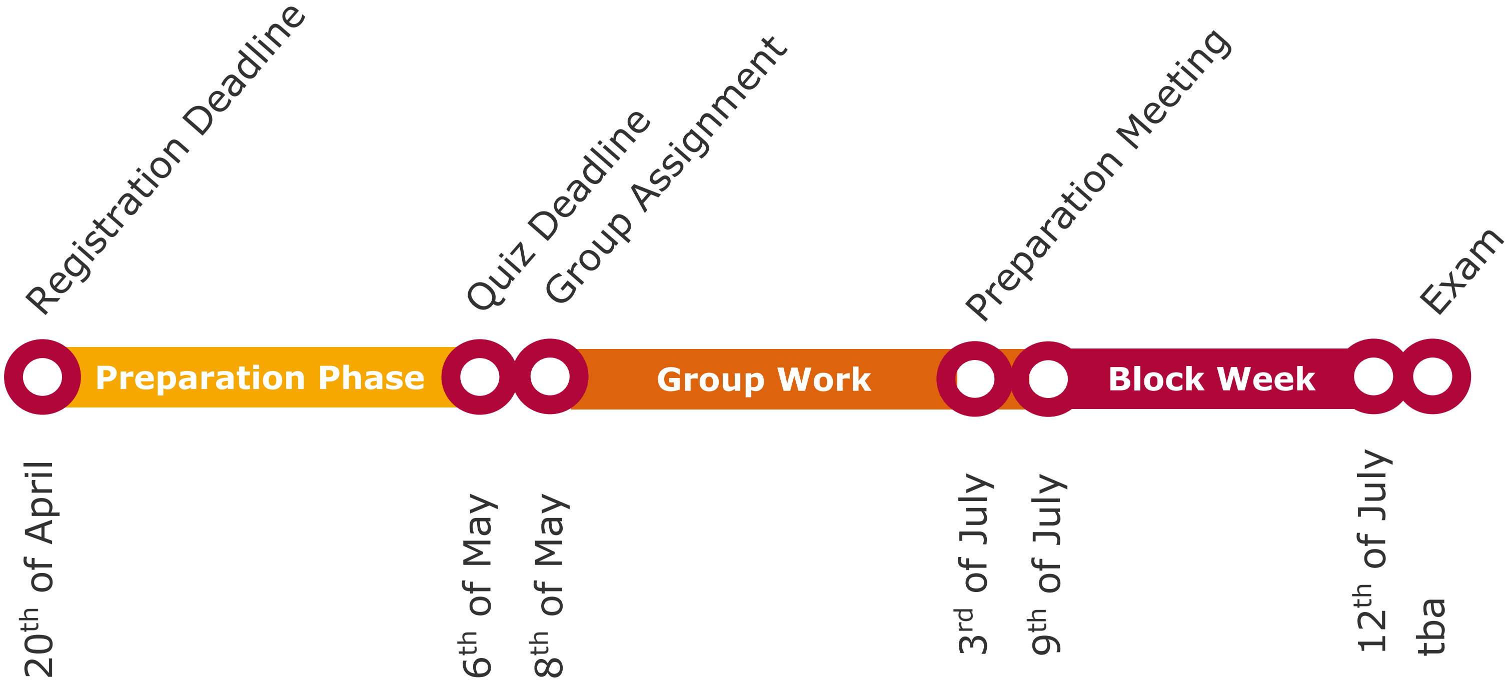
HPI Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Trends and Concepts in the Software Industry I

Schedule



Registration Deadline
20th of April

Preparation Phase

Quiz Deadline
6th of May

Group Assignment
8th of May

Group Work

Preparation Meeting
3rd of July

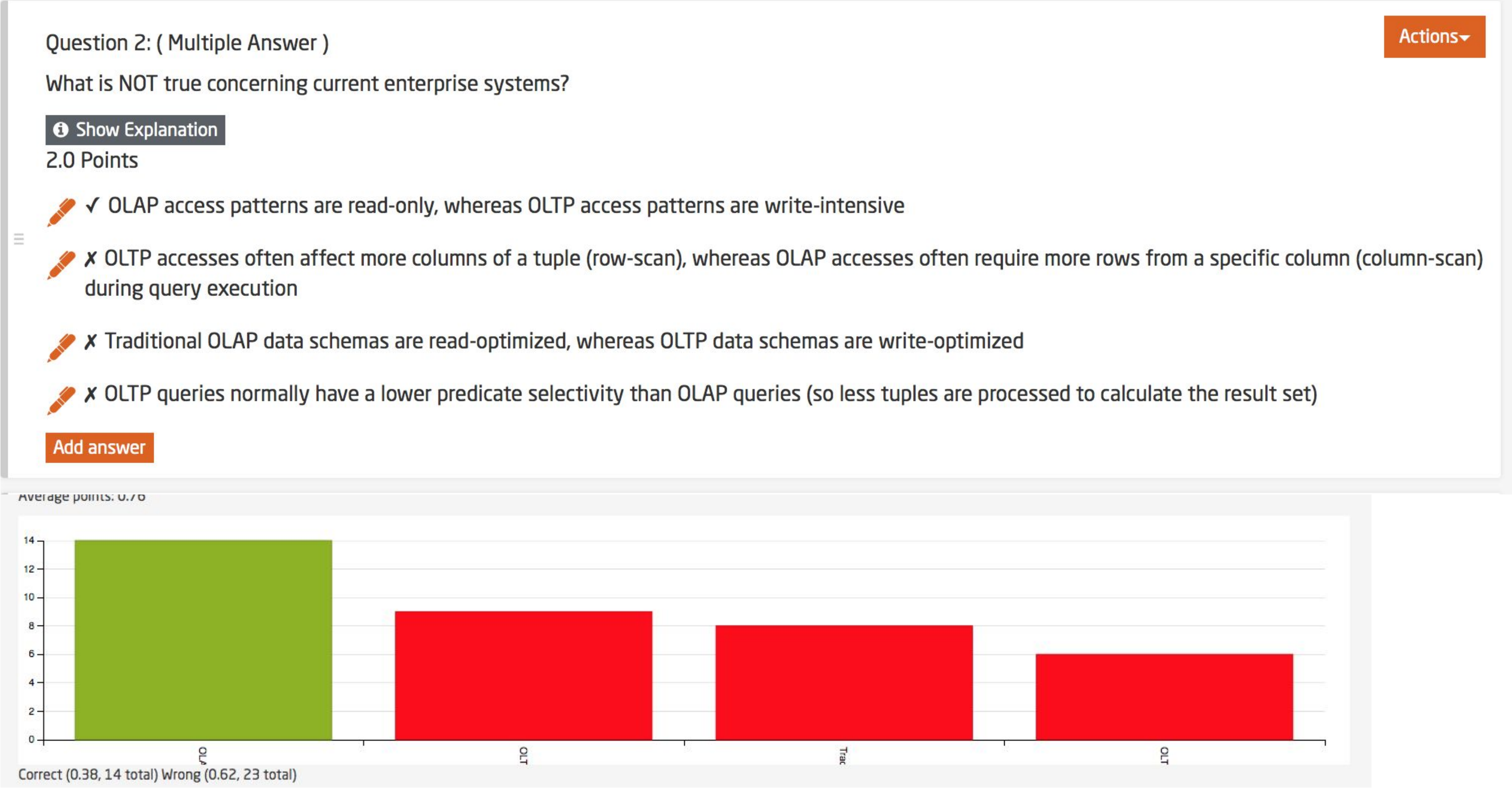
9th of July

Block Week

12th of July

Exam
tba

Results of Preparation Quiz



Question 2: (Multiple Answer)

What is NOT true concerning current enterprise systems?

Show Explanation

2.0 Points

- ✓ OLAP access patterns are read-only, whereas OLTP access patterns are write-intensive
- ✗ OLTP accesses often affect more columns of a tuple (row-scan), whereas OLAP accesses often require more rows from a specific column (column-scan) during query execution
- ✗ Traditional OLAP data schemas are read-optimized, whereas OLTP data schemas are write-optimized
- ✗ OLTP queries normally have a lower predicate selectivity than OLAP queries (so less tuples are processed to calculate the result set)

Average points: 0.76

Correct (0.38, 14 total) Wrong (0.62, 23 total)

Results of Preparation Quiz

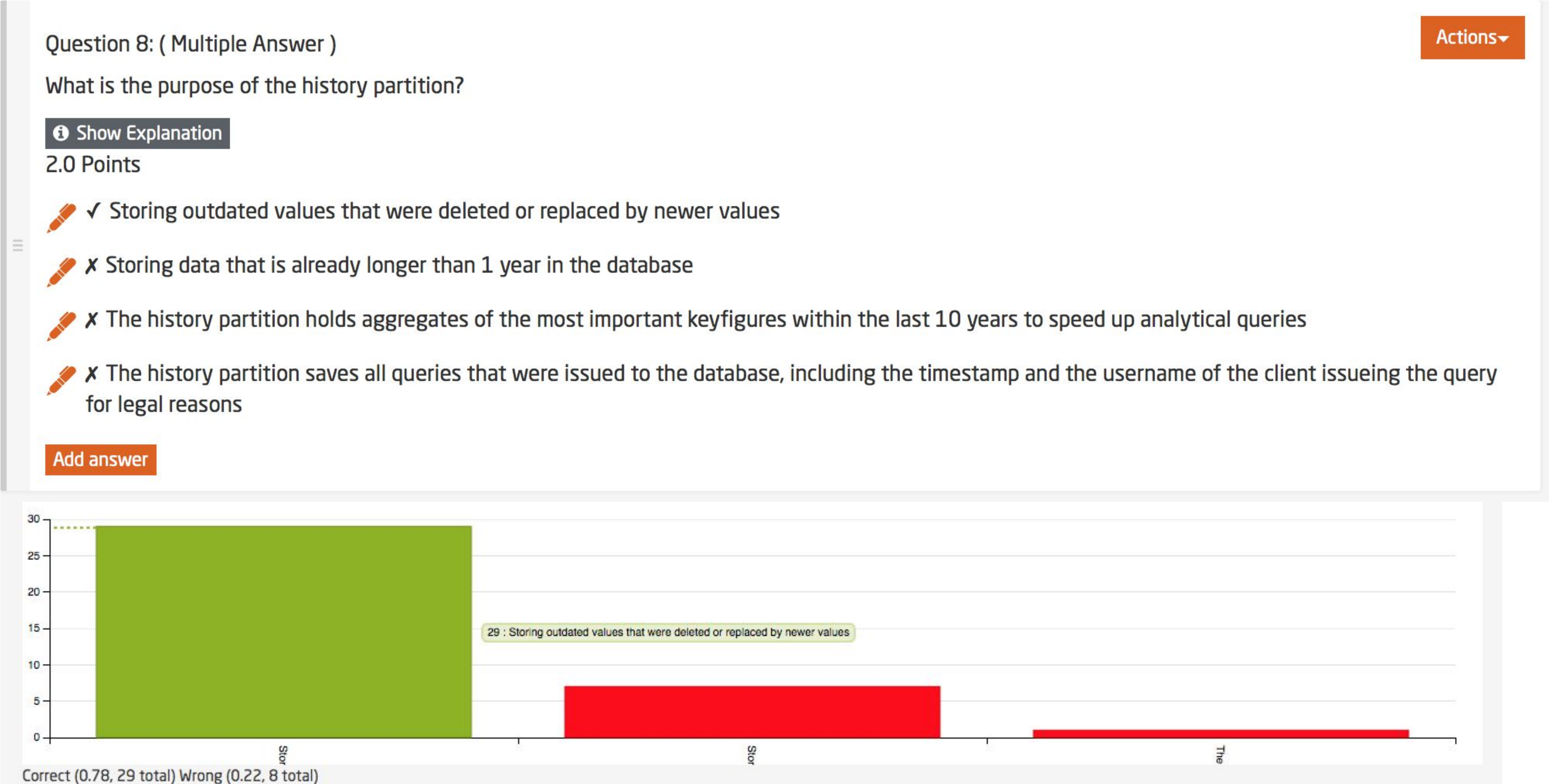
- ✓ OLAP access patterns are **read-only**, whereas OLTP access patterns are write-intensive

- ✗ OLTP accesses often affect more columns of a tuple (row-scan), whereas OLAP accesses often require more rows from a specific column (column-scan) during query execution

- ✗ Traditional OLAP data schemas are read-optimized, whereas OLTP data schemas are write-optimized

- ✗ OLTP queries normally have a **lower** predicate selectivity than OLAP queries (so less tuples are processed to calculate the result set)

Results of Preparation Quiz



Question 8: (Multiple Answer)

What is the purpose of the history partition?

2.0 Points

- ✓ Storing outdated values that were deleted or replaced by newer values
- ✗ Storing data that is already longer than 1 year in the database
- ✗ The history partition holds aggregates of the most important keyfigures within the last 10 years to speed up analytical queries
- ✗ The history partition saves all queries that were issued to the database, including the timestamp and the username of the client issueing the query for legal reasons

Add answer
30
25
20
15
10
5
0
29 : Storing outdated values that were deleted or replaced by newer values
Stor
Stor
The

Correct (0.78, 29 total) Wrong (0.22, 8 total)

Results of Preparation Quiz

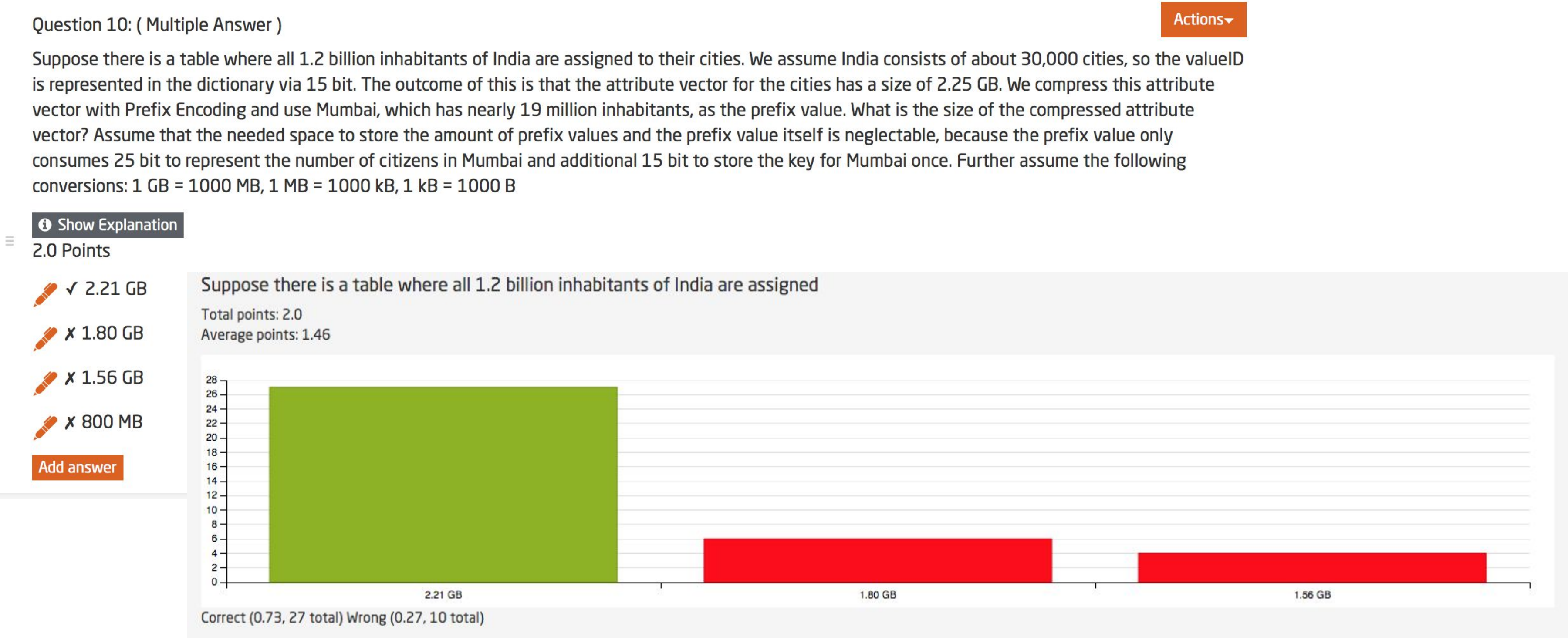
- ✓ Storing outdated values that were deleted or replaced by newer values

- ✗ Storing data that is already longer than 1 year in the database
  ➔ (concept: actual / historical)

- ✗ The history partition holds aggregates of the most important keyfigures within the last 10 years to speed up analytical queries

- ✗ The history partition saves all queries that were issued to the database, including the timestamp and the username of the client issueing the query for legal reasons

Results of Preparation Quiz



Actions▾

Question 10: (Multiple Answer)

Suppose there is a table where all 1.2 billion inhabitants of India are assigned to their cities. We assume India consists of about 30,000 cities, so the valueID is represented in the dictionary via 15 bit. The outcome of this is that the attribute vector for the cities has a size of 2.25 GB. We compress this attribute vector with Prefix Encoding and use Mumbai, which has nearly 19 million inhabitants, as the prefix value. What is the size of the compressed attribute vector? Assume that the needed space to store the amount of prefix values and the prefix value itself is neglectable, because the prefix value only consumes 25 bit to represent the number of citizens in Mumbai and additional 15 bit to store the key for Mumbai once. Further assume the following conversions: 1 GB = 1000 MB, 1 MB = 1000 kB, 1 kB = 1000 B

Show Explanation

2.0 Points

- ✓ 2.21 GB
- ✗ 1.80 GB
- ✗ 1.56 GB
- ✗ 800 MB

Add answer

Suppose there is a table where all 1.2 billion inhabitants of India are assigned

Total points: 2.0

Average points: 1.46

Correct (0.73, 27 total) Wrong (0.27, 10 total)

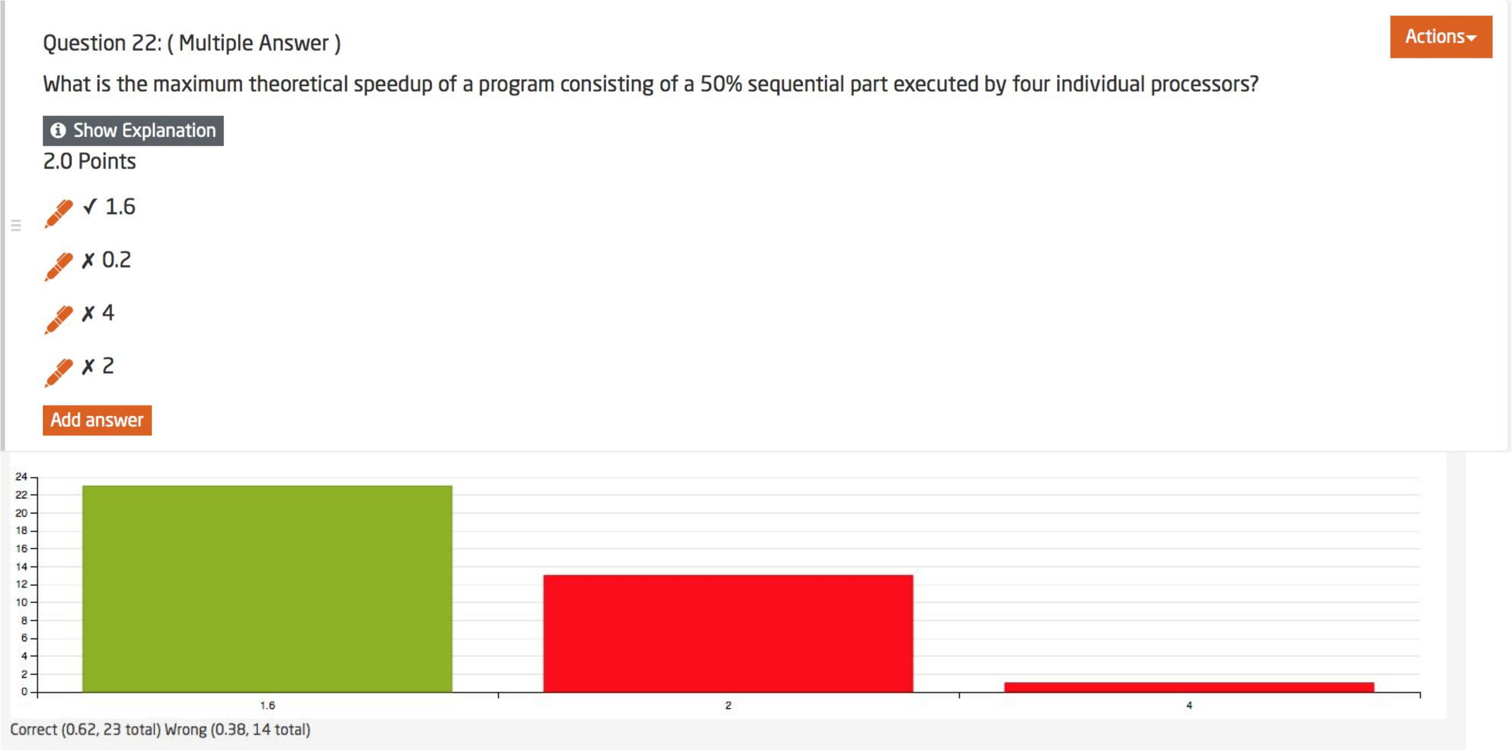
Results of Preparation Quiz

Mumbai: 19 million inhabitants, 1.2 billion in India

15 bit for valueID,
occurences: log2(19 mil.) = 25 bit
Remaining values: 1,181 million * 15 bit = 17,715,000,000 bit.

Total size: 15 bit + 25 bit +17,715,000,000 bit = 17,715,000,040 bit
→ about 2,21 Gbyte

Results of Preparation Quiz



Question 22: (Multiple Answer)

What is the maximum theoretical speedup of a program consisting of a 50% sequential part executed by four individual processors?

2.0 Points

✓ 1.6

✗ 0.2

✗ 4

✗ 2

Correct (0.62, 23 total) Wrong (0.38, 14 total)

Results of Preparation Quiz

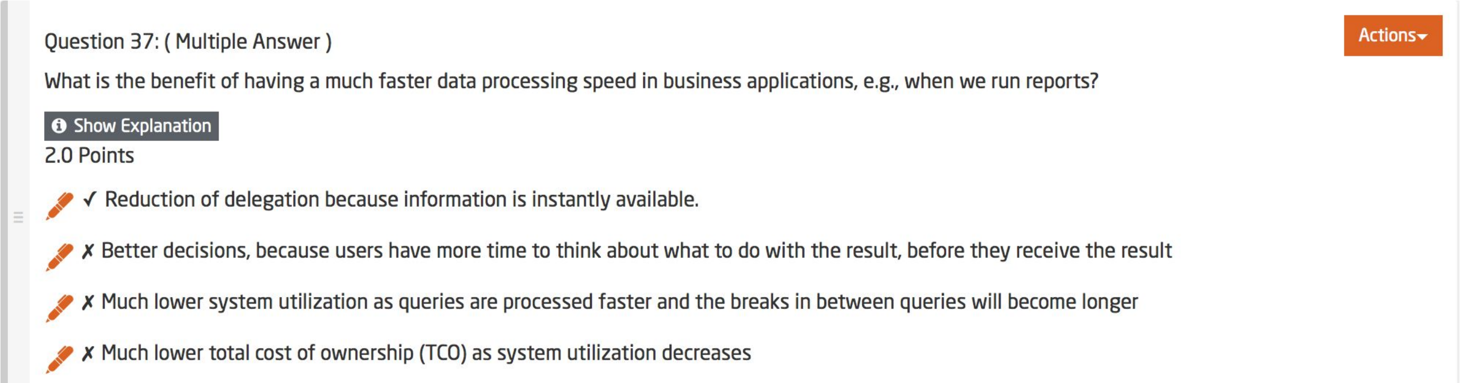
The calculation is as follows:

$$\frac{1}{(s+\frac{p}{N})}=\frac{1}{(0.5+\frac{0.5}{4})}=1.6$$

where s is the serial part of the program, p (p = 1-s) the parallel part and N is the number of used cores.

- ✓ 1.6
- ✗ 0.2
- ✗ 4
- ✗ 2

Results of Preparation Quiz

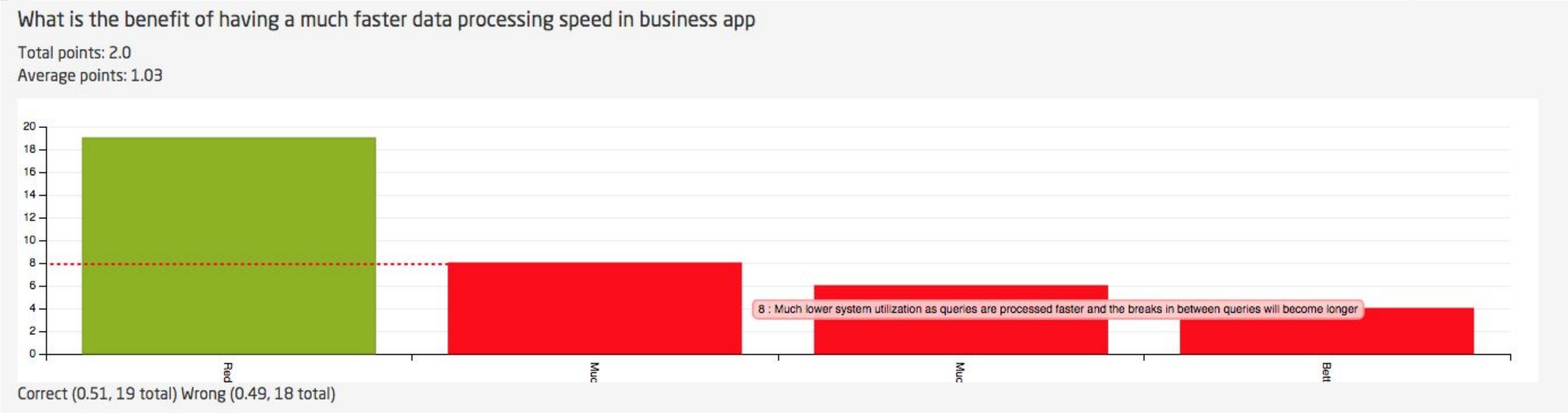


Question 37: (Multiple Answer)

What is the benefit of having a much faster data processing speed in business applications, e.g., when we run reports?

2.0 Points

- ✓ Reduction of delegation because information is instantly available.
- ✗ Better decisions, because users have more time to think about what to do with the result, before they receive the result
- ✗ Much lower system utilization as queries are processed faster and the breaks in between queries will become longer
- ✗ Much lower total cost of ownership (TCO) as system utilization decreases



What is the benefit of having a much faster data processing speed in business app

Total points: 2.0
Average points: 1.03

Correct (0.51, 19 total) Wrong (0.49, 18 total)

Group Phase

- Preparation of interactive group part
  - □ Teams of 6 to 8 students guided by WiMis
  - □ Regular meetings
  - □ Team assignment: 8th of May

- Hands on experiments
  - □ Familiarization with existing research
  - □ Implementation part in C/C++
  - □ Evaluation of the results
  - □ Presentation in the block week (~30 minutes)

About Speed-Demons and Brainiacs Hardware Optimizations on modern CPUs

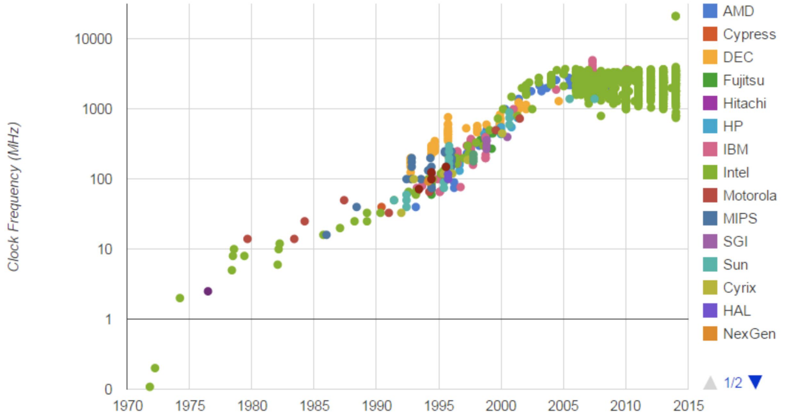
Motivation

While important in the past, clock frequency has become less and less important as a measure of CPU performance. To optimize for modern CPUs, more and more factors have to be taken into account.

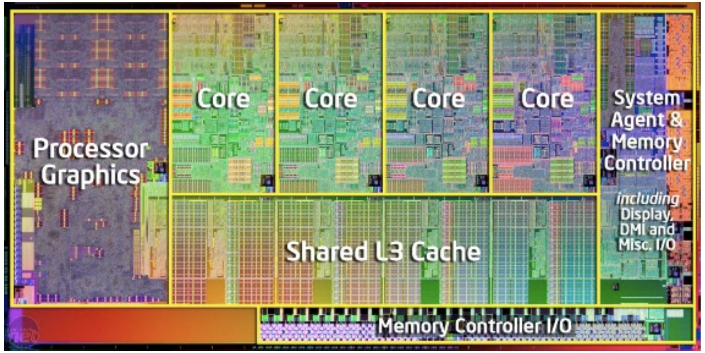
Research Questions

How do hardware features such as Hyperthreading, Prefetching, Out-of-Order Execution, Memory Bandwidth, Execution Units, and others influence the effective performance of a CPU?

How do they differ between CPU generations?



http://cpudb.stanford.edu/visualize

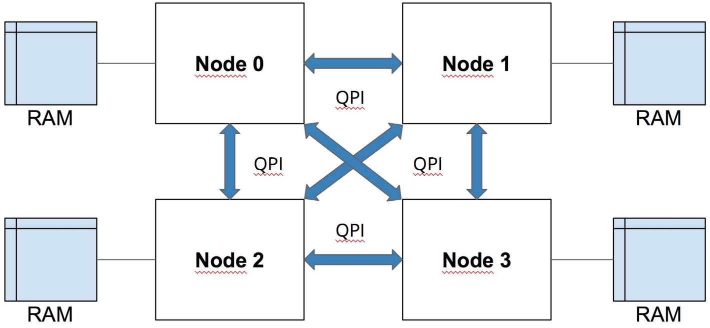


Intel

NUMA-aware Optimization of Data Locality

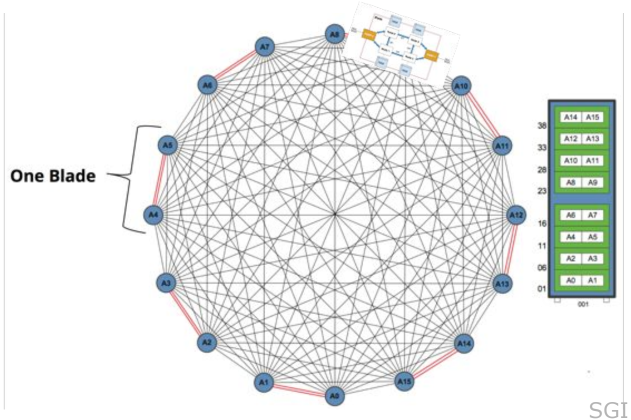
Motivation

NUMA has become the standard server architecture for enterprise systems. To deliver high performance, modern databases need to be aware and leverage these hardware resources. One particular aspect of interest are memory accesses, as data can reside in local or remote memory, resulting in different costs to process data.



Experiments and Tasks

- Measure the impact of various data locations (NUMA nodes) with respect to different access patterns (sequential & random)
- Compare & measure NUMA-effects for join queries
- Which are factors influencing a query optimizer’s decision in a NUMA scenario?

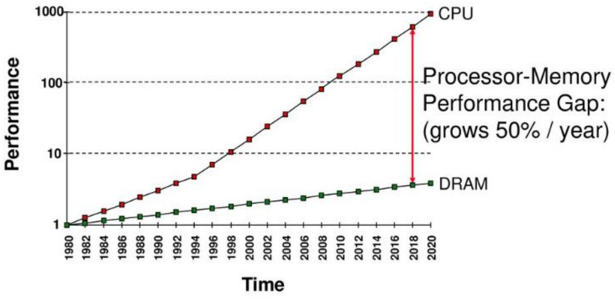


SGI

Column-oriented Database Compression

Motivation

With recent advances in CPU architectures (SIMD, many cores), the database performance bottleneck moved further up the memory hierarchy. To counteract this trend, database systems employ compression, thereby trading off "cheap" compute against "expensive" transfer costs. Particularly promising are these techniques for columnar data.



Rappoport and Yoaz: Cache Memory

Experiments and Tasks

- Implement compression schemes and evaluate their compression ratios as well as (de)compression overhead against data of various shapes
- Measure query performance on differently compressed data and explore the trade-offs between the applied algorithms.
- Discuss which compression technique is most adequate for which tier of the memory hierarchy.

Data Reorganisation Current/Historical Partitioning

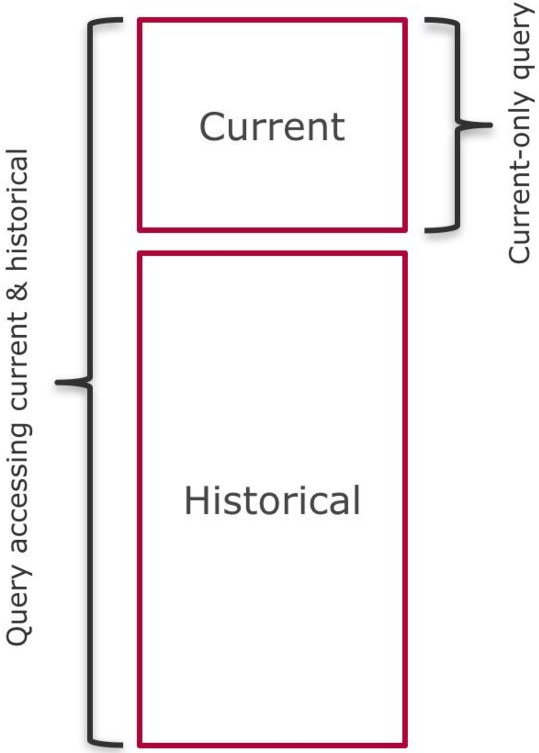
Motivation

Using the knowledge of domain experts, data can be separated in current and historical data

As the majority of queries only needs to access the most recent current data, we can prune accesses to the historical partition and skip all historical data

Experiments and Tasks

- Create two synthetical data sets: one unpartitioned and one split in current/historical partitions by 1:5
- Implement three access patterns (single scan, multi-column scan, and index scan) to simulate queries
- Evaluate throughput performance for queries accessing (i) the current-only and (ii) both partitions



Data Reorganisation
Partial Replication: Cost-Efficient Read-Only Scale-Out

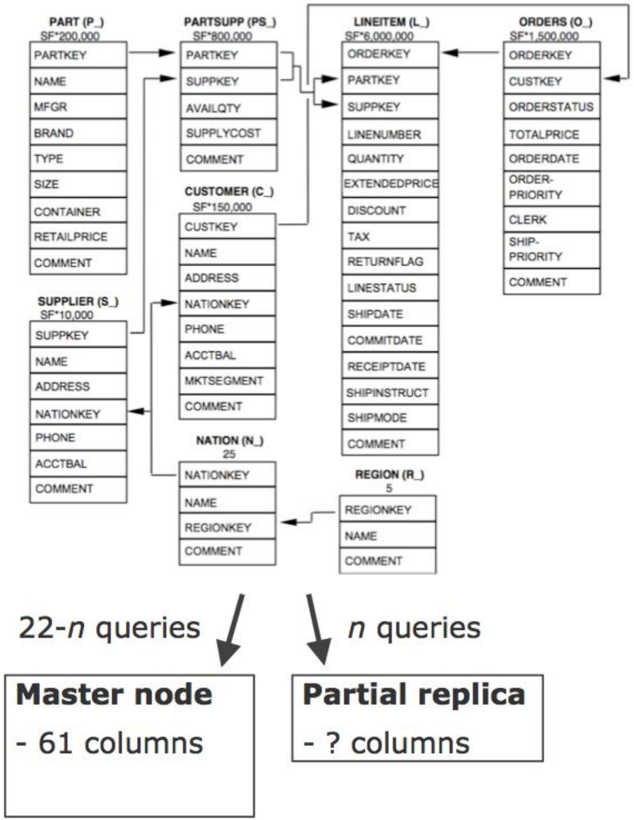
Motivation

Read-only queries can be processed on database replicas without violating transactional consistency. The majority of queries access only a small fraction of the overall data set.

We want to use partial replicas, which store table subsets, to answer the most frequent and costly queries.

Experiments and Tasks

- Set up a replicated TPC-H benchmark environment
- Implement algorithms to calculate replication configurations
- Demonstrate the throughput scaling with partial replication



22-*n* queries

n queries

Master node

- 61 columns

Partial replica

- ? columns

Group Assignment

Hardware Optimizations Markus Dreseler		**NUMA** Christopher Schmidt		**Compression** Jan Koßmann		**Data Reorganisation** Martin Boissier	
Team A	*Team B*	*Team A*	*Team B*	*Team A*	*Team B*	*Team A*	*Team B*
Arne M.	Felix W.	Julian W.	Alexander K.	Alexander P.	Volker S.	Leana N.	Felix M.
Jonathan J.	Daniel T.	Nils T.	Mathias F.	Frederic S.	Julian N.	Ramin G.	Reem A.
Oliver A.	Jonathan S.	Philipp B.	Lukas E.	Hendrik F.	Pascal C.	Maximilian D.	Tobias M.
Carl G.	Marvin M.	Julius R.	Marcel W.	Justin T.	Albert M.	Theresia B.	Tim F.
						Felix S.	

Block Week

- General information
  - □ 9th of July to 12th of July
  - □ Lectures given by Prof. Plattner
  - □ Discussions about open questions in in-memory computing are a vital part of the lecture!
- Focus areas
  - □ Basic principles of in-memory databases
  - □ Characteristics of modern enterprise systems
  - □ Advanced data structures for in-memory databases
  - □ Trends in enterprise computing

Contacts

- Guenter Hesse
  - Room: V-2.03
  - Email: guenter.hesse@hpi.de
  - Phone: 1381
- Ralf Teusner
  - Room: V-2.01
  - Email: ralf.teusner@hpi.de
  - Phone: 1301

Questions