开发高性能ASP.NET应用

主讲：桂素伟 / ネットスターズ

我的公众号《桂迹》

题纲

1、制定指标

2、设计应用

3、正确测试

4、性能优化

（注：本次分享仅是个人经验，没有方法论）

高性能：

不一定是架构出来的，
但一定是优化出来的。

制定指标——收集

1. 定位“热路径”的API；定位核心API
2. 确定API性质：CPU密集型，内存密集型

项目名称	API	功能说明	性能聚焦	性质
风控系统	风控验证	黑白名单验证：在黑名单或不在白名单中即为违反规则。	✓	CPU密集型
	规则管理	黑白名单规则的管理维护	✗	
	……	……		

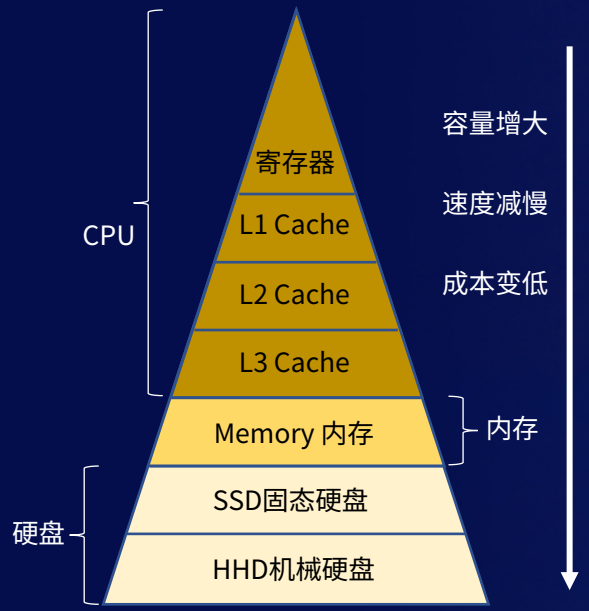
项目名称	API	功能说明	性能聚焦	性质
账务系统	入账	负责完成账务的验证，试算平衡和账户的复式记账	✓	内存密集型
	开户	开通账户	✓	?
	……	……		

制定指标——制定

	指标名称			指标值
指标	业务指标	TPS（2C2G）		2000
		响应时间	ART	10
			P95	12
			P99	15
		成功率		100%
		并发线程		20
	稳定性指标	压力持续时间		>=8h
		压力阈值		CPU <80%，TPS≈2000
		内存泄露		无
		TPS波动		<5%
	应用资源（2C2G）	MEM		<2G
		CPU最大使用率		<90%
	DB资源(8C8G)	MEM		<8G
		CPU最大使用率		<90%
	缓存（1C1G）	MEM		<80%
		CPU最大使用率		<90%

设计应用——通用设计

- CPU，内存，硬盘



- 用Redis来加速：缓存，分布式锁；注意穿透
- 用Queue来削峰填谷

设计应用——.NET体系

- 利用异步：Demo
- 谨防阻塞：讲个故(事)事(故) Redis阻塞
- 大集合化整为零：讲个故(事)事(故) 内存回收方式
- 避免在后台生成文件操作：讲个故(事)事(故) 后台生成文件
- 复杂方法要比对： Demo
- 让每个API轻巧快速
- 有一颗追求性能的心——注意升级.NET版本
- ……

设计应用——发布

配置：GC方式 工作站方式，服务器方式

CPU 使用率比内存更重要，服务器 GC性能更好。

内存利用率较高而 CPU 使用率相对较低，工作站 GC 性能更高。

发布方式：默认和R2R 首次请求时间和体积不同

	普通模式	R2R模式	AOT模式
大小	29.8 MB	62.2 MB	19.5M
首次请求用时	360ms	90ms	20ms

正确测试

测性能的正确姿势

- 尽量与生产环境一致
- 要有监控，通过监控数据对比发现问题
- 不要打满资源：CPU<90%，内存少于最大值
- 让子弹多飞会，观察内存是有什么不一样

遇到问题借助工具

- dotnet-dump
- dotnet-counters

前人经验也很保贵

- **资源**https://learn.microsoft.com/zh-cn/dotnet/core/diagnostics/

性能优化

减少响应时间：

- 优化&简化流程
- 优化调用链路上的函数
- 把关系数据库操作转成缓存操作
- 用BenchmarkDotNet来检测优化结果

提升TPS：

- 让应用性能是线性的，可以轻松的通过扩容来提升TPS
- 降低性能测中的较高资源

.NET Conf China 2022
开源 · 安全 · 赋能

这里加一张PPT

是为了不忘记show一下《风控性能测试结果》

高性能：

一定不是架构出来的，
但一定是优化出来的。


.NET Conf China
2022
开源 · 安全 · 赋能
THANK!
感谢聆听
欢迎关注我的公众号《桂迹》