
Continuous resource monitoring for self-predicting DBMS

Eno Thereska

Carnegie Mellon University

Dushyanth Narayanan **Anastassia Ailamaki, Greg Ganger**

Microsoft Research, UK

Carnegie Mellon University

Microsoft
Research

Carnegie Mellon
Parallel Data Laboratory

Management as major challenge

- Much effort into building high-performance, scalable database systems
- DB systems and workloads are complex

- Ease of management as evolutionary step
 - difficult to tune systems to get their full potential
 - large number of (expensive) administrators required

2

Eno Thereska September, 2005

Performance tuning and policy questions

- Important subset of management problems
- Some nightmare scenarios for a DB admin
 - clients complain about the performance they are getting from the database
 - admin needs to track system bottlenecks: CPU, memory, network, disks
 - at upgrade time, which resources should be upgraded?
 - can the admin quantify the effect of buying more RAM for each of the storage bricks?
- Ideally the system itself would answer these

3

Eno Thereska September, 2005

Common approaches

- DB admins as external expert systems
 - paid to act on their rules of thumb
 - difficult to quantify the effect of admin decisions
 - DB help: 400+ performance counters



4

Eno Thereska September, 2005

Common approaches (cont...)

- DB modules as internal expert systems
 - use if...then... rules of thumb
 - thresholds and rules may be incorrect or outdated
 - difficult to resolve conflict among different rules
 - no root-cause analysis of problem
- Overprovision
 - expensive for multi-million dollar systems
 - tuning problems do exist, even in an over-provisioned system
- Self-predictability is not built-in from the start

5

Eno Thereska September, 2005

What we propose: self-prediction

- DB should incorporate ability to answer What...If... questions about own performance

6

Eno Thereska September, 2005

Example What...If... question

- What would be the throughput/latency of new-order transactions if the buffer pool size of the DB is doubled?
- What would be the throughput/latency of new-order transactions if I buy faster disks?
- If the DB disk queues > 8, then double the buffer pool size

7

Eno Thereska September, 2005

What we propose: self-prediction

- DB should incorporate ability to answer What...If... questions about own performance
- What...If... modules built-in from the start
- What...If... interfaces convert complex tuning decisions into search-based approaches
- Main contribution:
 - prototype implementation and evaluation of a Resource Advisor for a commercial DBMS

8

Eno Thereska September, 2005

Outline

- Background and motivation
- Design and implementation of a Resource Advisor
- Evaluation
- Related work
- Future work
- Summary

9

Eno Thereska September, 2005

Resource Advisor goals

- Built inside SQL Server 2005 branch
- Answers *What...If...* questions about resource upgrades (CPU, buffer pool, network, disks)
 - point predictions and trend forecasting
- Provides visualization of system performance

10

Eno Thereska September, 2005

Architectural view of Resource Advisor

- Detailed, yet general, system instrumentation
 - away from performance counters
 - towards end-to-end activity tracking
 - enables resource accounting and critical path info
 - built-in at system development time
- Specific *What...If...* modules
 - query system instrumentation framework to answer specific *What...If...* questions
 - some *What...If...* modules built-in from the start, others can be added as needed
 - utilize operational laws and simulation

11

Eno Thereska September, 2005

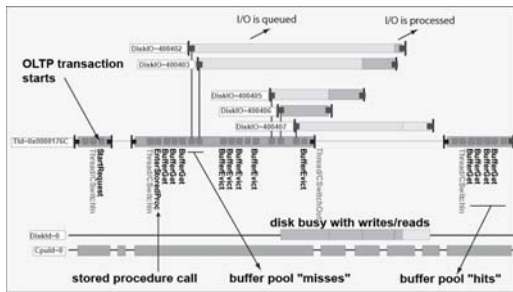
General system instrumentation

- Per-transaction, per-resource monitoring
 - stored procedure calls
 - buffer cache accesses + prefetching
 - SQL server task scheduler activity
 - NT threads scheduling
 - I/O activity
- Magpie used for trace collection [OSDI2004]
- Virtual performance counters & visualization

12

Eno Thereska September, 2005

An aside: visualization



13

Ero Thereska September, 2005

Models for answering *What...If...* questions

- Two fundamental system assumptions
 - past is good indicator of future
 - DB performance is some function of system resources and workload
 - goal of prediction is to approximate the above function
- What we are and are not targeting
 - no help for initial purchase decision
 - no predictions during drastic changes
 - can predict when not to predict

14

Ero Thereska September, 2005

What-if modules

- Goal: predict client performance as buffer pool size changes
- Performance metrics of interest
 - closed-loop throughput with a given think time
 - system saturation throughput
 - open-loop request response time

15

Ero Thereska September, 2005

Throughput prediction

- Analytical model for throughput predictions
 - bottleneck analysis on CPU/RAM/disks/(client)
 - resource with highest demand determines the overall throughput of the system's workload
 - query demands and predict bottleneck
- $T = \min \{T_{max/cpu}, T_{max/io}, T_{max/workload}\}$
- $T_{max/cpu} = t_{cpu}^{-1}$
- $T_{max/io} = (n_{io} t_{io})^{-1}$
- $T_{max/workload} = N_{users} / t_{think}$ (for closed-loop)

n_{io}, t_{io} : need to be predicted

16

Ero Thereska September, 2005

Response time prediction

- Response time depends on *critical path*
- $t_x = t_{\text{CPU}} + t_{\text{storage}}$
- t_{CPU} is measured
- t_{storage} depends on number of **blocking I/Os** per transaction and **service time per I/O**

blocking I/Os, service time/I/O : need to be predicted

17

Ero Thereska September, 2005

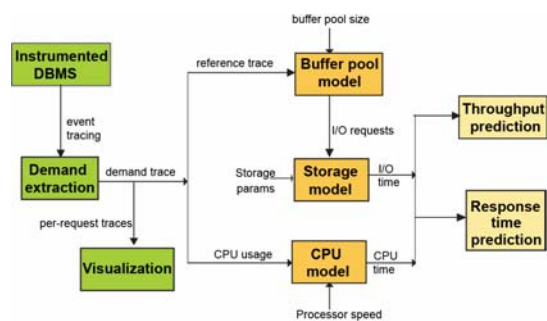
System integration

- System resources are augmented with analytical or simulation modules
 - DB cache manager: simulation module
 - predicts blocking and non blocking I/Os as function of cache size
 - DB storage subsystem: analytical module
 - predicts I/O service time as function of storage parameters (seek, rotational latency, queue size, etc)
- A self-predicting system must integrate these models in and treat them as first-class citizens

18

Ero Thereska September, 2005

System integration



19

Ero Thereska September, 2005

Outline

- Background and motivation
- Design and implementation of a Resource Advisor
- Evaluation
- Related work
- Future work
- Summary

20

Ero Thereska September, 2005

Experimental setup

- 2 variants of TPC-C (open, closed)
 - 100 warehouses (~10GB)
 - 200 clients accessing DB concurrently
- Xeon 2.66GHz machines
- 64-128-256-512-1024 MB of RAM
- DB server runs SQL Server 2005

21

Eno Thereska September, 2005

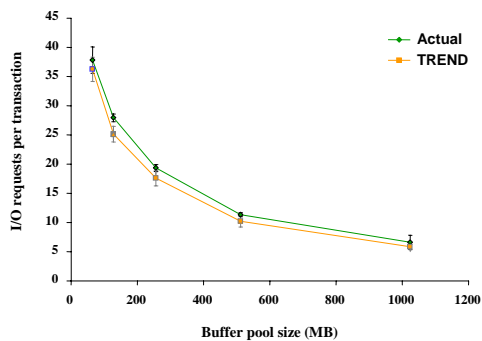
Buffer pool model evaluation

- How accurate is the buffer pool model in predicting I/O sequence?

22

Eno Thereska September, 2005

Buffer pool model evaluation



23

Eno Thereska September, 2005

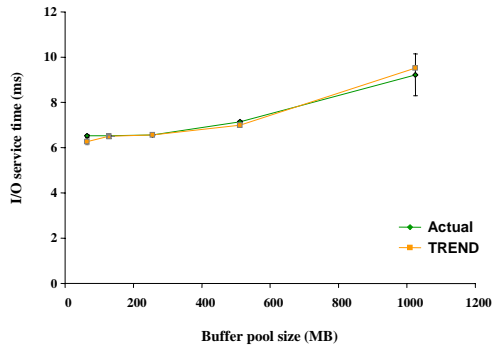
Storage model evaluation

- How accurate is the storage model in predicting service times?

24

Eno Thereska September, 2005

Storage model evaluation



25

Eno Thereska September, 2005

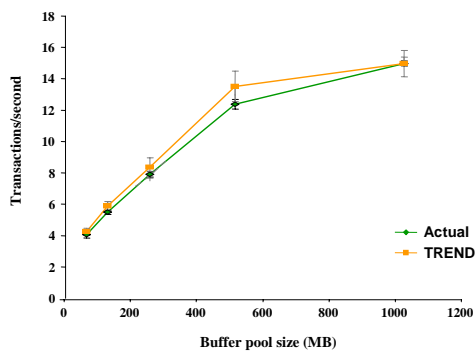
Closed-loop throughput prediction

- For a closed-loop workload, can the Resource Advisor predict throughput at different buffer pool sizes?

26

Eno Thereska September, 2005

Closed-loop throughput prediction



27

Eno Thereska September, 2005

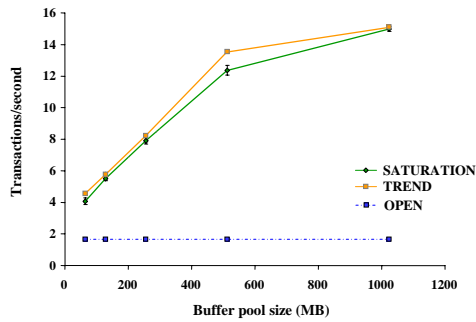
Predicting saturation from open-loop

- Given a non-saturation workload, can it predict saturation throughput at different buffer pool sizes?

28

Eno Thereska September, 2005

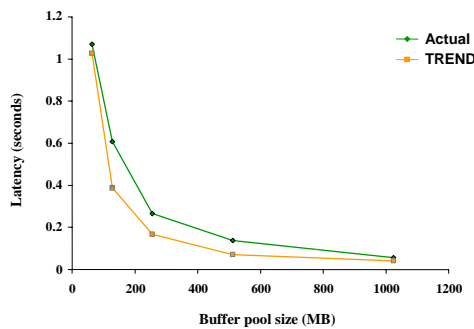
Predicting saturation from open-loop



Predicting open-loop response time

- Given a non-saturation workload, can the Resource Advisor predict response time at different buffer pool sizes?
- Response time prediction is per-transaction type

Predicting open-loop response time



Resource Advisor overheads

- 6.2% CPU overhead
 - 1.2% for tracing and logging
 - 0.44MB/s of trace data
- Instrumentation burden
 - 189 lines of code in 6 source files
- What-if modules burden
 - 1150 lines of code

Outline

- Background and motivation
- Design and implementation of a Resource Advisor
- Evaluation
- **Related work**
- Future work
- Summary

33

Eno Thereska September, 2005

Related work

- Instrumentation frameworks & upgrades
 - Magpie [Barham04], Pinpoint [Chen02]
 - instrumentation of middleware [Chen04]
 - instrumentation of black-box systems [Aguilera03]
- Analytical laws
 - operational laws [Denning79]
 - applied to capacity planning [Menasce98]
- Other DB self-tuning approaches
 - DB2 Advisor [Lohman00]
 - SQL server advisor [Agrawal04]
 - AutoPart [Papadomanolakis04]
- DB work on understanding where the time is spent [Ailamaki99, Barroso98]

34

Eno Thereska September, 2005

Future work

- In SQL Server
 - predict locking effects as more CPUs are added to the DB
- Apply similar techniques to a distributed storage system
 - Ursa Minor developed at CMU [FAST05]
 - opportunity to solve real admin's *What...If...* questions

35

Eno Thereska September, 2005

Summary

- Common tuning approaches inadequate
 - system offers little help to external admins
- Self-predictability should be built-in
 - need general instrumentation
 - need specific *What...If...* modules
- Main contribution: prototype Resource Advisor inside a commercial DBMS

36

Eno Thereska September, 2005