基准测试程序
(Benchmarks)

詹剑锋

中科院计算所

2013年6月17日

内容

- **什么是Benchmark?**
- 常见Benchmark
  - 如何评价高性能计算机
  - 如何评价处理器？
  - 如何评价OLTP系统？
  - 文件系统/数据库？

Why benchmarking?

- “If you cannot measure it, you cannot improve it”.
  - Lord Kelvin

为什么第一讲要讲Benchmark？

- Benchmark是研究问题的Setup。
- 可以借助Benchmark实现solution。
- 评价和比较方法好坏的基准。

- **选择Benchmark的代表性和多样性直接决定研究的深度**
  - **副产品:发表论文的水平。**

什么是基准测试程序？

- 基准测试程序指一组(性能)测试程序
  - 它能刻画应用负载的计算处理、数据移动等行为特征
  - 测试和预测系统的性能
  - 对不同的系统的优缺点给出评价。

合理基准测试程序应具备特点

- 应用开发时,考虑并充分利用了目标机器的系统结构特性
- 负载应具有充分的多样性，能够覆盖一定的目标应用范围
- 程序采用最新的算法和实现技术

C. Bienia *et al.* The PARSEC benchmark suite: characterization and architectural implications. In Proc. of PACT'08.

工业界标准组织

- Standard Performance Evaluation Corporation (SPEC)
- Transaction Processing Performance Council (TPC)
- Business Applications Performance Corporation (BAPCo)
  - personal computers based on popular software applications and operating systems
- Embedded Microprocessor Benchmark Consortium (EEMBC)

基准测试程序分类（1）

- Toy benchmarks
- Micro benchmarks: designed to measure the performance of a very small and specific piece of code.
  - Processing micro benchmarks
  - Local memory micro benchmarks
  - Input-output micro benchmarks
  - Communication micro benchmarks
  - Synchronization micro benchmarks

基准测试程序分类（2）

- Kernels
  - contains key codes
  - normally abstracted from actual program
  - popular kernel: Livermore loop
  - Linpack benchmark

基准测试程序分类（3）

- Synthetic benchmarks
  - Synthesize the diverse range of characteristics of workloads
  - 抓不住重点!
- Application benchmarks
  - top-down method
  - select a suite of representative programs.
  - the best programs using for benchmarking are real applications that the user employs regularly or simply applications that are typical.

常见评价指标

- **Instructions per second**
  - **Linpack:** MFLOPS
- **Performance per watt**
- **IOPS**
  - Input/Output Operations Per Second, pronounced i-ops

内容

- 什么是Benchmark?
- **常见Benchmark**
  - 如何评价高性能计算机
  - 如何评价处理器？
  - 如何评价OLTP系统？
  - 文件系统评价。
- 数据中心Benchmark
- 讨论问题

Linpack

- **LINPACK Benchmarks** 用来度量系统的浮点计算能力。
- 对于大规模分布式内存系统，High Performance Linpack的运行结果, 是Top 500排名的理论依据。
- millions of floating point operations per second (MFLOPS).

Linpack干了什么？

- Introduced by Jack Dongarra,

- they measure how fast a computer solves a dense N by N system of linear equations Ax = b, which is a common task in engineering.

问题

- Linpack适合用于评价Data center 吗？
- 为什么？

Green 500

- The Green500 list ranks computers from the TOP500 list of supercomputers in terms of energy efficiency.

- Typically measured as LINPACK FLOPS per watt

Graph 500

- **Graph 500 Benchmark 1 ("Search")**
  - Contributors: David A. Bader (Georgia Institute of Technology), Jonathan Berry (Sandia National Laboratories), Simon Kahan (Pacific Northwest National Laboratory and University of Washington), Richard Murphy (Sandia National Laboratories), E. Jason Riedy (Georgia Institute of Technology), and Jeremiah Willcock (Indiana University).

Graph Benchmarks的组成

- 可扩展的数据生成器
  - 生成edge tuples (start vertex, end vertex) for each edge.
- The first kernel constructs an *undirected* graph in a format usable by all subsequent kernels.
  - No subsequent modifications are permitted to benefit specific kernels.
- The second kernel performs a breadth-first search of the graph. Both kernels are timed.

输入大小定义的六类规模的问题

- toy
  - 17GB or around 10^{10} bytes, which we also call level 10,
- mini
  - 140GB (10^{11} bytes, level 11),
- small
  - 1TB (10^{12} bytes, level 12),
- medium
  - 17TB (10^{13} bytes, level 13),
- large
  - 140TB (10^{14} bytes, level 14), and
- huge
  - 1.1PB (10^{15} bytes, level 15).

评价指标

- traversed edges per second (TEPS).
- We measure TEPS through the benchmarking of kernel 2 as follows.
  - $time_{K2}(n)$: the measured execution time for kernel 2.
  - *m* be the number of input edge tuples within the component traversed by the search, counting any multiple edges and self-loops.
- We define the normalized performance rate (number of edge traversals per second) as:
- $TEPS(n) = m / time_{K2}(n)$

Graph 500适合评价数据中心？

- 可以吗？
- 为什么？

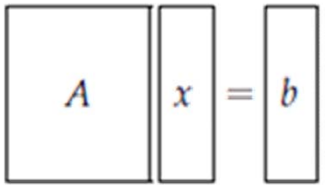
HPCC（HPC Challenge）

- examine the performance of HPC architectures using kernels with **more challenging memory access patterns** than just the High Performance Linpack (HPL) benchmark used in the Top500 list.
  -- Jack Dongarra

- To provide benchmarks that bound the performance of many real applications as a function of memory access characteristics e.g., spatial and temporal locality.

测试程序描述

HPL

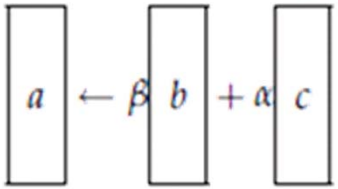


Compute x from the system of linear equations $Ax = b$.

DGEMM

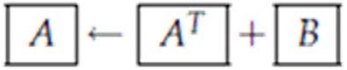
Compute update to matrix C with a product of matrices A and B.

STREAM



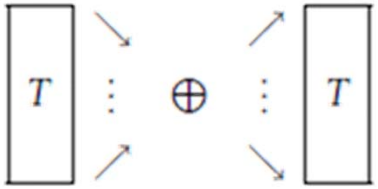
Perform simple operations on vectors a, b, and c.

PTRANS



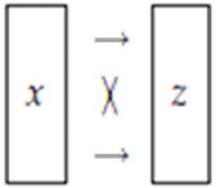
Compute update to matrix A with a sum of its transpose and another matrix B.

RandomAccess



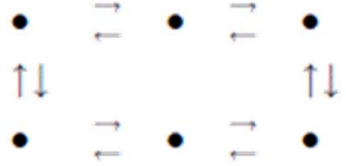
Perform integer update of random vector T locations using pseudo-random sequence.

FFT



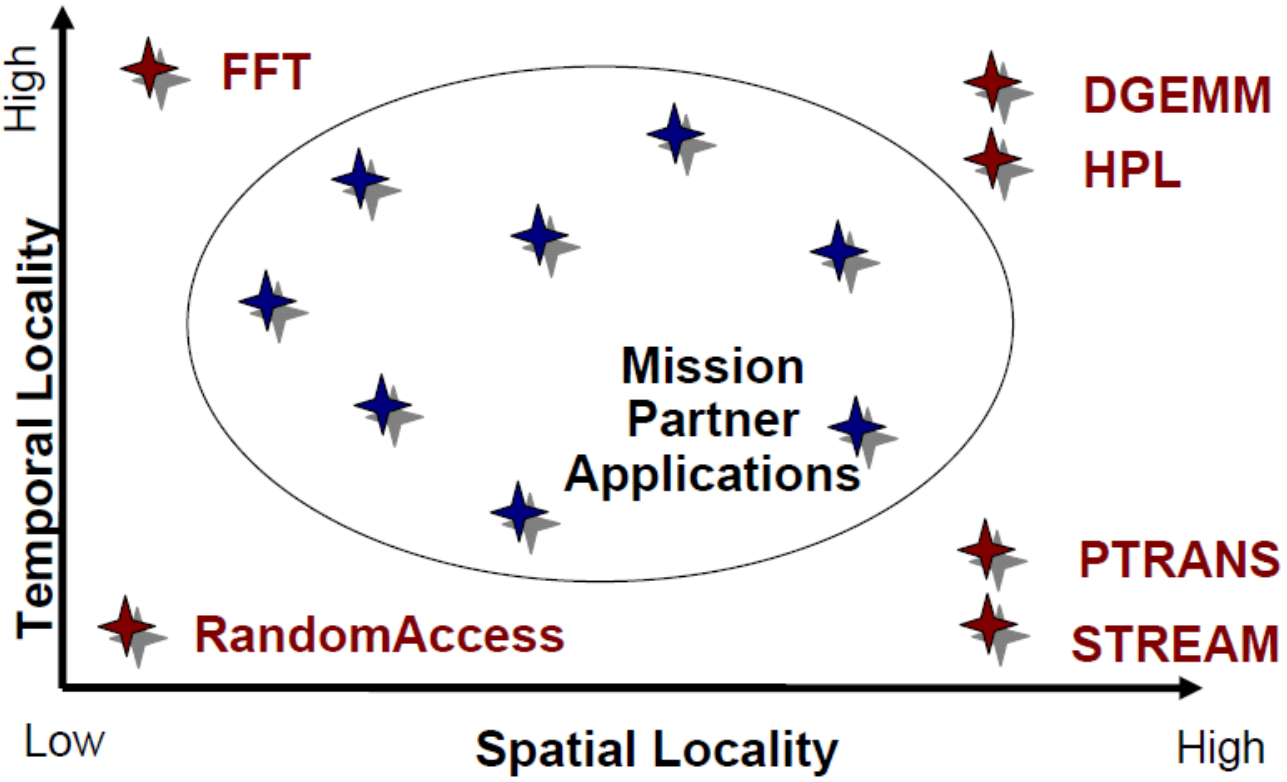
Compute vector z to be the Fast Fourier Transform (FFT) of vector x.

b_eff



Perform ping-pong and various communication ring exchanges.

测试集的Locality



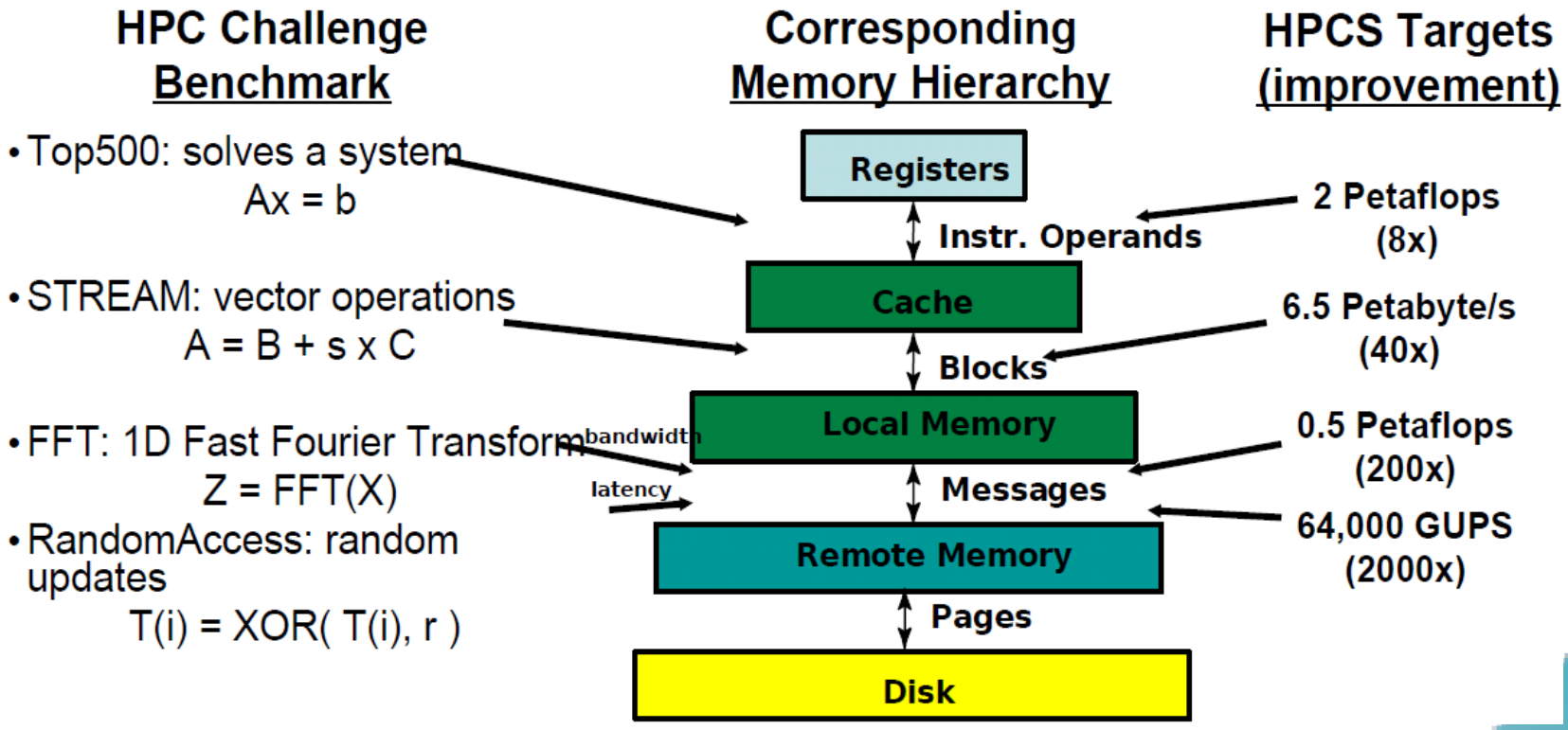
Locality of References (Wikipedia)

- Temporal locality
  - A resource that is referenced at one point in time will be referenced again sometime in the near future.
- Spatial locality
  - Likelihood of referencing a resource is higher if a resource near it was just referenced.

An Operational Definition of Locality

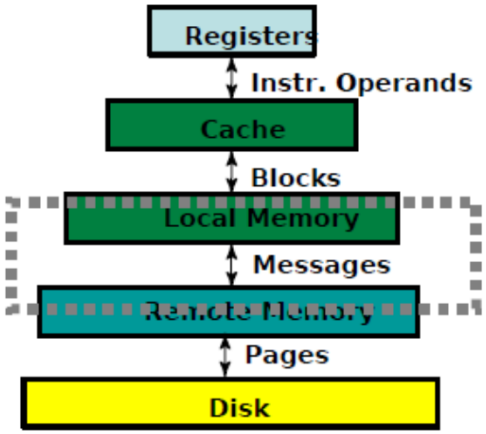
- Good temporal locality ⇒cache miss traffic decreases fast when cache size increases.
- Good spatial locality ⇒cache miss traffic does not increase much when line size increases.

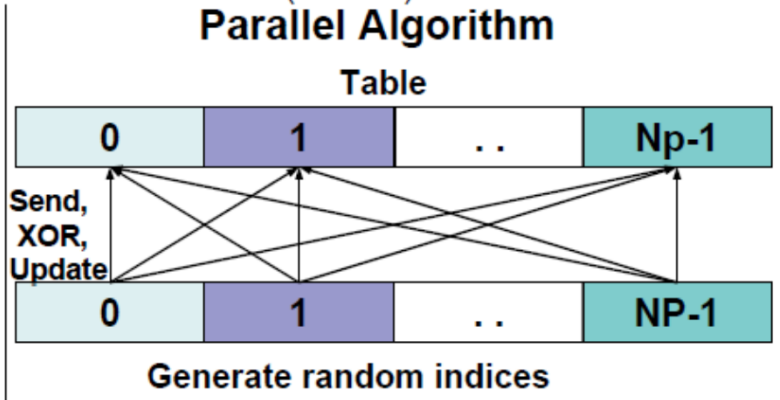
测试集对Memory hierarchy的考验



HPCC Tests - RandomAccess

- Randomly updates N element table of unsigned integers
- Each processor generates indices, sends to all other processors, performs XOR
- Results are reported in Giga Updates Per Second (GUPS)





- Randomly updates memory (requires all-to-all communication)
- Stresses interprocessor communication of *small* messages
- DARPA HPCS goal: 64,000 GUPS (2000x over current best)

HPCC Tests – b_eff

- latency (send an 8-byte message from one node to another)
- bandwidth (message size divided by the time it takes to transmit a 2,000,000 byte message) of network communication using basic MPI routines.
- non-simultanous (ping-pong benchmark) and simultaneous communication (random and natural ring pattern)
  - covers two extreme levels of contention (no contention and contention) that might occur in real application.

问题： 如何提高top 500排名 ？

- Linpack,
- Green 500
- Graph 500
- HPCC看起来更全面，为什么没有取代Linpack?

内容

- 什么是Benchmark?
- 常见Benchmark
  - 如何评价高性能计算机
  - 如何评价处理器？
  - 如何评价OLTP系统？
  - 文件系统评价？
- 数据中心Benchmark
- 讨论问题

PARSEC

- Princeton Application Repository for Shared-Memory Computers（PARSEC）是用于测试CMPs(Chip-Multiprocessors)的多种应用程序的集合。

- 负载和工作集覆盖了很多领域

不同于其他benchmark的特点

- 多线程
- Emerging workloads
- 多样性
- Employ State-of-Art Techniques
  - Visual applications for example have started to increasingly integrate physics simulations to generate more realistic animations
- 支持研究

SPLASH-2

- a suite composed of multi-threaded applications.
- 偏向HPC 和图形程序
- 不包括pipeline 等并行编程模型

组成

- framework
  - 用户工具
  - 文档
  - 配置文件
- package
  - 源代码
  - 输入集

配置文件种类

- System configurations：配置系统，如源代码位置
- Build configurations：生成负载相关的配置
- Run configurations：配置执行benchmark

测试程序

Program	Application Domain	Parallelization		Working Set	Data Usage	
		Model	Granularity		Sharing	Exchange
blackscholes	Financial Analysis	data-parallel	coarse	small	low	low
bodytrack	Computer Vision	data-parallel	medium	medium	high	medium
canneal	Engineering	unstructured	fine	unbounded	high	high
dedup	Enterprise Storage	pipeline	medium	unbounded	high	high
facesim	Animation	data-parallel	coarse	large	low	medium
ferret	Similarity Search	pipeline	medium	unbounded	high	high
fluidanimate	Animation	data-parallel	fine	large	low	medium
freqmine	Data Mining	data-parallel	medium	unbounded	high	medium
raytrace	Rendering	data-parallel	medium	unbounded	high	low
streamcluster	Data Mining	data-parallel	medium	medium	low	medium
swaptions	Financial Analysis	data-parallel	coarse	medium	low	low
vips	Media Processing	data-parallel	coarse	medium	low	medium
x264	Media Processing	pipeline	coarse	medium	high	high

输入集

- Test:小规模输入集，测试基本功能
- Simdev:小规模输入集，guarantees basic program behavior similar to the real behavior, intended for simulator test and development.
- simsmall, simmedium and simlarge:大小不同的输入集，用于模拟器的微体系结构研究
- Native:大规模的真实输入

a breakdown of instructions and synchronization primitives of the simlarge input set

Program	Problem Size	Instructions (Billions)				Synchronization Primitives		
		Total	FLOPS	Reads	Writes	Locks	Barriers	Conditions
blackscholes	65,536 options	2.67	1.14	0.68	0.19	0	8	0
bodytrack	4 frames, 4,000 particles	14.03	4.22	3.63	0.95	114,621	619	2,042
canneal	400,000 elements	7.33	0.48	1.94	0.89	34	0	0
dedup	184 MB data	37.1	0	11.71	3.13	158,979	0	1,619
facesim	1 frame, 372,126 tetrahedra	29.90	9.10	10.05	4.29	14,541	0	3,137
ferret	256 queries, 34,973 images	23.97	4.51	7.49	1.18	345,778	0	1255
fluidanimate	5 frames, 300,000 particles	14.06	2.49	4.80	1.15	17,771,909	0	0
freqmine	990,000 transactions	33.45	0.00	11.31	5.24	990,025	0	0
streamcluster	16,384 points per block, 1 block	22.12	11.6	9.42	0.06	191	129,600	127
swaptions	64 swaptions, 20,000 simulations	14.11	2.62	5.08	1.16	23	0	0
vips	1 image, 2662×5500 pixels	31.21	4.79	6.71	1.63	33,586	0	6,361
x264	128 frames, 640×360 pixels	32.43	8.76	9.01	3.11	16,767	0	1,056

多线程

Program	Pthreads	OpenMP	TBB
blackscholes	✓	✓	✓
bodytrack	✓	✓	✓
canneal	✓		
dedup	✓		
facesim	✓		
ferret	✓		
fluidanimate	✓		✓
freqmine		✓	
raytrace	✓		
streamcluster	✓		✓
swaptions	✓		✓
vips	✓		
x264	✓		

the Intel Threading Building Blocks (TBB)

问题？

- PARSEC与SPECCPU 2006 的不同？

内容

- 什么是Benchmark?
- 常见Benchmark
  - 如何评价高性能计算机
  - 如何评价处理器？
  - **如何评价OLTP系统？**
  - 文件系统评价？
- 数据中心Benchmark
- 讨论问题

TPC-C

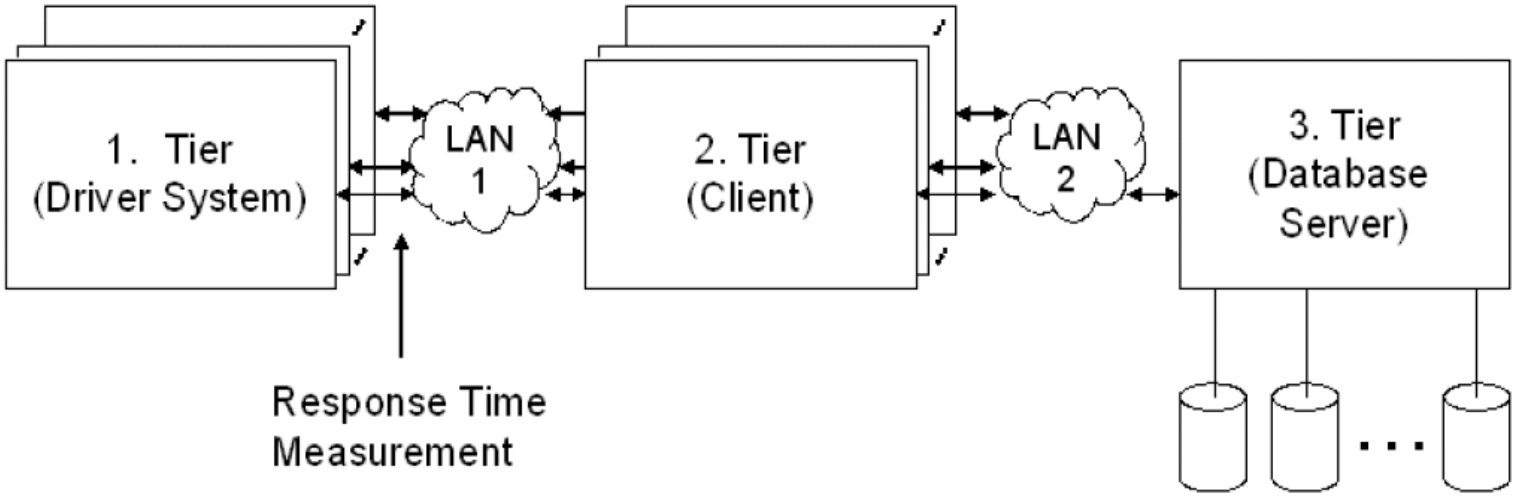
- 一种旨在衡量联机事务处理（OLTP）系统性能与可伸缩性的行业标准基准测试项目。
- 1992年发布
- TPC-A 的替代品

TPC-C 特性

- 同时处理多种类型transaction
- 支持多在线终端session
- 中等的系统和应用的执行时间
- 存在来自硬盘的输入输出
- 数据库满足ACID（Atomicity, Consistency, Isolation and Durability）
- 数据访问不均匀
- 数据库包含多个不同大小、属性、关系的表
- 数据访问和更新存在竞争

TPC-C系统构成

- The typical TPC-C system is designed in 3 tiers
  - 1. Tier:Driver System
  - 2. Tier:Client
  - 3. Tier:Database Server



TPC-C database

Structure of the TPC-C database	
Table	Number of entries
Warehouse	n (specified in a measurement)
Item	100,000
Stock	n x 100,000
District	n x 10
Customer	3,000 per district, 30,000 per warehouse
Order	number of customers (initial value)
New order	30% of the orders (initial value)
Order line	approx. 10 per order
History	number of customers (initial value)

Transactions

- New-Order：客户输入一笔新的订货交易；
- Payment:更新客户账户余额以反映其支付状况
- Delivery:发货(模拟批处理交易);
- Order-Status:查询客户最近交易的状态；
- Stock-Level:查询仓库库存状况，以便能够及时补货
- 对于前四种类型的交易，要求响应时间在5秒以内；对于库存状况查询交易，要求响应时间在20秒以内。

transaction 分布

TPC-C transactions and required distribution	
Name of transaction	Share of all transactions
New order	≤ 45%
Payment	≥ 43%
Order status	≥ 4%
Delivery	≥ 4% (batch transaction)
Stock level	≥ 4%

性能指标

TPC-C metrics		
Throughput	Cost of ownership	Report date
tpmC	$/tpmC	-

- 吞吐量指标(Throughput，简称tpmC)
  - the number of processed new-order transactions per minute(至少运行2个小时）。
- 性价比(Price/Performance，简称Price/tpmC)
  - 软件成本
  - 硬件成本
  - 3年的维护费用

问题？

- TPC-C是如何定义用户行为的？

内容

- 什么是Benchmark?
- 常见Benchmark
  - 如何评价高性能计算机
  - 如何评价处理器？
  - 如何评价Web服务？
  - 如何评价文件系统和数据库？
- 数据中心Benchmark
- 讨论问题

如何评价文件系统和数据库？

- 有哪些I/O访问模式？有哪些代表性的I/O Benchmark？为什么要运行真实的应用。

- 如何评价NoSQL

中科院计算所

谢谢