

360技术嘉年华—测试之美

智能任务调度系统

任务调度系统的智能化进阶之路

个人简介

- 李珂
- 测试开发专家
- 专注于测试技术创新与落地

提纲

- AI改造测试
- 背景介绍
- 任务调度系统简介
- 任务分发的痛点
- 适合AI落地的原因
- AI实现介绍
- 应用结果
- 优化计划

AI改造测试

- 测试为什么需要AI
 - 生产资料——测试数据
 - 生产力——AI
 - 生产关系——质量体系
- AI落地场景
 - 符号主义
 - 分析海量数据（用例、代码、崩溃等）得到模型，辅助测试
 - 连接主义
 - 计算机视觉（文字识别、目标检测等）
 - 行为主义
 - 强化学习（UI遍历、游戏自动化测试）
- AI测试愿景
 - 人——培养具备AI工程应用和算法基础的人才梯队
 - 事——tester训练AI，AI为tester提效

背景介绍

- 什么是任务调度系统？
 - 将任务派发到执行节点的中央系统
- 任务调度系统的目标是？
 - 将任务发送到执行最快的节点上
- 哪个是执行最快的节点？
 - 服务器空闲资源大
 - 服务器磁盘读写速度快、碎片率低
 - 服务器错误日志少
 - 虚拟机回滚、运行快
 -
- 业界相似场景
 - 工业界的设备故障预测
 - 网络设备流量的智能控制
 - Mate20 Flex-Scheduling

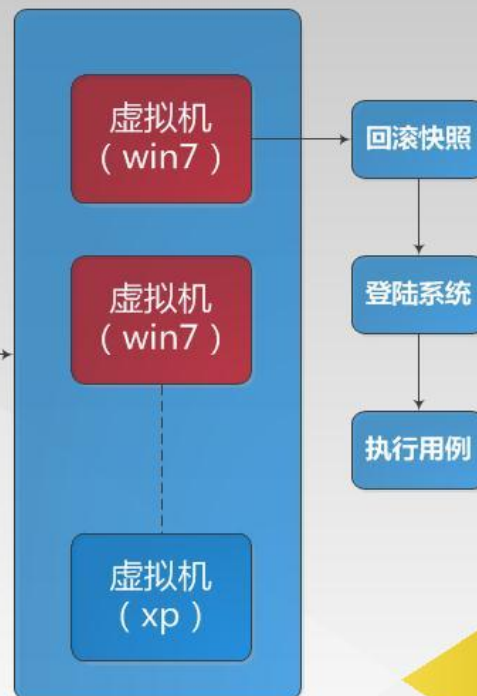
任务调度系统简介

自动化用例管理平台



用例分发中控

虚拟化云任务执行平台

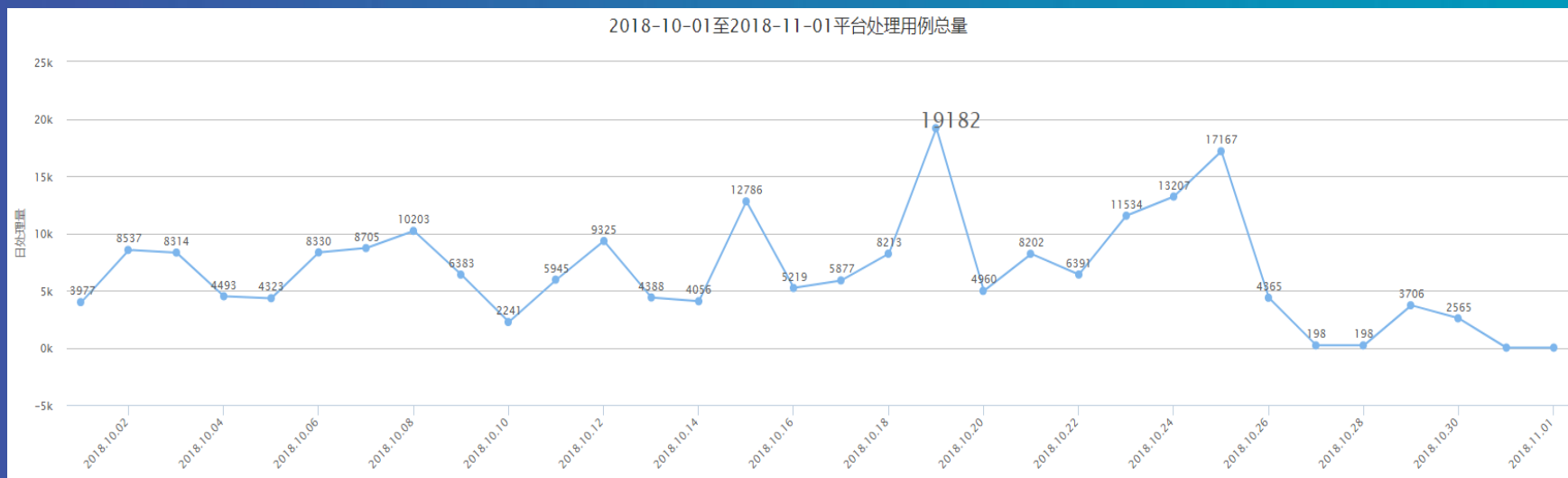


任务分发的痛点

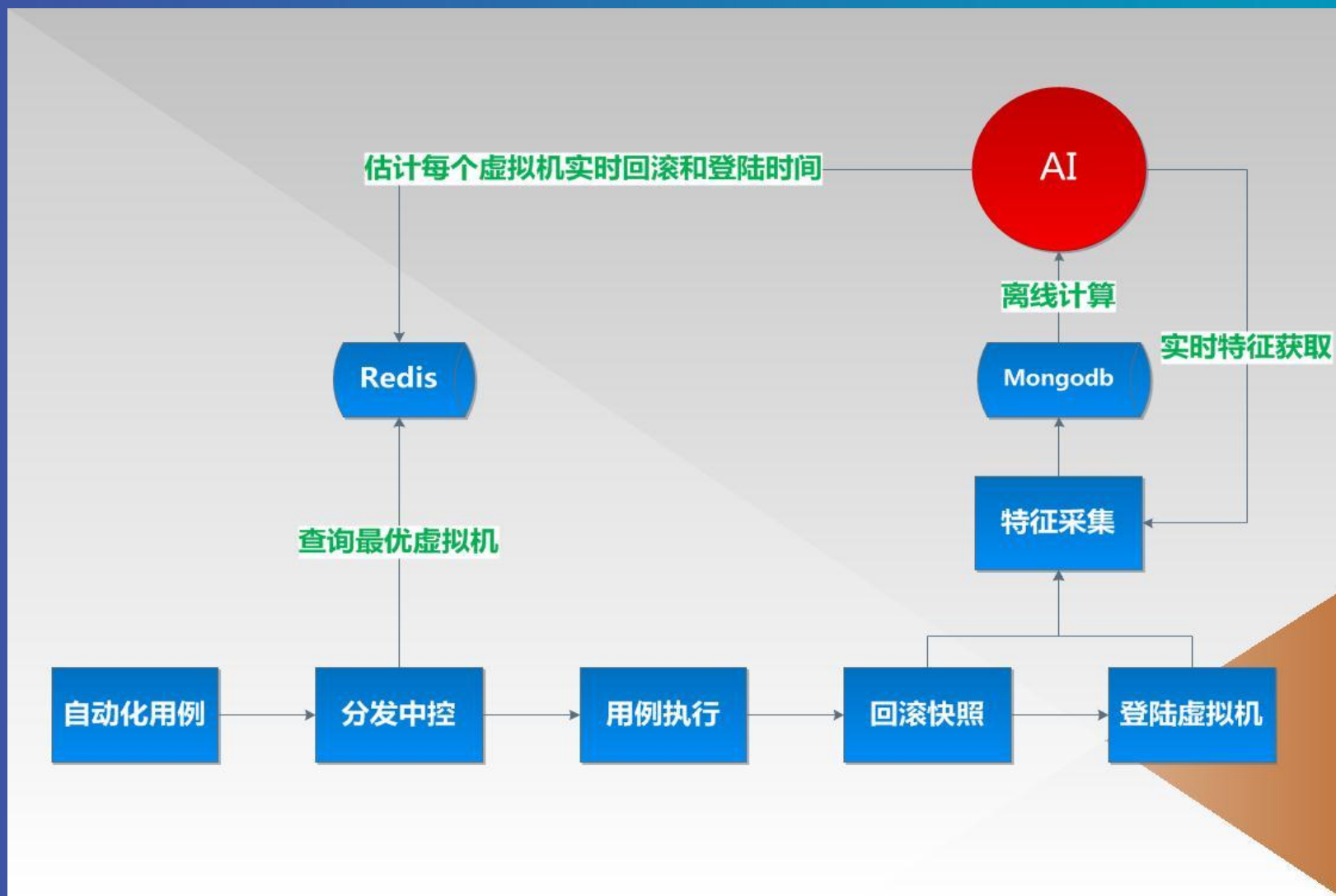
- 从多个目标虚拟机中随机分发
- 回滚虚拟机超时延长任务执行时间
- 登陆虚拟机超时延长任务执行时间
- 回滚与登陆的速度与固定的特征相关

适合AI落地的原因

- 任务分发与任务执行环境特征相关
- 日均万次用例执行方便数据积累、算法验证
- 算法简单成熟快速落地
- 提升结果影响大



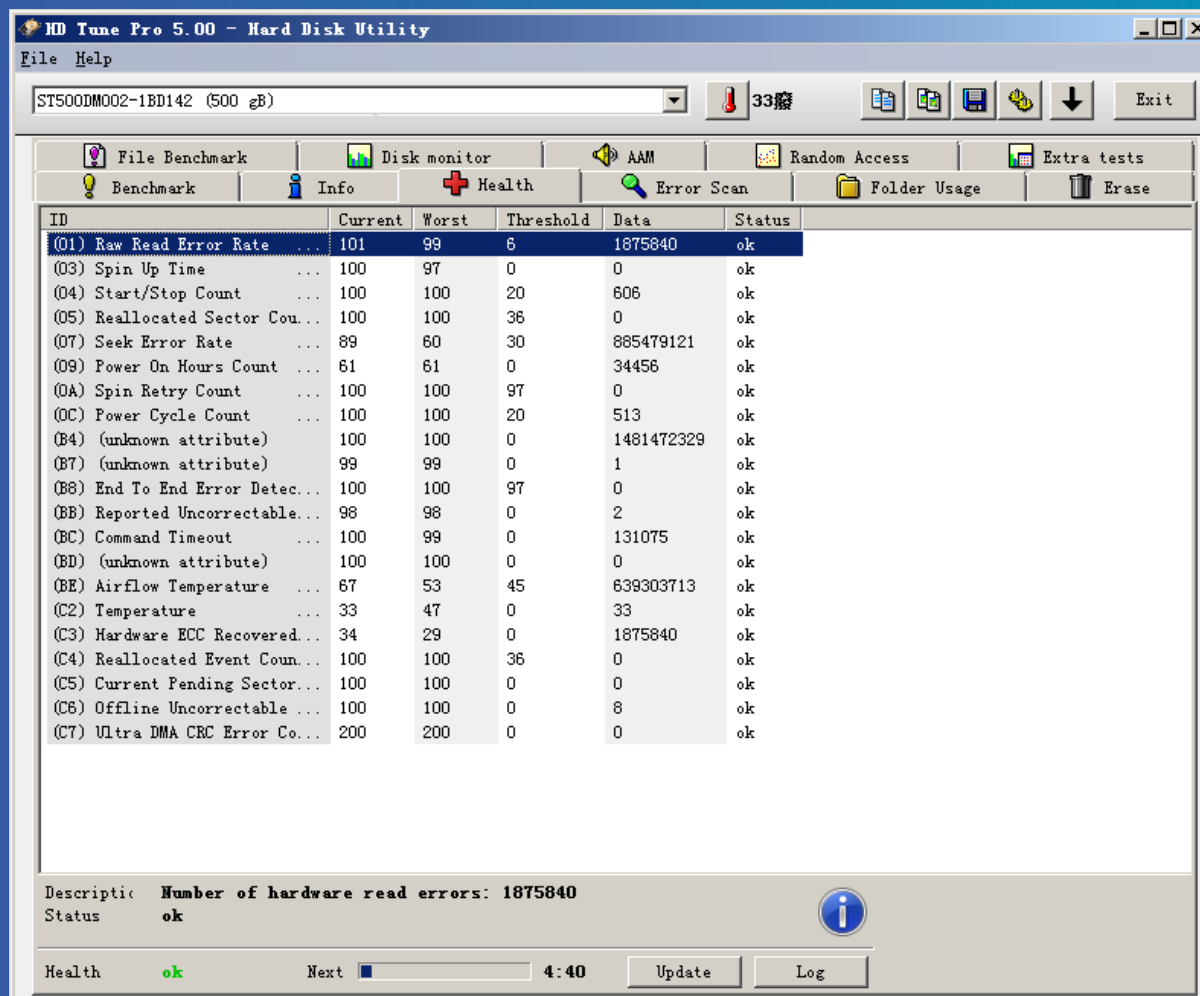
AI实现介绍——框架结构



AI实现介绍—数据特征与采集

- 服务器特征
 - CPU
 - Memory
 - Disk IO
 - Disk Usage Percent
 - Net IO
 - Running VM Number
 - Disk Health Features
- 虚拟机特征
 - Memory Size
 - CPU Kernel Number
 - System
- 虚拟机回滚和登陆后采集特征数据以及时间存储到
mongodb

AI实现介绍—Disk Health Features



AI实现介绍——Disk Health Features

大部分服务器采用Raid存储方式，导致硬盘检查工具无法使用

```
smartHddinfo = wmi.WMI(moniker="//./root/wmi").query("Select * from MSStorageDriver_FailurePredictData")
hddDrives = c.query("SELECT * FROM Win32_DiskDrive")
for index, hddDrive in enumerate(hddDrives):

    hddDrive.SerialNumber
    hddDrive.Model
    smartHdd.setdefault('InterfaceType',hddDrive.InterfaceType)
    smartdata = smartHddinfo[index].VendorSpecific
    for i in range(30):
        id = smartdata[i * 12 + 2]
        if id == 0:
            continue
        flags = smartdata[i * 12 + 4]
        failureImminent = (flags & 0x1) == 0x1
        value = smartdata[i * 12 + 5]
        worst = smartdata[i * 12 + 6]
```

AI实现介绍——数据处理

- 去掉方差变化小的特征 `sklearn.VarianceThreshold`
 - 方差小的特征对于结果的影响小
- one-hot编码 `sklearn.OneHotEncoder`
 - 处理枚举变量
- 标准归一化 `sklearn.StandardScaler`
 - 统一特征的值范围
- 选取主要特征 `sklearn.SelectKBest`
 - 选取与结果相关性强的特征，需要预设特征数量

AI实现简介——回归算法选择

- linearRegression
- ElasticNet
- svmRegression
- decisionTreeRegression
- **lassoRegression**
 - **L1正则化 稀疏特征**
- **ridgeRegression**
 - **误差最低**
 - **L2正则化 提高泛化能力**

应用结果

- 根据最近30天的数据训练模型
- 随机算法平均回滚时间76秒，登陆时间7.1秒，统计周期一周
- AI优化后平均回滚时间63秒，登陆时间6.7秒，统计周期一周
- 效率总体提升约20%左右

优化计划

- 特征挖掘
 - Raid硬盘的状态数据
 - Windows系统错误日志 (win32evtlog)
- 利用时间序列相关性
 - 目前是单点瞬时采样，不具备时间轴上的预见能力
- 深度学习
 - 优化模型，提高精度
 - RNN/LSTM处理时间序列数据

谢谢 Q&A

