

 北京大学计算机学院本科生课程

计算机组成与 系统结构实习



性能评测

北京大学计算机学院系统结构研究所

易江芳

2022-09-19

■ 衡量计算机的指标：性能

- 核武器数值模拟——全面核禁试条约签订后，核武器的数值模拟成为唯一可能进行的全系统试验。美国为了满足核武器库管理的需求，需要每秒运算 10^{16-17} 次的计算机。



具有重要的战略意义，分秒必争，目的就是越快越好

衡量计算机的指标：价格（成本）



计算机越来越亲民，要让更多人买得起，价格很重要

衡量计算机的指标：功耗



无处不在的计算机，电池如何用的久

性能的重要性

- 关键任务应用
- 生命支持系统
- 安全类应用
- 战争环境下的应用
- 个人通信
-

性能评测的不同目的

- 比较性的性能评测和分析
 - 租赁/购买软硬件系统
 - 选择供应商
 - 区分现有系统
 - 评价系统软硬件改进
- 分析性的性能评测和分析
 - 改善系统性能
 - 维护系统（往往是在现有条件下）
 - 设计和实现新的系统

计算机系统性能的多种理解

- “The degree to which a computing system meets the expectations of the people involved with it”
—— Doherty, W.J.
“Scheduling TSS/360 for Responsiveness”
- “Performance...is the effectiveness with which the resources of the host computer system are utilized towards meeting the objectives of the software system”
—— Graham, R.M.
“Performance Prediction”

影响性能的因素

- 算法影响最大
 - 如冒泡排序复杂度为 $O(N*N)$ ，快速排序复杂度为 $O(N\log N)$
- 编译器影响
 - 一般有几倍的差距
- 指令系统
 - 复杂指令如三角函数、FFT、AES等是否硬件实现
- 微结构：IPC（Instructions per cycle）
 - 通过乱序执行、多发射、存储层次等提高IPC
- 主频
 - 受工艺和微结构（流水线）的影响

性能公式（三角铁律）

Matrix Multiply: relative speedup to a Python version (18 core Intel)

Version	Speed-up	Optimization
Python	1	
C	47	Translate to static, compiled language
C with parallel loops	366	Extract parallelism
C with loops & memory optimization	6,727	Organize parallelism and memory access
Intel AVX instructions	62,806	Use domain-specific HW

$$CPU时间 = \frac{秒数}{程序} = \frac{指令数}{程序} \times \frac{时钟数}{指令} \times \frac{秒数}{周期}$$

	指令总数	CPI	时钟周期
算法	X	(X)	
编程语言	X	X	
编译	X	X	
指令系统	X	X	(X)
组成		X	X
实现			X

性能 vs. 主频

CPU型号	频率(GHz)	SPEC_Int	SPEC_Fp	微体系结构	硬件时间
Pentium4 670	3.80	11.5	12.2	Netburst/P4	2005.05
T7600	2.33	14.0	12.5	Core/Core 2 Duo	2006.09
Xeon 5160	3.00	17.5	15.4	Core/ Woodcrest	2007.01
Xeon X5482	3.20	25.3	21.2	Core/Harpertown	2007.11
Core i7-965 EE	3.20	32.1	38.5	Nehalem/Desktop	2008.11
Xeon X5570	3.33	34.5	38.4	Nelamen/Gainestown	2009.03
Xeon X5650	2.67	35.9	51.3	Nehalem/Gulftown	2010.04
Core i3-2100	3.10	34.1	48.0	Sandy Bridge/Desktop	2011.05
Xeon E3-1280	3.50	45.8	58.9	Sandy Bridge	2011.03
Xeon E5-2690	2.90	55.4	89.6	Sandy Bridge-EP	2012.05

- 主频降低了，单核性能大幅度提高（SPECint/fp2006）
 - 性能的提高少量来自于自动并行化，主要是浮点
- 主要通过结构优化提高性能
 - 骨架变大了（马变成了骆驼）：多访存部件、向量化、大队列.....
 - 细节做精了（结合应用的具体优化）

计算机性能评测原则

- 拿程序来测，而不是看个别技术指标（如主频）
 - 是骡子是马，拉出来溜溜：一系列的基准程序
- 不同计算机侧重点不一样，需要不同测试程序
 - 个人PC：单任务关注响应时间
 - 计算中心：多任务关注吞吐率
- 测试程序要有代表性，要足够大，而不是拿个别程序测
 - 基准测试程序套件选取典型应用，能全面综合评价计算机性能。但其属概要测试，不一定能准确反映用户程序的执行性能
- 多个程序时，做归一化和几何平均比较公平
- 测试报告要足够详细，要公开，要可以检查

以航班为例

机 型	载 客 量	续 航 能 力 (英里)	巡 航 速 度 (英里/小时)	运 输 能 力 (人·英里/小时)
波音777	375	4630	610	228,750
波音747	470	4150	610	286,700
协和式飞机	132	4000	1350	178,200
道格拉斯DC-8-50	146	8720	544	79,424

何时使用几何平均值

	A 上的执行时间	B 上的执行时间	根据 A 规格化		根据 B 规格化	
			A	B	A	B
程序 1	1	10	1	10	0.1	1
程序 2	1000	100	1	0.1	10	1
执行时间的算术平均值	500.5	55	1	5.05	5.05	1
执行时间的几何平均值	31.6	31.6	1	1	1	1

算术平均与几何平均

- 算术平均值
 - 算术平均值的权重与特定程序在特定机器上的执行时间成正比，所以权重不仅与该程序的执行频度有关，也与运行该程序的机器特性和输入数据相关。可能导致测试者把自己机器上运行最快的程序的输入量增到最大，以提高计算该程序平均值时的权重，干扰了真实结果
- 几何平均值
 - 小于算术平均值、与参考机无关、强调性能平衡。
 - 可能导致硬件和软件设计者集中精力提高最易提高速度的软件的性能，而不是提高速度最慢的软件的性能。

建模类型

- 结构级模型 (Structural Model)
- 功能级模型 (Functional Model)
 - 流程图、有限状态自动机、并行网络 (例如 Petri网)、队列模型
- 性能分析模型
 - $t = T * IC * CPI$
 - $AMAT = \text{Hit time} + \text{Miss rate} * \text{Miss penalty}$
- 经验模型
 - 回归模型预测
 - 系统剖视

评测程序类型

- 指令混合集
- 合成程序
- 评测程序
- 程序运行踪迹
- 概率负载模型
- 交互式的系统驱动模块
 - 更像是生成器而非静态模型

■ 处理器评测程序、评价指标和平台

❖ 评测程序

- ❖ SPEC 95/2000/2006
- ❖ Dhrystone、Coremark、Media bench、DSP bench

❖ 评测指标

- ❖ 代码密度 (Code Density)
- ❖ 延迟 (Delay - Cycle count)
- ❖ 功耗 (Power & Energy)
- ❖ 响应时间 (Response Time)

❖ 评测平台

- ❖ 各种级别的模拟器、监视器、采样器等
- ❖ 编译器和操作系统



北京大学计算机学院本科生课程

计算机组成与 系统结构实习



基准评测程序

北京大学计算机学院系统结构研究所

刘先华 易江芳

2022-9-19



It is not simply how busy you are,
but why you are busy.
The bee is praised;
the mosquito is swatted.

基准测试程序—benchmark

1 Benchmark类型

- 1.1 单条指令
- 1.2 混合指令
- 1.3 合成的Benchmark程序
- 1.4 Microbenchmarks
- 1.5 程序内核(program kernels)
- 1.6 真实应用程序(real program)

2 Benchmark策略

- 2.1 固定计算量
- 2.2 固定时间
- 2.3 时间和计算量都可变

3 Benchmark程序举例

- 3.1 科学与工程
- 3.2 事务处理
- 3.3 服务器和网络
- 3.4 大数据领域
-

3.1 科学与工程

- 3.1.1 Livermore Loops
- 3.1.2 NAS kernels
- 3.1.3 LINPACK
- 3.1.4 PERFECT club
- 3.1.5 SPEC CPU
- 3.1.6 Whetstone 和 Dhrystone

3.1.1 Livermore Loops

- 由Lawrence Livermore National Lab开发
- 从大量的数值计算程序中提取的24个kernel组成
- MFLOPS为量度
- 很小，容易移植和测试

3.1.2 NAS kernels

- NASA开发
- 7个程序，从计算流体力学程序中抽取
- 包括复数的FFT、矩阵乘、Cholesky分解和高斯消去等
- 使用MFLOPS作为性能量度
- 并行版本NPB(NAS Parallel Benchmark),
免费，使用广泛

3.1.3 LINPACK

- LINPACK
 - LU分解法解 $N \times N$ 的线性方程组
 - TOP 500所用的指标
 - 采用BLAS(Basic Linear-algebra subroutines), 该函数库得到广泛应用
 - (Intel MKL, AMD ACML)

3.1.4 PERFECT club

- 13个完整的Fortran应用
- 包括计算流体力学、化学、物理、工程设计和信号处理
- 目标是测试高性能计算系统的性能
 - 编译器变换，自动并行化
- 报告实际执行时间和CPU时间
- 计算结果验证

3.1.5 SPEC CPU

- 使用最广泛的单CPU性能测试Benchmark
- 及时更新
 - 1989,1992,1995,2000,2006
- 分为若干个组
 - HPG,OSG,GPC
- OSG
 - SPEC CPU2000 (SPECfp2000, SPECint 2000)
- HPG
 - SPEC HPC 2002, SPEC OMP 2001

SPEC2006测试程序套件

SPEC CPU2000 Integer Benchmarks		
400.perlbench	C	PERL Programming Language
401.bzip2	C	Compression
403.gcc	C	C Compiler
429.mcf	C	Combinatorial Optimization
445.gobmk	C	Artificial Intelligence: go
456.hmmer	C	Search Gene Sequence
458.sjeng	C	Artificial Intelligence: chess
462.libquantum	C	Physics: Quantum Computing
464.h264ref	C	Video Compression
471.omnetpp	C++	Discrete Event Simulation
473.astar	C++	Path-finding Algorithms
483.xalancbmk	C++	XML Processing

SPEC CPU2000 Floating Point Benchmarks		
410.bwaves	Fortran	Fluid Dynamics
416.gamess	Fortran	Quantum Chemistry
433.milc	C	Physics: Quantum Chromodynamics
434.zeusmp	Fortran	Physics / CFD
435.gromacs	C/Fortran	Biochemistry/Molecular Dynamics
436.cactusADM	C/Fortran	Physics / General Relativity
437.leslie3d	Fortran	Fluid Dynamics
444.namd	C++	Biology / Molecular Dynamics
447.dealII	C++	Finite Element Analysis
450.soplex	C++	Linear Programming, Optimization
453.povray	C++	Image Ray-tracing
454.calculix	C/Fortran	Structural Mechanics
459.GemsFDTD	Fortran	Computational Electromagnetics
465.tonto	Fortran	Quantum Chemistry
470.lbm	C	Fluid Dynamics
481.wrf	C/Fortran	Weather Prediction
482.sphinx3	C	Speech recognition

3.1.6 Whetstone和Dhrystone

- 合成Benchmark
 - Whetstone : 浮点运算, +, * ...
 - Dhrystone: 整数运算, 字符串复制、比较

3.2 事务处理

- TPC - Transaction Processing Council
 - TPC-C: OLTP(Online Transaction Processing)
 - TPC-H/ TPC-R :决策支持
 - TPC-W:Web为基础的电子商务
- 多用户Benchmark
- 其它:SpecJAppServer2001/2002

3.3 服务器和网络

- SPECWeb 99/2004
 - Web服务器性能
 - 性能量度?

满足一定响应时间和差错率的同时连接数

3.4 大数据领域

- Hibench

- 是由Intel开发的一个针对Hadoop的基准测试工具。它包含有一组Hadoop工作负载的集合,有人工模拟实验环境的工作负载,也有一部分是生产环境的Hadoop应用程序

工作负载	所属领域	类型	实现
Sort	基本 Benchmark	I/O 密集型, CPU 利用率中等	内部实现
WordCount	基本 Benchmark	CPU 密集型	内部实现

(续)

工作负载	所属领域	类型	实现
Tera Sort	基本 Benchmark	Map 和 Shuffle 阶段 CPU 密集, I/O 中等, Reduce 阶段 I/O 密集, CPU 中等	内部实现
Nutch Indexing	搜索引擎	Map 阶段 CPU 密集, Reduce 阶段 I/O 密集, CPU 中等	基于 Nutch
Page Rank	搜索引擎	CPU 密集型	基于 SmartFrog
Bayesian Classification	机器学习	I/O 密集型	基于 Mahout
K-means	机器学习	中心点计算阶段, CPU 密集型 聚类阶段, I/O 密集型	基于 Mahout
DFSIO	HDFS I/O	I/O 密集型	自己实现

云教程中心

MPRC 2022.02

大数据领域

- Berkeley BigDataBench
 - 是随着Spark、Shark的推出,由AMPLab开发的一套大数据基准测试工具。其一部分是基于Hibench的数据集和数据生成器。同时它有一部分数据集是Common Crawl上采样的文档数据集。

工作负载	说明
javasort	数据排序,使用的是 Java 的原生接口
pipesort	数据排序,使用 pipe 接口对数据进行排序
streamsort	数据排序,使用 streaming 接口对数据进行排序
webdatascan	根据设定的一定采样率对原始数据进行一轮采样,使用 GenericMRLoader 采样器
webdatasort	设定采样率为 100%,对原始数据进行三轮迭代的采样。同样,使用的采样器是 GenericMRLoader
monsterQuery	设定一定的采样率对数据采样,进行三轮迭代,使用的采样器是 GenericMRLoader
maxent	设定一定采样率进行八轮迭代,使用的采样器是 GenericMRLoader

云教程中心

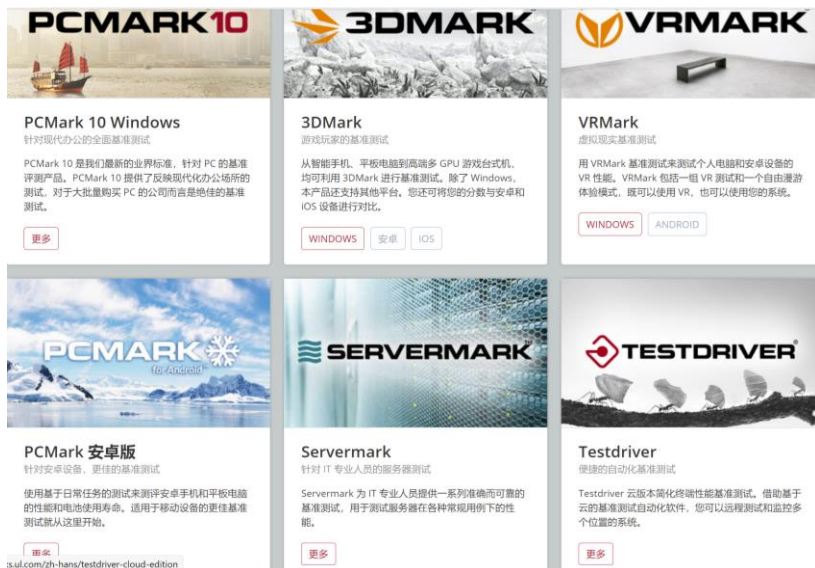
市场上主流的评测软件

- Futuremark (目前已经并入UL)

- PCmark
- 3Dmark
- VRmark



- <https://benchmarks.ul.com/zh-hans/>



4. Benchmark方面的研究工作

- 4.1 Benchmark负载分析和简化
- 4.2 分析工具对比

Benchmark负载分析与简化

- Benchmark在实际执行时，由于模拟速度很慢，可能需要好几个月才能完成模拟
- 需要对Benchmark的进行负载分析（抓程序热点），并合理简化（提高模拟效率）
- 对运行过程进行监控
- 记录被测系统的状态变化



■ Benchmark负载分析与简化

- 负载分析方法

- 插桩

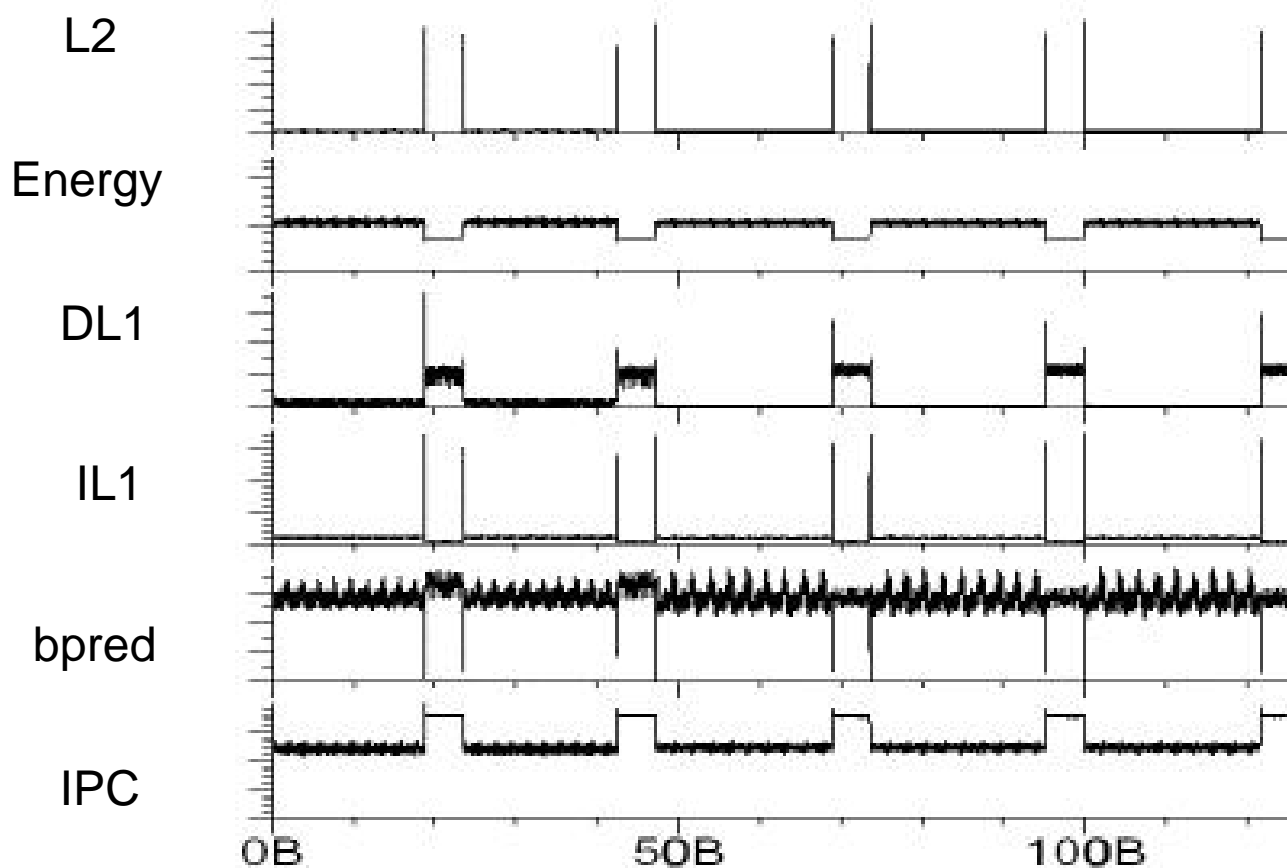
- 通过插入探针指令收集性能信息
 - 静态插桩: Gprof
 - 动态插桩: DynInst、Dtrace
 - 无需硬件支持

- 采样

- 利用操作系统的中断机制周期性地记录程序运行过程中的性能信息
 - 基于时间采样
 - 基于事件采样: VTune Amplifier、Perf、Oprofile
 - 能够收集程序与硬件交互的性能信息、无需添加额外指令、数据较为精确

Benchmark负载分析与简化

- 用采样法进行简化
- 对并行程序的情况会更加复杂(必须保留通信)



分析工具对比

	Intel VTune Amplifier	Gprof	Perf	OProfile
是否开源	否	是	是	是
硬件性能事件分析	支持	不支持	支持	支持
负载分析方式	基于事件/时间采样	静态插桩	基于事件/时间采样	基于事件/时间采样
其他特点	仅支持X86体系结构	gcc内置工具	Linux内核自带工具， 面向进程采样，适合对单个程序进行性能分析	Linux内核自带工具， 全系统采样，适合对系统整体进行分析

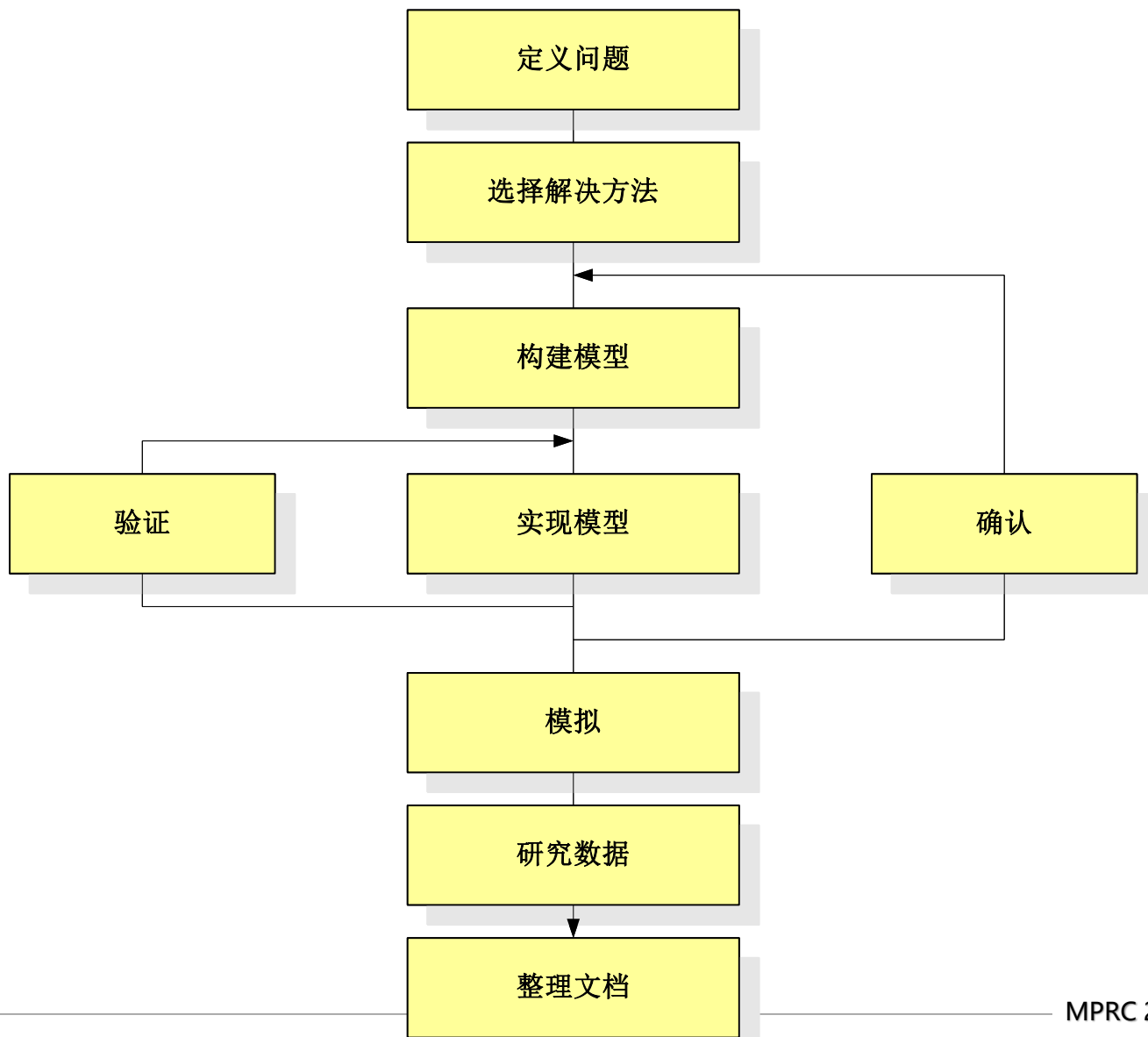
总结

- Benchmark是重要的性能测试工具
- 性能评测有不同的目标，因此有不同种类的Benchmark和不同的Benchmark策略
- SPEC CPU2000是处理器性能评测中最通用的benchmark之一
- 研制Benchmark，除了技术因素外，还应重视运行规则、辅助工具和发布审查机制



欢迎提问

性能评测的方法：模拟及其流程



性能分析关键工作

1. 声明目标、确定系统需求
2. 明确系统提供的服务及输出
3. 选择合适的评价指标
4. 列出相关参数
5. 确定要观察、研究的因素
6. 选择评测的技术路线
7. 确定负载
8. 做好实验设计
9. 分析和理解数据
10. 做好数据表达和解释

Maybe ... goto 01😊