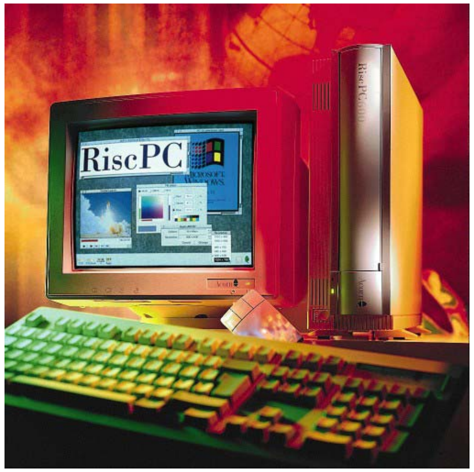
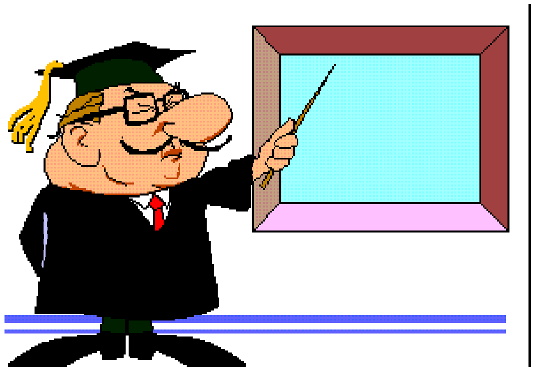
TOWARDS A SELF - TUNING RISC-STYLE DATABASE SYSTEM



By
Lakshmi Dhevi Baskar.



INTRODUCTION

INTRODUCTION

- Database has proven its importance as backbone of information technology.
- ‘***Success is a lousy teacher***’- by Bill Gates – applies for this backbone technology.
- Gain/Pain ratio is poor for the current database architecture .
- Focus is on new departure of database system – self tuning and RISC style.

CRISIS INDICATORS

CRISIS INDICATORS

Traps: Opportunities and Complexity.

Universality Trap :

• More featurism added into single product for marketing issue .

As a result

- Increases the code size and complexity .
- Installing and maintaining the database system is crucial.
- Performance is unpredictable.

CRISIS INDICATORS

Cost Trap :

- DBMS is packaged as monolithic systems (with too many features).
- Disregards the guaranteed performance ,cost of maintaining the system.
- More problematic for customers than for vendors.

CRISIS INDICATORS

Transparency Trap :

- Union of all conceivable features in SQL is complex for application developer.
  - ➢ No high confidence about the results from high level SQL query .
- SQL is painful with hidden execution costs(runtime) and careless programming.

CRISIS INDICATORS

Resource Sharing Trap:

- The hardware is shared for different purposes in single box .
- Example :

  Video streaming.

  OLTP applications.
- paves way for tuning problems with same resource sharing (disks).

CRISIS INDICATORS

Programmer Trap:

- To paraphrase Dick the Butcher
  - "***First thing we do, let's sack all the DBAs...***".
- Skilled DBA or tuning gurus are scarce and expensive .

  dominates the cost of ownership for database system.
- Auto-tuning the critical parameters is wishful thinking.
- To put in short '***too much of anything is good for nothing...***'.

CRISIS INDICATORS

GPR:

- The gain of using a full fledged database system is low with the pain of installing, managing and predicting performance.
- So we go for automation of tuning decisions leading to self-tuning database.

WHERE DO WE GO FROM HERE ?

WHERE DO WE GO FROM HERE ?

- For trouble free and autonomic systems

  we need a radical departure from current architecture .
- Following the role models in other engineering fields (aircrafts)

  we try the idea ***'think globally ,fix locally'***.
- So a major incentive to move towards to RISC architecture is to

  enable Auto-tuning of database components.

WHY IS RISC STYLE ATTRACTIVE ?

WHY IS RISC STYLE ATTRACTIVE?

- The components have
  - Narrow functionality – new hope for predicting performance.
  - Highly componentized - paves way for building varied applications.
  - Stable and narrow interfaces - reducing complexity between components.

RISC PHILOSOPHY FOR DATABASE SYSTEMS

RISC PHILOSOPHY FOR DATABASE SYSTEMS

- Simpler the interfaces and underlying internals are fewer the tuning knobs and predicting becomes easier.
- The layering in querying a database system.
  - **Layer 1** Single Selection Processor - single table and simple updates.
  - **Layer 2** SPJ query engine - for OLTP and business applications. Adding support for aggregation – helps for OLAP decisions.
  - **Layer 3** SQL processor that uses the layer 2.

RISC PHILOSOPHY FOR DATABASE SYSTEMS

Advantage:

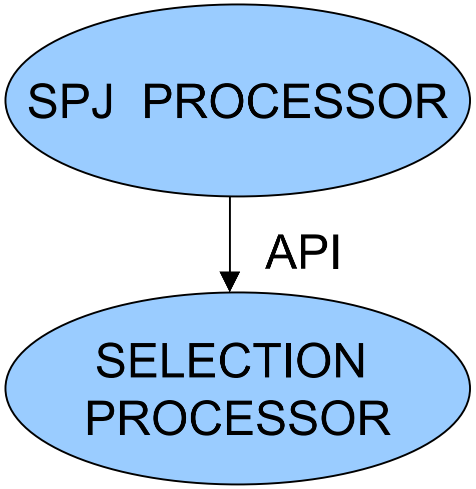
- Controls the search space for each layered component much more tightly.
- Independent usuage of layers and manageable components.

RISC philosophy for IT systems in a large:

- More building blocks.
    - ➢ Each component constructed by RISC style.

POINTS TO BE NOTICED

- Limited Interactions among components



- API – two interface classes.
  - ➢Functionality –specification of query request.
  - ➢Import/export of meta information.

DEPARTURE FOR AUTO-TUNING

NOTABLE DEPARTURE FOR AUTO -TUNING

- Support only limited data types.
  - Tables with elementary data type.
  - More advanced APIs.
- No more SQL.
  - Use operator trees to the database server module.
- Disjoint , manageable resources.
  - Dedicated hardware for simpler tuning.

PREREQUISTES OF SUCCESS

PREREQUISITES OF SUCCESS

Universal Glue:

- Multiple components must be composed into value added services without re-introducing a poor GPR.
- Simple interfaces with standardized cross-talk protocol is required for manageability and composability.
- We require some middleware to communicate to each underlying data server.
- Such universal glue is available today.

PREREQUISITES OF SUCCESS

Apply Occam's Razor:

- Features that are to be supported and internal mechanisms needed.
  - ➢ To minimize the complexity of both interfaces and internals.
- Avoid certain mechanisms that may improve the performance slightly but add the tuning complexity.
- Example
  - ➢ Use of Null Values at the application than in underlying data manager.

PREREQUISITES OF SUCCESS

Need for a Self-Tuning :

- Earlier tuning was done based on mathematical model.
- But these models work on limited set of interrelated knobs.
- For tuning the full spectrum of tuning issues, accurate model is not available.
- Using RISC style we have hope to handle individual component.
- A Simple Thought ??

But how can we tune the interplay of several RISC data managers.

➢ Hierarchical self-tuning framework to solve .

EVALUATION OF SUCCESS

- Demonstrate the usefulness of the components in variety of data management applications
- Example:

  For OLTP and OLAP we use the SPJ and SPJ+Aggregation layer

RESEARCH OPPORTUNITIES

Challenges in large scope :

- Make an open , worldwide testbed for RISC style management.
- Work out lean APIs for each component.
- Encourage world wide competition for the best instantiation of each block.
- All the components in the testbed must correctly cooperate with each other.
- Identify ‘universal glue’ for the above kind.

CONCLUSION

CONCLUSION

- Universal database system is one of the milestones in IT.
- But it has low GPR.
- We introduced RISC comparing other engineering fields (space-craft).
- The key aspect of this paper is to improve the gain/pain ratio.
  - ➢ Eliminating the pain of manual tuning.
  - ➢ Improving the gain by tolerating the interface crossing across the boundaries.
- Understanding and usage of narrow API - difficult.
- The acceptance of this new architecture by IT industry-unpredictable.

REFERENCES

1. G.Weikum , C.Hasse , A.Moenkeberg , P.Zabback :

The COMFORT Automatic Tuning Project , Information systems.VOL.19,No.5,1994

2. A.Geppert, K.R.Dittrich :

Towards New Construction Paradigm For Persistent Systems,Networking and Information systems Journal, Vol.1,No.1,1998

3. http://www.intelligententerprise.com/021115/518decision1_1.shtml/imp6

ANY QUESTIONS ?