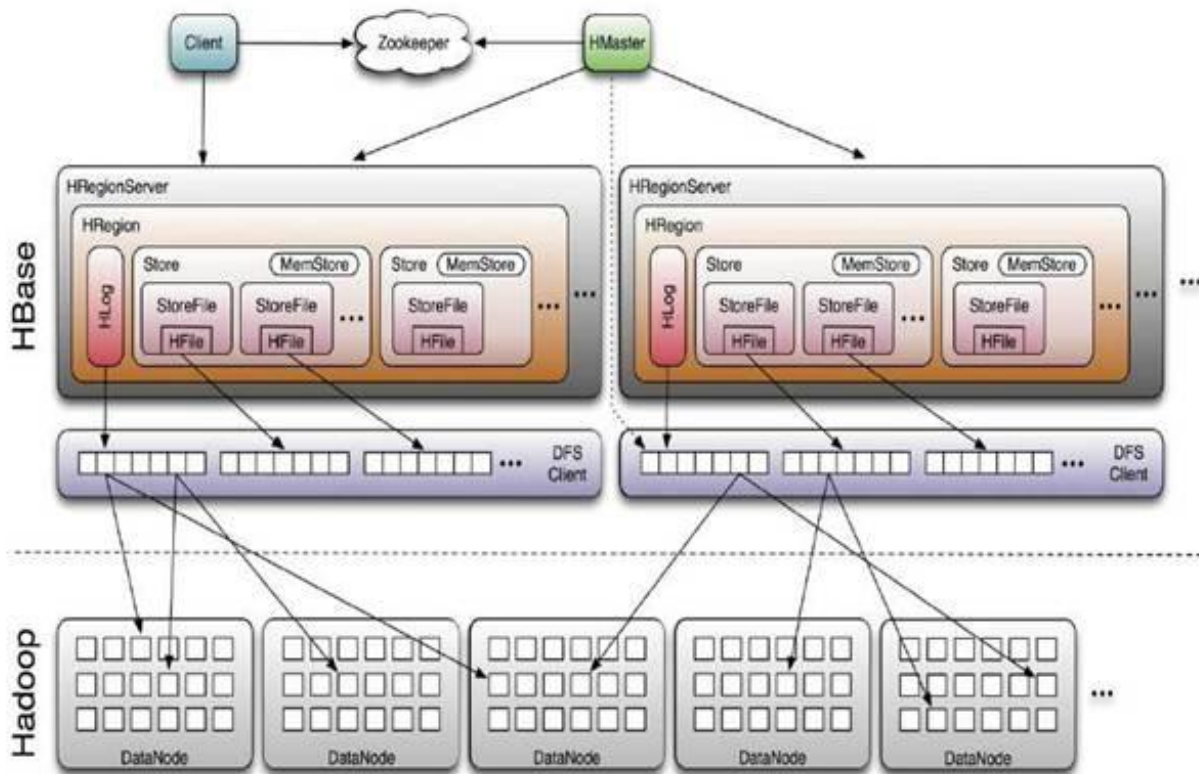
序号	开始时间	结束时间	时长	总数	失败数	成功数
1	2012年05月25日 19:35:24	2012年05月25日 20:38:26	1小时3分2秒	7	0	7
2	2012年05月25日 18:24:34	2012年05月25日 19:35:24	1小时10分50秒	7	0	7
3	2012年05月25日 17:15:24	2012年05月25日 18:24:34	1小时9分10秒	7	0	7
Σ	-	-	3小时23分2秒	21	0	21

测试数据 (Test Data): This section shows performance metrics for "perf-dfsio-write-004". It includes a table comparing current performance (2012年5月DFSIO性能) with a baseline (2012年2月DFSIO性能).

序号	名称	汇总	基线对比	
			2012年5月DFSIO性能	2012年2月DFSIO性能
1	Launched map tasks	6.03 K	6.03 K (0.00%)	6.03 K (0.00%)
2	Data-local map tasks	6.02 K	6.03 K (-0.08%)	6.03 K (-0.09%)
3	Combine output records	0.00	0.00 (0.00%)	0.00 (0.00%)
4	Counters	17.00	17.00 (0.00%)	17.00 (0.00%)
5	Namenode(version)			
6	Namenode(compile time)	May 25 15:41:30 2012	Apr 26 11:47:59 2012	Mar 14 10:16:37 2012
7	Map output bytes	517.85 K	517.80 K (0.01%)	517.86 K (-0.00%)
8	Jobtracker(compile time)	Apr 26 11:47:59 2012	Apr 26 11:47:59 2012	Mar 14 10:16:37 2012
9	HDFS bytes read	698.37 K	698.37 K (0.00%)	698.37 K (0.00%)
10	Jobtracker(version)			
11	fileSize (MB)	500.00	500.00 (0.00%)	500.00 (0.00%)
12	bufferSize	4.10 K	4.10 K (0.00%)	4.10 K (0.00%)
13	created control files for	6030 files	6030 files	6030 files
14	Rack-local map tasks	7.67	3.00 ↑ (155.56%)	2.33 ↑ (228.57%)
15	Reduce output records	5.00	5.00 (0.00%)	5.00 (0.00%)
16	replication	3.00	3.00 (0.00%)	3.00 (0.00%)
17	HDFS bytes written	3161.46 G	3161.46 G (0.00%)	3161.46 G (-0.00%)
18	Reduce input groups	5.00	5.00 (0.00%)	5.00 (0.00%)
19	Test exec time sec	832.34	845.49 (-1.56%)	865.67 (-3.85%)

分布式事务最难搞

- 单系统、单机出现异常不能影响事务正确性
- 不可过分信任依赖系统
- 系统设计时的检查更为重要



■ 日志分析-发现bug的重要手段

- 健全和准确的日志（完整、级别得当、准确、可读）
- 测试用例预期达到并不代表程序无bug
- 通过日志检验程序行为是否得当，发现性能问题
- 日志是快速理解程序和定位bug的重要手段

■ 没有压力就没有bug

- 集成测试侧重于系统满负荷的测试
- 多线程bug需要压力来放大出现概率
- 极限情况的稳定性才能说明问题
- 系统需要对压力有自我适应能力

■ 性能的小问题不容忽视

- 关键性能指标看不到是系统稳定性的地雷
- 通过关注测试系统的性能表现可以快速发现线上系统隐患
- 测试人员比开发人员对线上性能更有发言权
- 及时的给出测试数据和改进意见是测试价值的体现
- 关注线上性能表现可以完善测试用例，更贴近实际

■ GC是性能的重要因素

- 减少GC暂停时间是优化的目标
- 避免内存碎片对应用的影响
- 观察线上系统GC状况避免故障（内存泄露、FullGc）
- Gc 日志和gc 监控帮助我们发现最合理的配置

随机写 TPS: 11060 → 12763 (+15%)

■ 线上最容易发现隐患

- 复杂性决定着不可能100%穷举测试场景
- 线上日志和性能监控数据帮助及时发现bug
- 线上问题分析可帮助准确抓住测试重点
- 硬件参数开发不关注+测试不知道=线上出故障
- 理解用户需求，为提升产品可用性提供建议也是测试人员的价值

■ 有bug也不能影响系统稳定

- 系统足够健壮的必须要求
- 假设出现bug，系统是否能够自我调节或报警
- 有没有足够的应对措施和手段
- 系统不可能没bug，往往出bug如何处理更重要

Q&A

旺旺账号：神秀

E-mail：shenxiu.cx@taobao.com

谢谢！