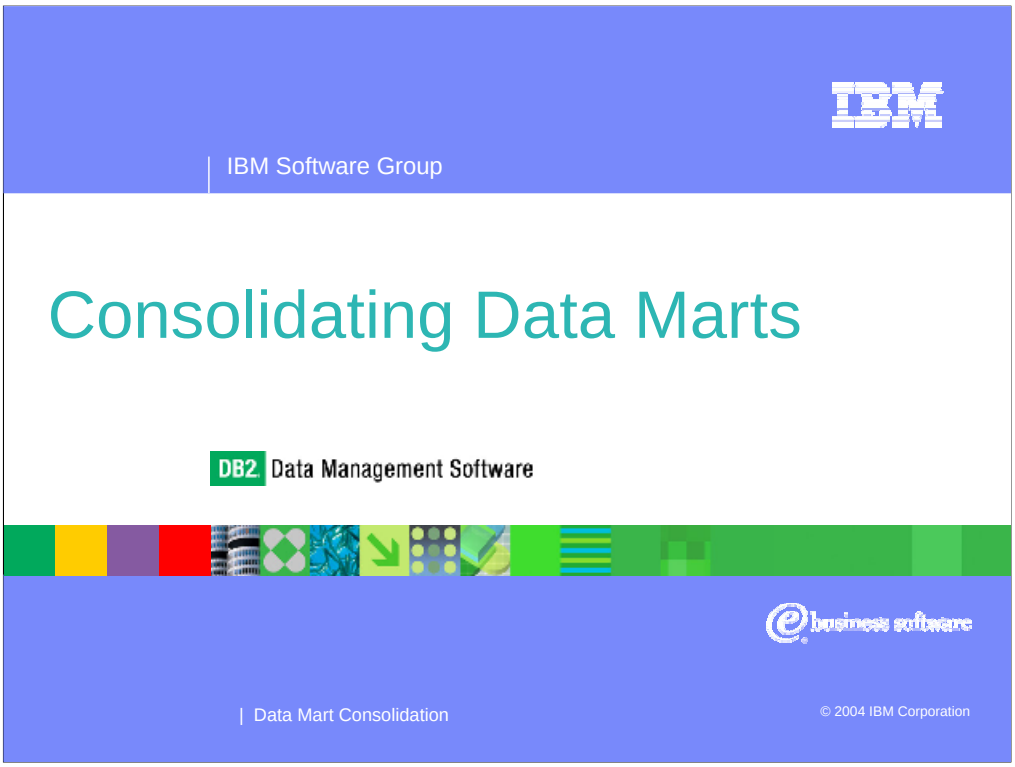


Thank you for attending this session, we believe that what we'll cover can potentially save or make you a lot of money.

What we'll talk about is how you *optimise* the return you get from your BI system.

A key point here is that most BI systems provide a return on investment, they pay for themselves many times over usually – and that was good enough in the early days – but now companies want to know they're getting the maximum return for the optimum investment.

When I first got involved in BI systems over 14 years ago it was easy to show ROI, for example the Marketing Director of a large national Telco once told me that the way they found out their best customers was to go to a big city, go to the top of the tallest building – the other tall buildings they could see were their best customers. In that environment finding out who your best customer was, by profit rather than size or by activity was something that couldn't have been done before, and simply knowing that for the first time was worth a lot of money.

These days that type of analysis is business as usual – and we want to do it at reasonable cost.

Collections of Marts have typically given benefit, but we'll see that that's often not at optimum cost. To be truly competitive customers want to be able to get the full benefits of BI analysis but not to pay riduculous costs to obtain it. That is one of the themes of todays cost.

Disclaimers and Trademarks

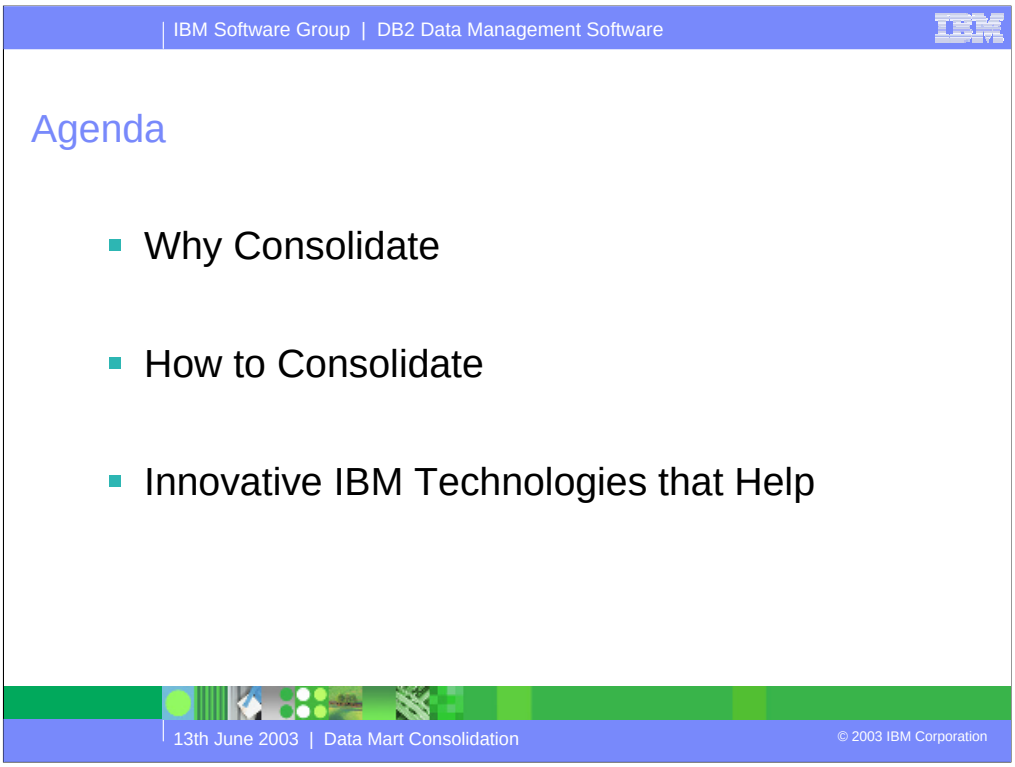
REFERENCES IN THIS PUBLICATION TO IBM PRODUCTS, PROGRAMS, OR SERVICES DO NOT IMPLY THAT IBM INTENDS TO MAKE THESE AVAILABLE IN ALL COUNTRIES IN WHICH IBM OPERATES.

IBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS COVERING SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.

TRADEMARKS

THE FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: AIX, AS/400, DB2, OPERATING SYSTEM/2, OS/400, ES/9000, OS/390, OS/2, RISC, RISC SYSTEM/6000, SQL, SQL/DS, VM/ESA, IBM, APPROACH, NOTES

THE FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES:MICROSOFT, WINDOWS, WINDOWS NT, ODBC, WINDOWS 95, WINDOWS 98, WINDOWS 2000, WINDOWS ME



What we'll go through is straightforward

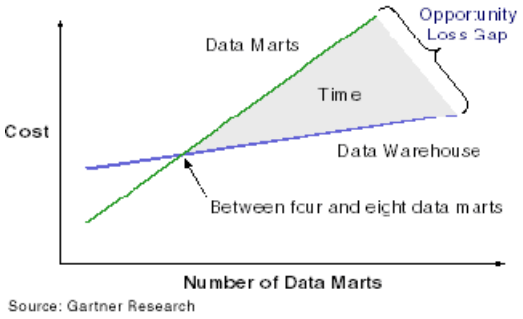
We'll talk about why to consolidate – introducing the background of the analysts who first raised this, and the advantages of doing this,whether this is technical, economic or strategic.

Then we'll talk about how to go about it and the issues that need to be addressed.

We'll then finish off by talking about how new, unique technologies from IBM will help these types of efforts.

Gartner – High Cost of Data Marts

- As # data marts increase
  - Delivery costs increase by 70% per additional mart
  - BUT ROI decreases by 50% per additional mart
- By 4-8 data marts costs outweigh ROI



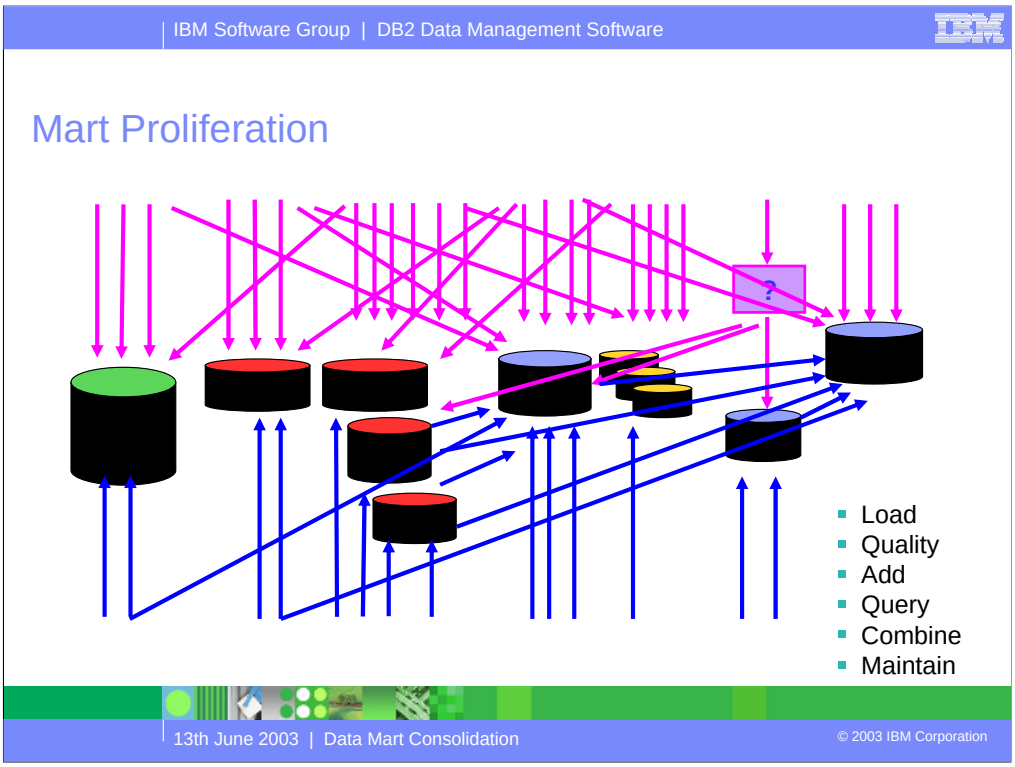
Source: Gartner Research

Interest was first stimulated by the observations of independent consultants, in particular Gartner which authored several influential papers.

Data Marts are typically small and simple, relatively quick to implement, but limited in scope.

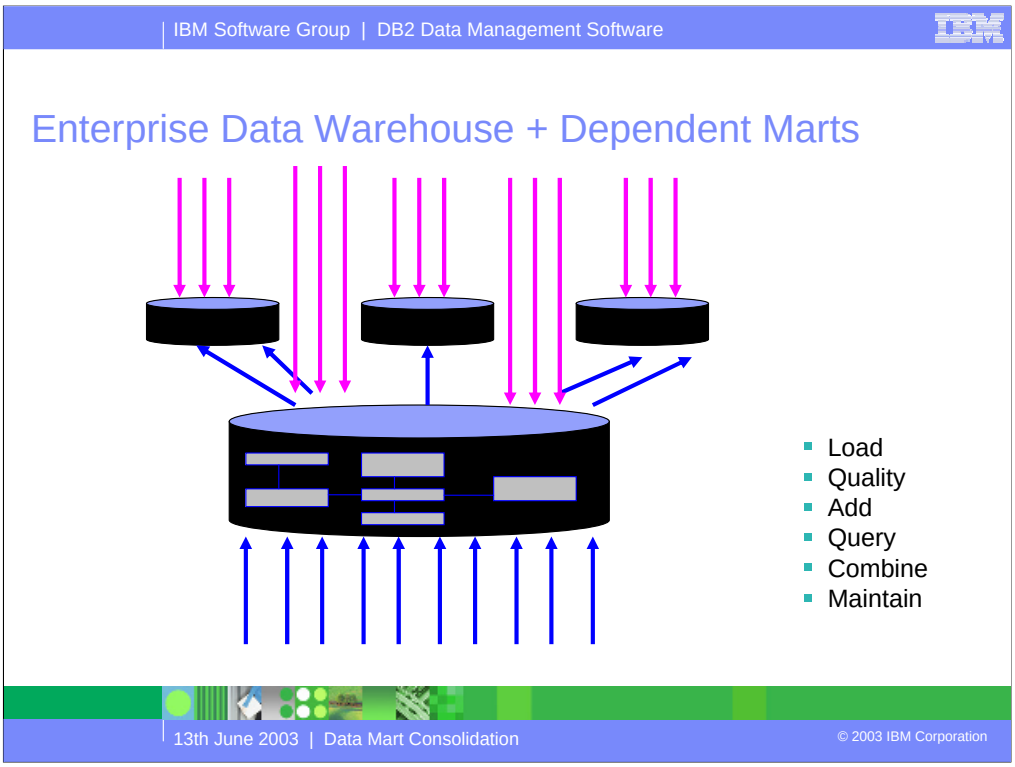
Data Warehouses are much more flexible, need more preparation and have a wider scope, but typically require more up front costs.

What they observed was that whilst Data Marts were individually quick and easy to implement by the time 4-8 marts had been implemented you had an information delivery infrastructure that cost just as much as implementing a full Data Warehouse but cost more and was less effective.



We can see how Marts Proliferate by taking a simple illustration – on the way through we'll discuss the pro's and con's of this.

- The org builds a Mart, e.g. Marketing. Purple arrows going downwards show user access, blue arrows going up show the data feeds.
- Anoth part of org builds a Mart, e.g. Finance. We notice that these two marts are independent and serve different users – no problems at all.
- Series of Marts implemented copying the technology of the 2nd, different subject areas. Starting to see cross access between Marts and Users. The Users have to know which Marts to access, and they may be on different technologies.
- The organisation then acquires a package. It was advertised as accessing any existing data. In practice that didn't work – so the org solved the problems by giving it its own Mart. Most of the data needed was already in other marts so we implemeted cross feeds. Users of the new application need access to the orginal marts, and vice versa.
- Meanwhile whilst our developers are tied up with the new app there are a group of users, power users, who've grown frustrated with waiting for new requirements to be built and have gone off and implemented their own collection of Marts using a different technology. This is a simple to use desktop technology.
- Note that from a benefits point of view we havn't got a problem each new implementation solves another problem and gives more benefits.
- Up until now we've had users who want to access different marts, and have to choose between the different ones available.
    - By this stage we have most of the organisations data available – now people want to write queries that combine the data already held.



Here we see the alternative. (The Counsel of Perfection !)

The bulk of the data is contained in an Enterprise Data Warehouse, and that is where most of the queries are serviced from.

Note that here we've not elimated Mart's completely –but there are fewer of them and they are integrated with, and dependent upon the EDW.

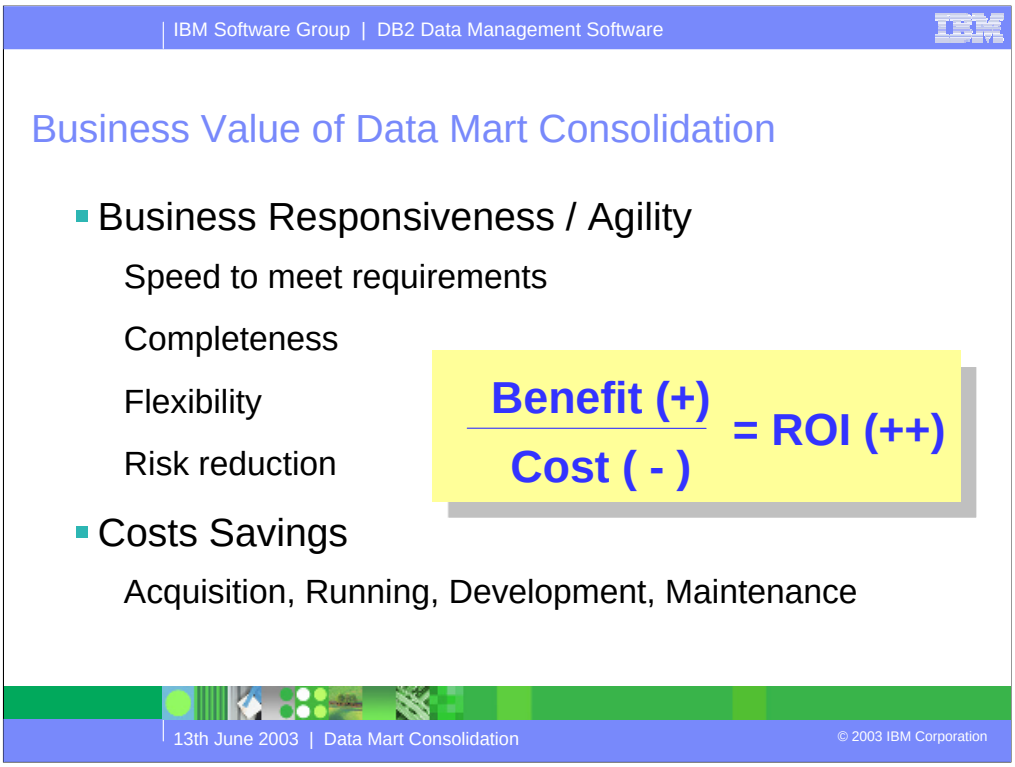
This is deliberate. There are examples of where the controlled use of marts provides the most cost effective and performant solution – so there is no reason not to use them. It is their uncontrolled use and proliferation that is the problem.

Here if we ask our basic questions we get some ready answers.

1. Where do I load my data to ?
2. Where do I satisfy my query from ?
3. How are my query results used ?
4. Where do I add new data ?
5. How do I do analysis that needs data from different sources ?

Now, having said that although we can see the attractions of this kind of set up we would also note that ***almost no organisation does this*** ! The Data Warehouse systems that we've looked at, from all vendors, almost always look like the previous slide rather than this one.

Therefore the issue is not 'how do we grow our system to revolve around the EDW' but rather 'given we have a divergent collection of data marts and maybe a DW



It is important to stress that these initiatives are not just about cost cutting, as we'll have seen there is a very large component that is about delivering more business benefit and delivering it more quickly.

Requirements met earlier / more completely

Business Information more accessible

Information is of assured quality

More resource available for 'customer facing analysis' rather than 'back end infrastructure support'

Cost Savings

DW vs. multiple DM's

Reduction in multiple vendor support

Reduction in implementation costs

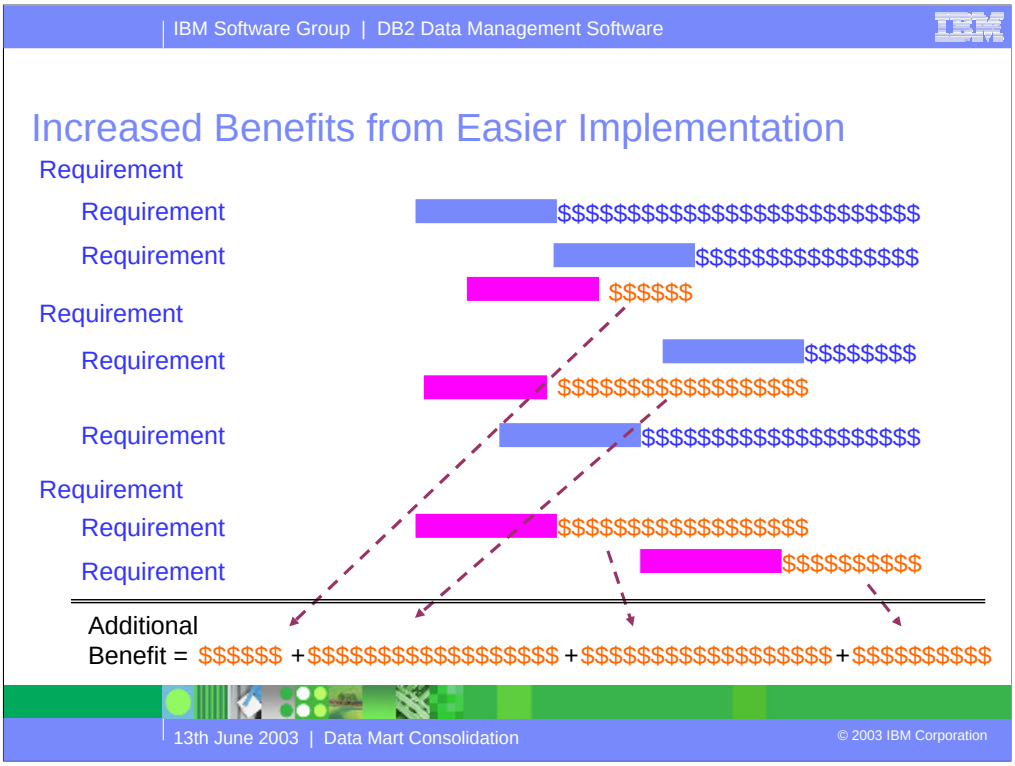
Reduce data feed complexity

Reduction in running costs – many of those marts may be underutilised a lot of the time.

Risk

Key Staff / Skills

to implement multiple source analysis need skills who can access each and all technologies used.



Why do we say that consolidation gives business benefit rather than just cost savings ?

We have seen why consolidating our data makes it easier to implement requirements.

In this simplified diagram we see a number of requirements, we'll have projects to implement them, and once implementation is done that project will provide a stream of benefits.

We'll be constrained on how many projects we can do at once by the resources we've got, availability of key skills, how readily accessible the data is etc.

Usually there will be some requirements that we can't put in plan at all.

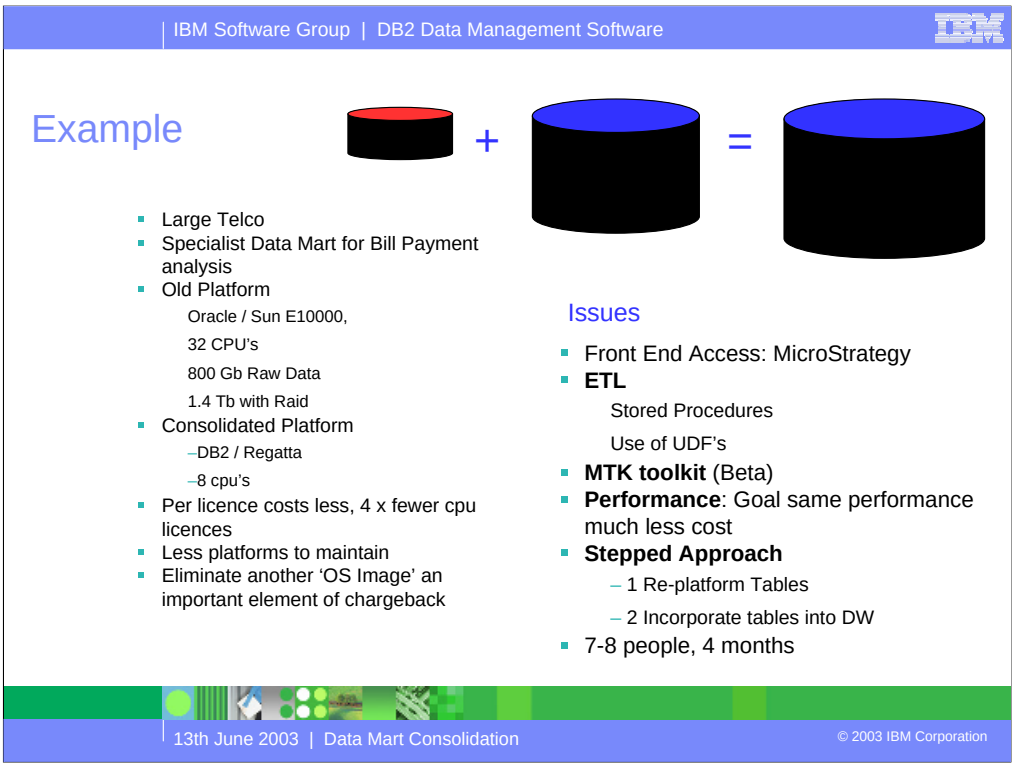
It is obvious that by making some of the chances we've suggested it becomes easier to implement new requirements.

Now this isn't a panacea, it's not going to affect all requirements – but we would expect some to be accelerated, and in addition there will probably be a few that are enabled that we wouldn't otherwise have gotten around to.

Bringing in requirements early means the benefit stream switches on early, and there are benefits streams we wouldn't otherwise have got.

This benefit would have been lost for all time had we not enabled it. If we can quantify this would probably pay for a consolidation effort.

Of course to do this we would need a definitive list of requirements – but that's a good thing to have anyhow, and most of us do.



Here's an example, this company is moving a large special purpose Data Mart into their general purpose DW.

The system currently runs on a a very large and expensive machine, a dedicated Sun E10K running Oracle, it is a 32 cpu machine running 800 Gb of raw data and 1.4 Tb of spinning data.

(this in itself looks like a very expensive solution, this is a very high cpu to DASD ratio)

This will be moved into the standard DW, and be run there on 8 IBM Regatta cpu's.

So we swap 32 Oracle licences for 8 DB2 ones – which are in any case less costly per CPU !.

In addition we make use of a much less costly hardware platform.

The **chargeback mechanism** also charges users for the number of 'OS Images' used. This will eliminate the OS image.

The organisation will have one less platform to support, and reduce the need for multiplatform support.

In doing this migration there have been a number of challenges.

First off the data definitions were easy to change. There is pretty much a straight one for one correspondence between Oracle and DB2.

We used the Beta version of our Migration Toolkit (MTK) to help automate, or at least semi-automate the conversion.

Marts Exist in Many Guises

- 'Summary Tables'
- Cubes
- Subject Specific Marts
- "User Work Areas"
- Shared Report Cache
- **A form of optimization**
  - Stand alone
  - Data Structures in Data Warehouse
  - Same or different technologies
  - Performance / SLA
- Security
- Data Marts are Very Common
  - No matter what vendor

"What's in a name? That which we call a rose by any other word would smell as sweet."

William Shakespeare, Romeo and Juliet

Different Types
Summary Tables
Cubes
Subject Specific Marts
User Work Areas
Stand alone machines
Either same or different technologies
'Summary Tables' or Cubes –are these actually marts (more on this later)

A) DM constructs in DW are common
B) That in some cases what's needed is the equivalent of an MQT or AST.

A form of optimization

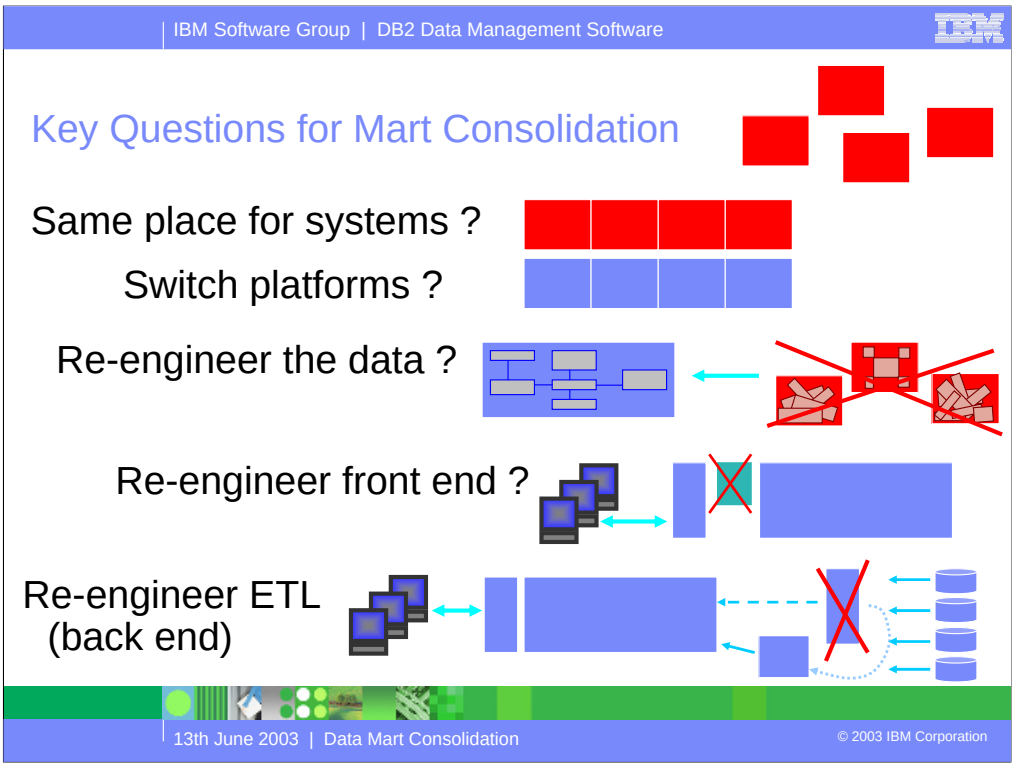
Laws of Physics don't change –
nor do benefits of optimization (as seen from the narrow perspective of beneficiary)

Data Marts are Very Common

We have undertaken numerous migration studies
Competitor marketing would have you believe that they promote single unified 3NF Data Warehouses
We've not found one yet !
In many cases the bulk of the system are local data extracts and data marts.
For example
Major Bank
22,500 tables in system – how can this possibly relate to a banking Data Model
Our BDW is 2,000 entities and is widely regarded as being comprehensive
Answer is- each month many users copy data into their own private ('data mart ?') areas
But this organisation will tell you that it has an enterprise data warehouse

•Most people find it convenient to have data constructs to aid peformance / security / SLA

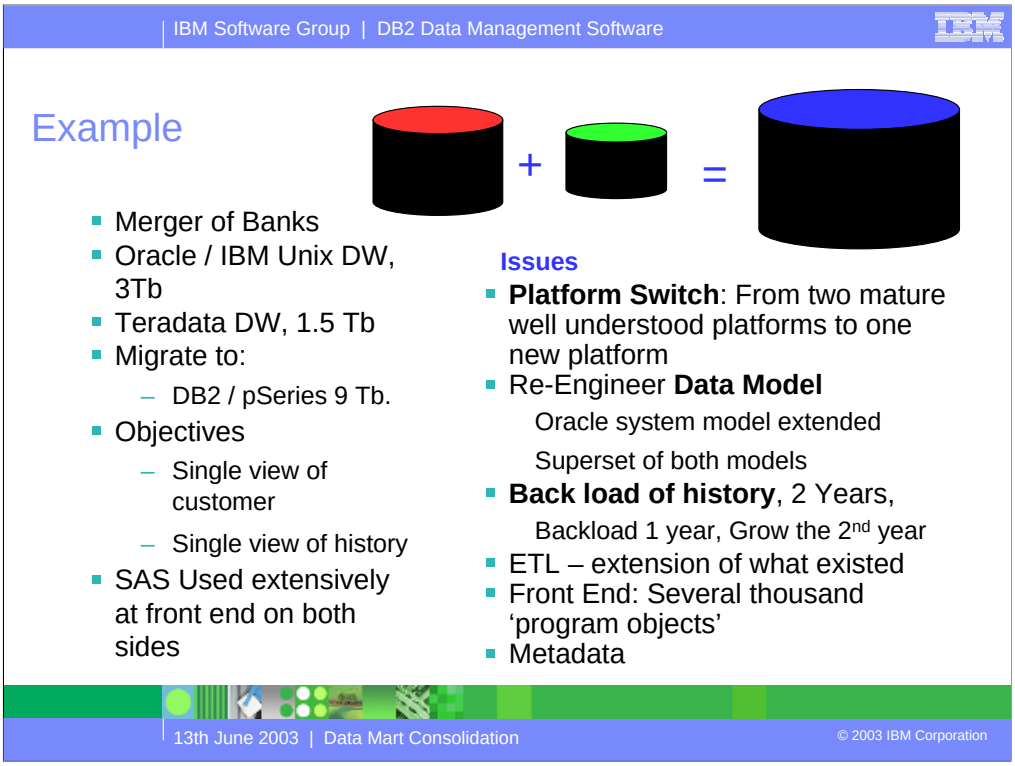
In many cases the bulk of the system are local data extracts and data marts.



There are a number of different variations in the type of consolidation that we can do.

Here we step through some of the key questions that determine what type of consolidation is appropriate.

{go through examples and discuss}



In this example two systems one Oracle one Teradata were amalgamated and ported onto DB2.

The purpose of this is not to talk about the relative merits of these systems, though we're happy to ! But rather to discuss what was done and how.

The Oracle system belonged to the main bank in the merger, it was larger and had a more comprehensively developed set of applications. The Teradata system was more of a marketing Data Mart.

Following an evaluation process the combined organisation decided to move to both platforms to standardise on DB2.

The key driver here was business agility – being able to use a single system to look at combined account base with a single version of the truth and to be able to access a single set of historical data – without this none of the combined banks business processes would work properly.

At the same time the combined platform would enable savings on the IT side through consolidation of both the platforms and support functions.

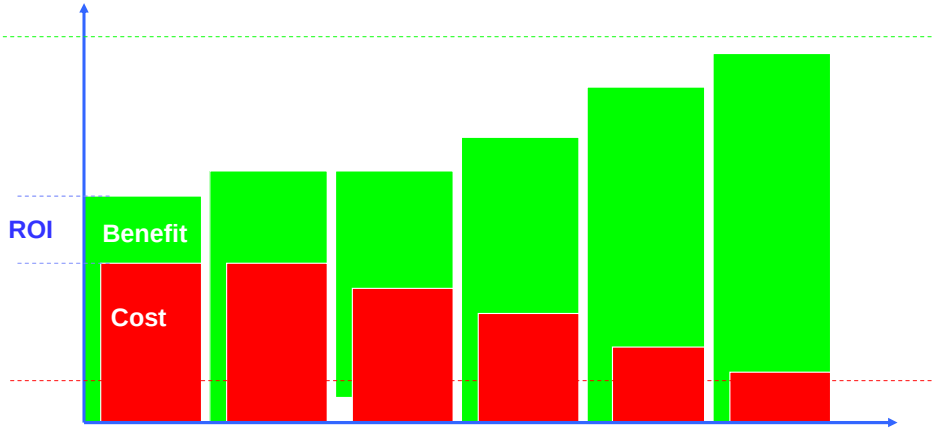
The data had to be restructured and the data content de-duplicated.

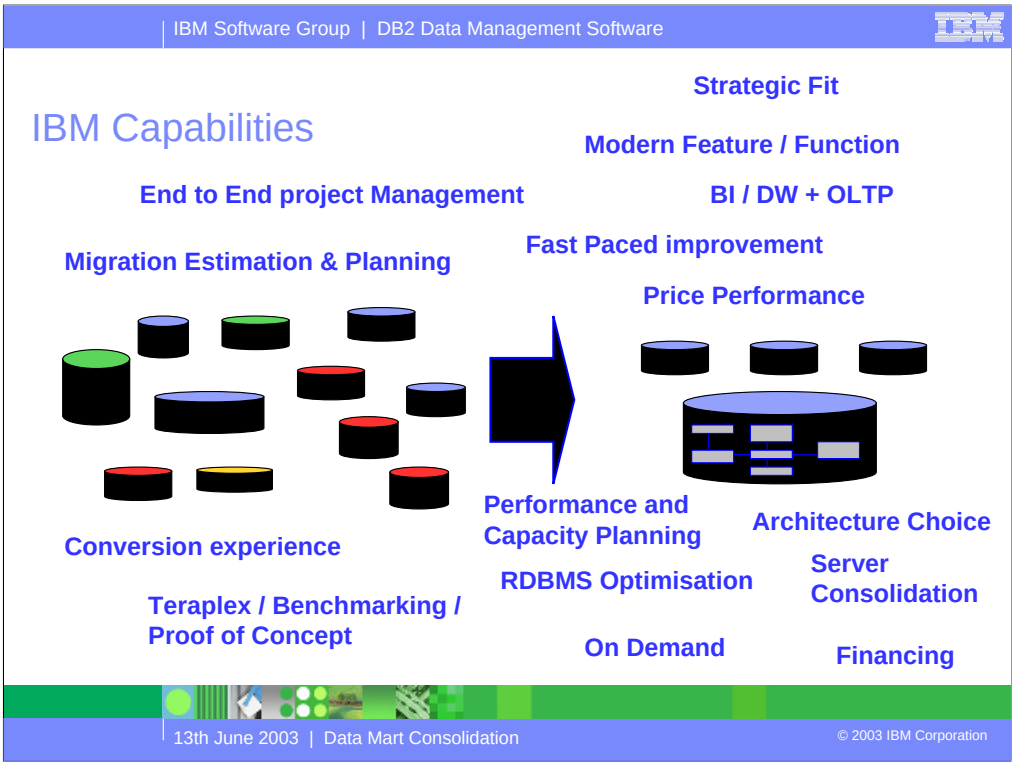
The new system was to have two years of historic transaction data. This was to be the merged transaction histories of both systems. In practice what was done was to go back and 'backload' (using special routines) the data from one year of the existing systems, and then to 'grow' the 2nd year during development.

The overall migration took about a year.

During this time several thousand (4,200) program objects (queries or rollups) were ported. In addition several hundred (320 + 150) ETL program objects were

We do not need, nor expect, to fix this instantly





Practices and methodologies – not technologies.

Consolidation can be done and is worthwhile, but it is also not trivial, and we wouldn't pretend it is.

IBM has the right mix of capabilities that are needed to successfully tackle a consolidation.

Some of these capabilities are reasons why consolidation is desirable, some provide the confidence to be able to execute.

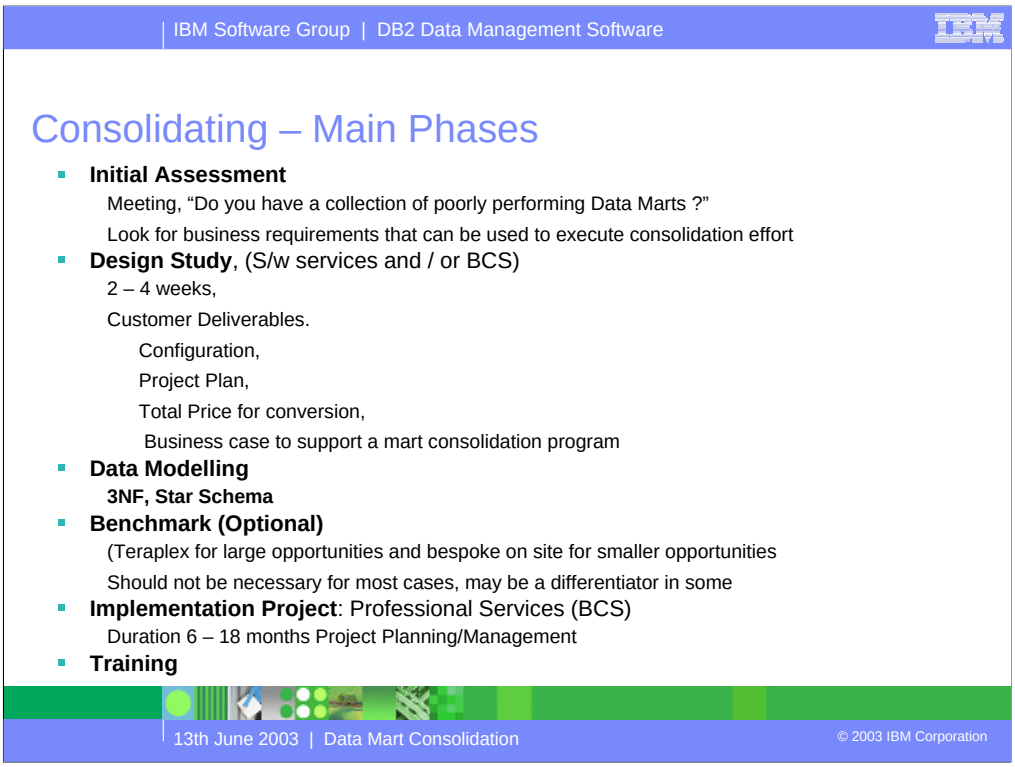
There are two sides to getting business benefits.

First, is there a business case ? Are there good reasons for doing it and will the exercise potentially give benefits ?

Second, if it is desirable can those potential benefits be realised by executing a practical plan ?

Desirable Reasons

- Fast pace of innovation
- Modern feature / function
- More likely to fit with future strategic direction (server consolidation, Open Interfaces, etc)
- Only RDBMS designed and proven to support OLTP and DSS mixed workloads
- Best TCO
- Moving to price performant platform



The means by which we consolidate are pretty straightforward.

It is a process of multiple phases, each phase building the confidence for the next phase to go ahead, and to provide checkpoints where we can take a go / no go decision on whether to proceed.

We'd work with you to do an initial assessment, this is normally just a few days, and what we're doing is establishing a basic business case and feasibility. There is no point in expending a lot of time and effort in planning a consolidation if there are some obvious show-stopping problems that would prevent it, this short phase aims to surface these.

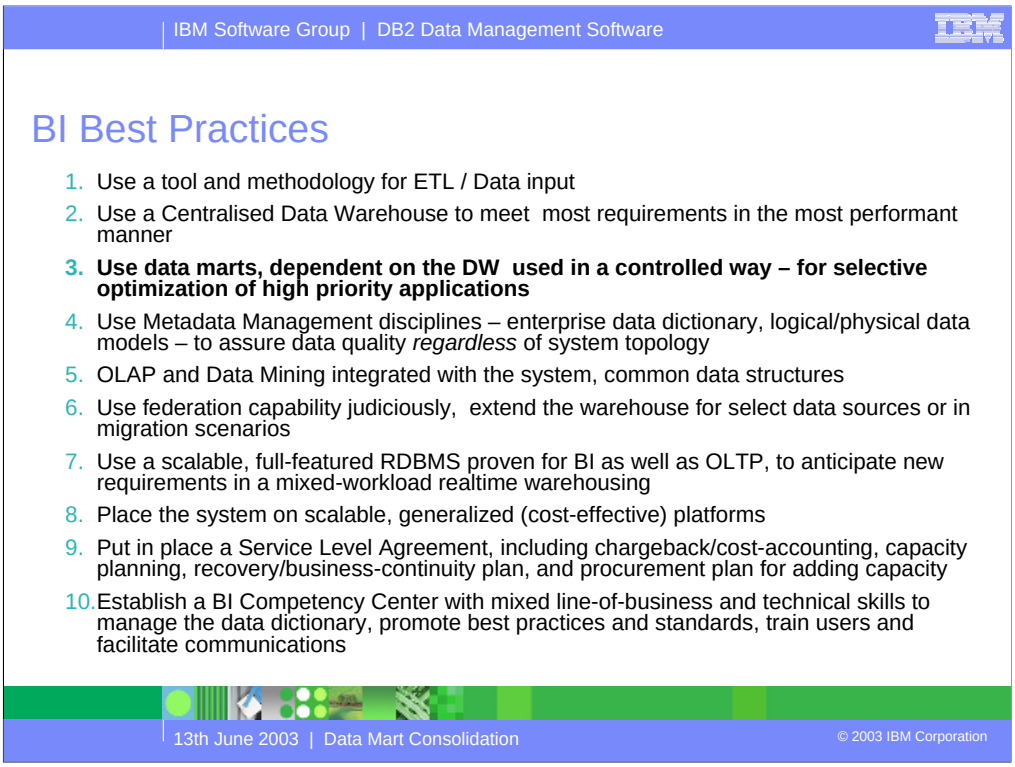
The bulk of the up front work is during the Design Study, where we look at the systems, taking a detailed inventory of those things to be consolidated and identifying how this would be done. We identify the methods needed for consolidation and establish both a business case and a firm project plan.

At the end of this phase we've either established that the consolidation is not possible, or not economically worthwhile, or more likely we now have a plan that we can be confident in executing to carry out the consolidation.

We may plan a major data modelling exercise into the project plan, if we are planning to reengineer the data.

Also we may be doing a Proof of Concept or benchmark if the consolidated system is going to be especially large or complex.

The remainder of the time is then the straightforward but significant work of consolidation – this is pretty much down to the number of person-hours put into it. We start at the beginning and after several weeks or months we have a consolidated system.



{Background:

BI has been around many years now

Many companies have systems that have given good service

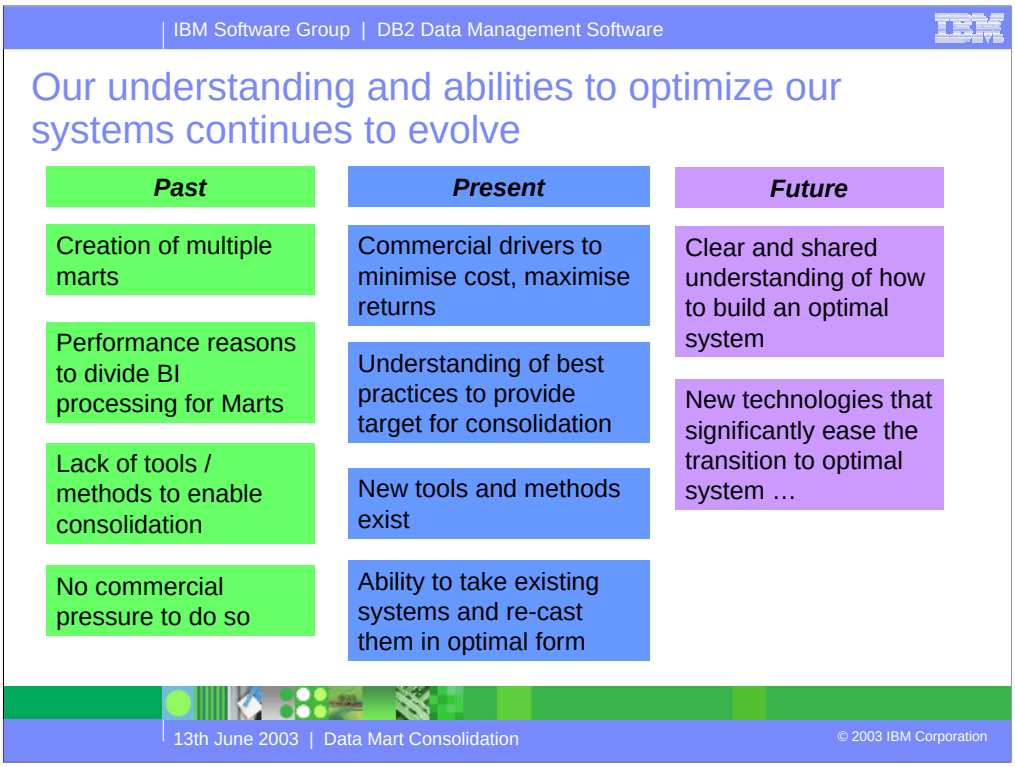
However, we now have new technologies and tools, and we have the benefit of hindsight to be able to see what represents best practice

Therefore there are opportunities to move to a BI system that incorporates best practice – producing a 3rd generation BI system

These best practices would include: (practice / benefit)

Outline best practices

The Best Practices recognise experience to date with current technology. I'll like to hand over now to Jon Rubin who'll talk about how we take this position forward using the unique technology that IBM has recently brought to market.



LINK FOIL

Past

We've seen how we got into this situation, there were good reasons for it, and frankly in the pat there was no commercial driver to do anything different, even if we wanted to the methods and tools didn't exist at the time to help us.

Present

Now however we have a pretty good idea of what an optimal system is – we've stopped the religious wars about how to build these systems and there is a good consensus emerging.

More than that there are now tools, methods and standards that let us build optimal systems – and also to be able to change them. I'm thinking here about Open Standards interfaces, the ability to hold, manipulate and use metadata, the isolation of the front and back end systems through tools etc.

Whilst the way we'd do this isn't automatic – the project is a systems integration project in its nature we do understand how to do this reliably to provide economic benefits.

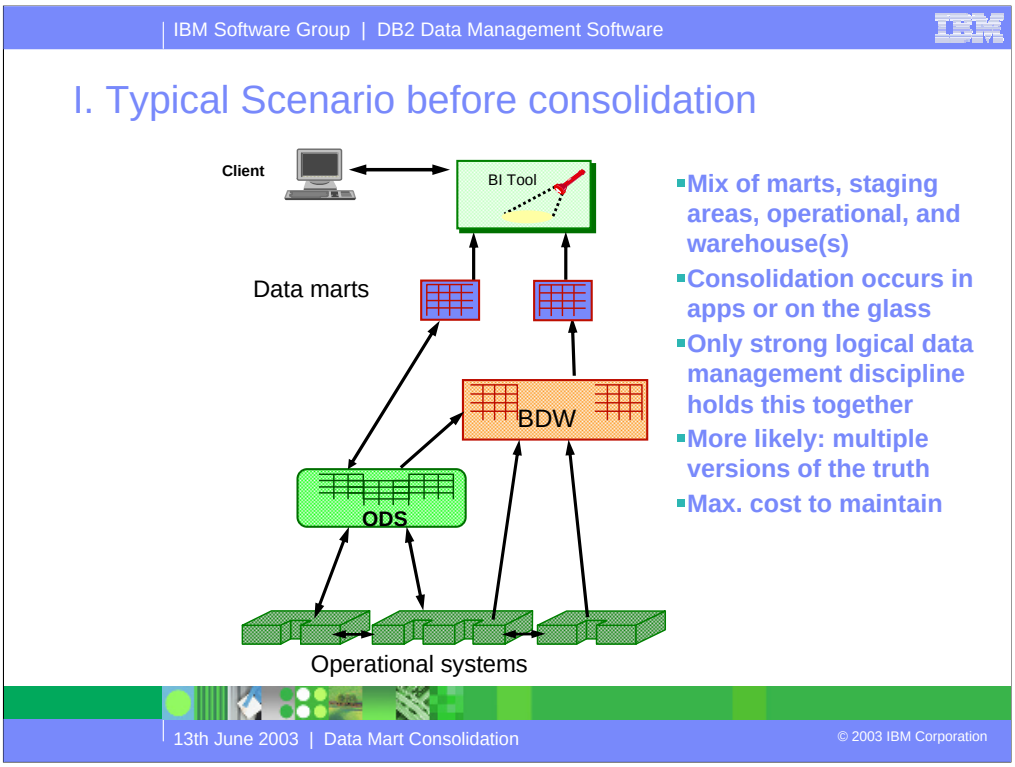
There is also a lot of pressure currently to provide the maximum returns from systems whilst minimising their costs.

Future.

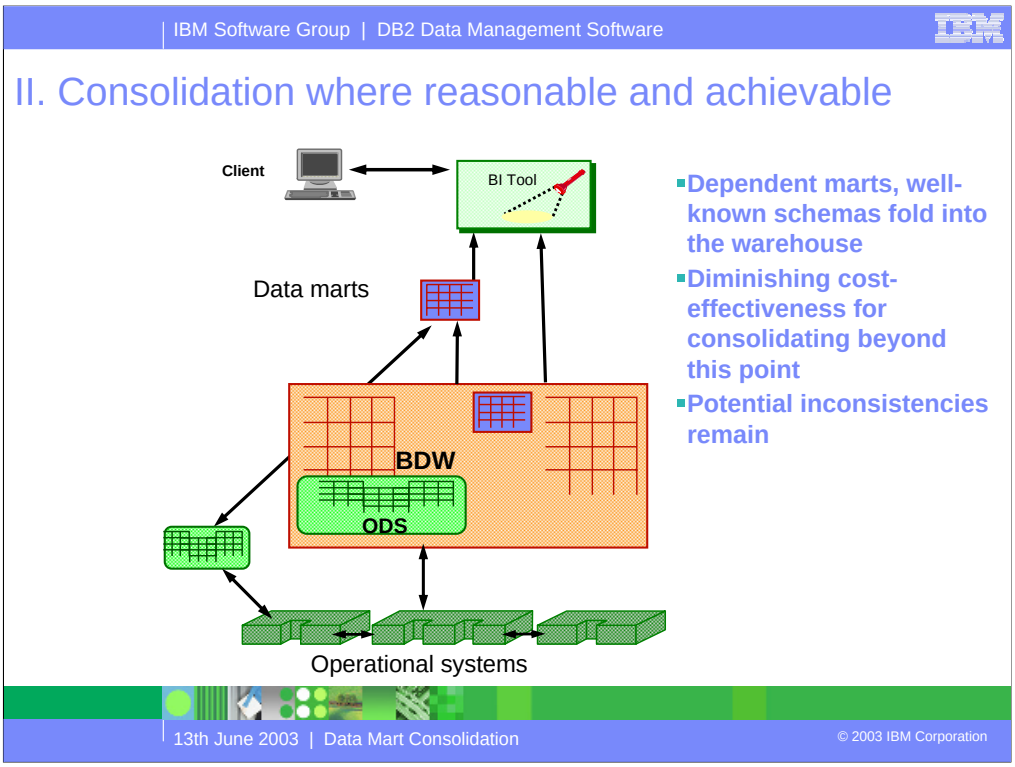
We would expect this to continue, in fact there are new technologies that we have just brought to market that we believe are especially relevant in this context

Consolidation At Different Levels

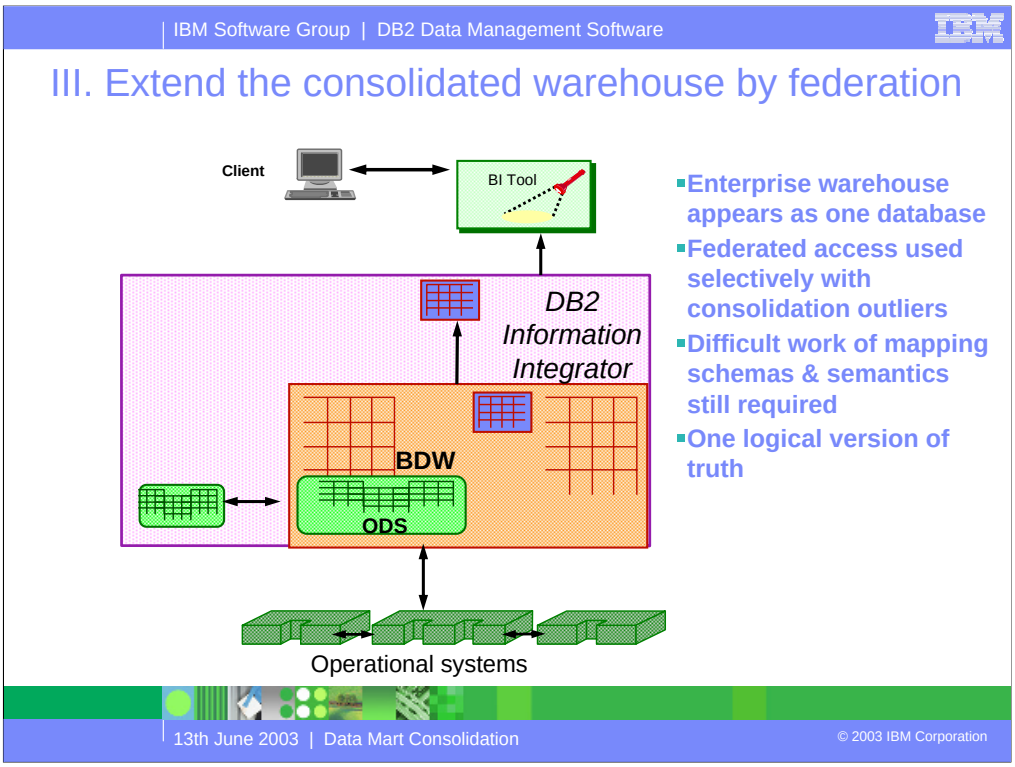
Level	How	Benefit
Logical Schema consolidation ***Treat all data as one database***	▪ **Federated Access and / or** ▪ **Combine physical models *(Note need to be abl e to provide common definitions / structure)***	▪ **Faster to access and manipulate data** ▪ **Easier development** ▪ **Faster to implement** ▪ **Benefits kick in earlier**
Physical Consolidation	▪ **Move data onto consolidation server** ▪ **Establish common data structures, e.g. Cube Views**	▪ **Consolidate server** ▪ **Lower equipment cost** ▪ **Use commodity hardware** ▪ **Lower maintenance cost** ▪ **Lower administration cost** ▪ **Lower software licence cost** ▪ **Position for capacity on demand**



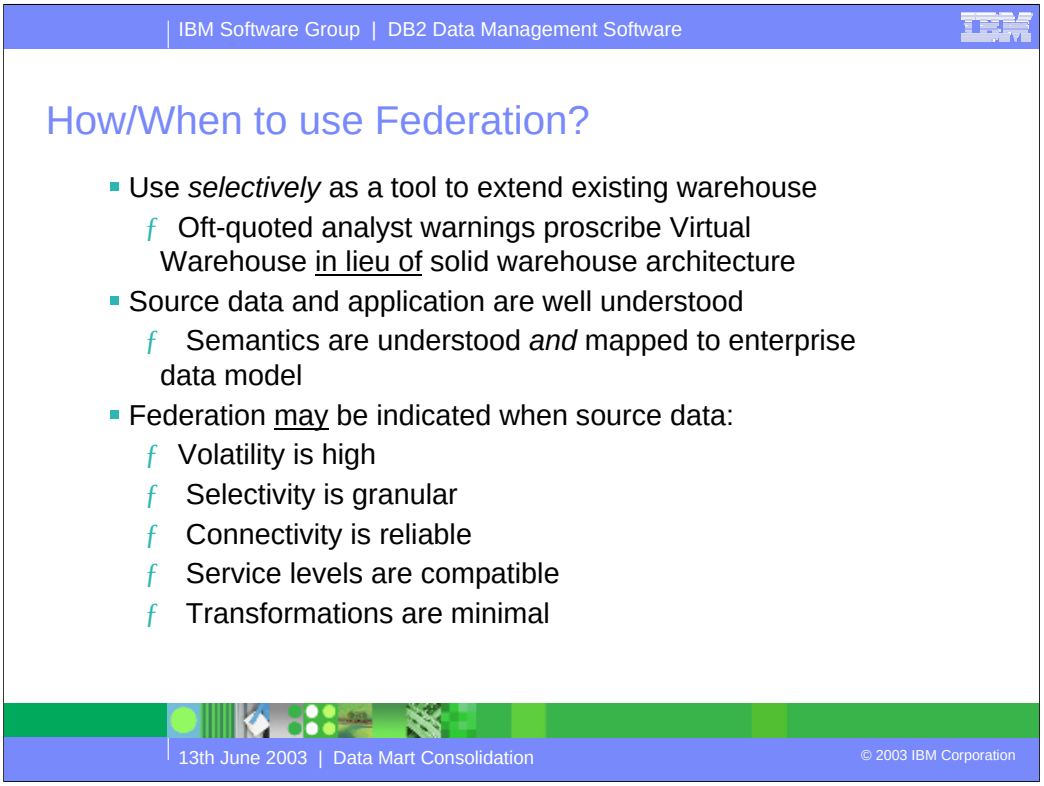
The typical large enterprise today is likely to have a confusing mix of marts, (multiple) warehouses, operational data stores and staging tables. That's just the DSS data infrastructure – there is still the entire range of transactional and operational data systems. Making sense of all this takes very strong skills and methodologies in the logical data disciplines: data modeling, data analysis, and data dictionaries. Yet this is one area of IT expertise where most enterprises are weak. Having a strong Information Resource Management function requires investing significant resources over a long period of time, for objectives which are often intangible or difficult to measure (e.g. data 'quality') and in any case are not directly tied to new revenue-generating applications. As a result, the common scenario pre-consolidation is a complex, costly to maintain and sub-optimal data topology.



Initial consolidation efforts often focus on dependent data marts, where – by definition – the schema mapping from warehouse to mart is well-defined. This means that folding the data models together is unlikely to uncover discrepancies in semantics, or require extensive analysis of data elements. As candidates for consolidation grow increasingly removed from the warehouse schema (e.g. independent marts) the potential for semantic differences increase, and required analysis becomes more difficult. At this stage enterprises must essentially perform the same logical data management tasks (modeling, analysis, dictionary administration) as they would have in building an enterprise data warehouse. Most efforts stop short of full consolidation, as per the EDW, when the increasing cost of consolidating data store 'outliers' outweighs the perceived benefits. Yet often there are good reasons why these remaining stores should NOT be physically consolidated.



Information Integration provides a useful tool for extending the partially-consolidated warehouse to encompass all of the DSS data topology. It lets applications and users 'see' a single rationalized schema, which can be queried as if it were a single large database. This simplifies development and realizes the benefits of the EDW in presenting a single version of the truth – although in this case it is a somewhat virtualized SVOT. But federation technology has limitations, and it should be applied judiciously. Most importantly, it does not eliminate the difficult work of schema mapping and the underlying logical data disciplines. If anything it is even more vital in this case, precisely because the temptation may be to short-cut that kind of foundation work of enterprise warehouse-building.



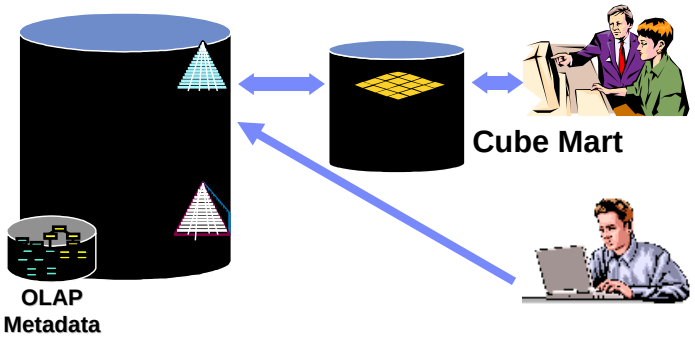
At IBM, we believe customers Information Integration requirements consists of distributed access (federation), but also consolidated access (Replication & ETL). This chart helps show how a customer would know when to use one or the other..but often they are used together. Example: Use ETL to build a warehouse, replication to keep it automatically updated on a scheduled basis and extend it with federation for queries that require data that it didn't make sense to put in the warehouse.

EII or distributed access approaches are indicated when

Access performance and load on source systems can be traded for overall lower cost implementation.

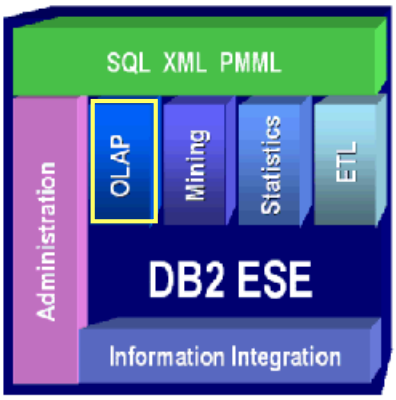
MOLAP Servers are a specialized kind of Data Mart

- **Optimized data platform for a given application**
- **Same drawbacks as proliferation of marts**
- **Same benefits for warehouse consolidation… *if HOLAP gap narrowed with DB2 Cube Views***

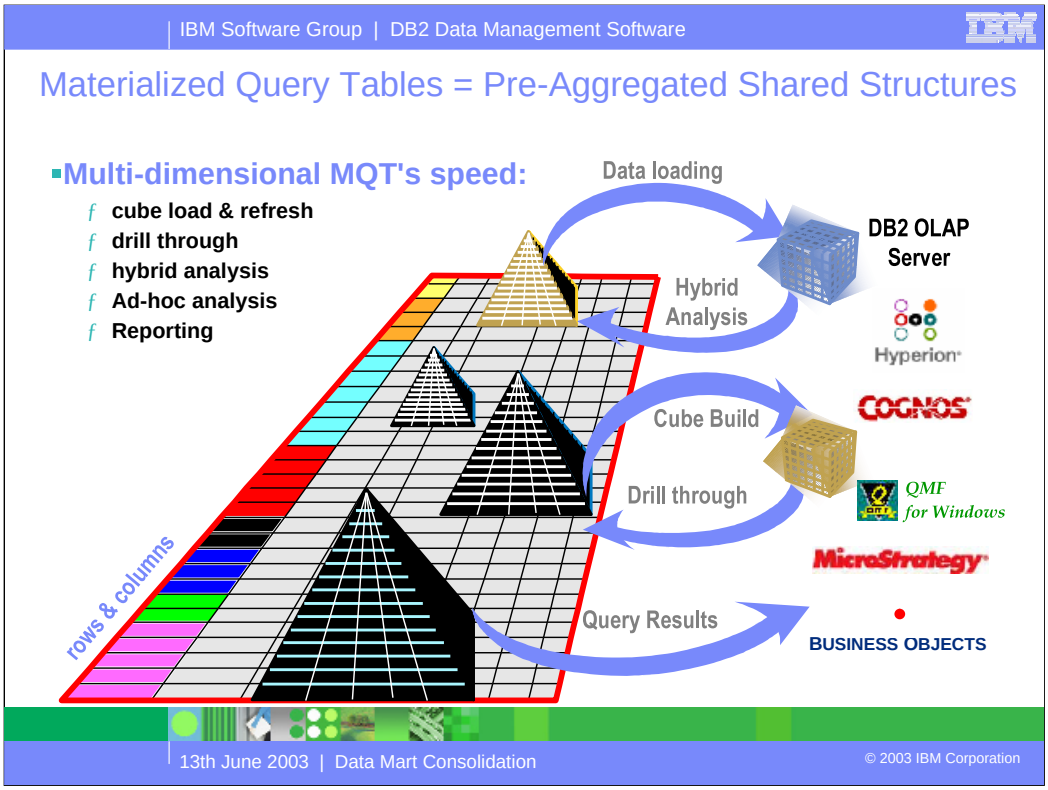


Unlike other data mart consolidation, we aren't suggesting the MOLAP server/cube mart goes away entirely. Partner tools based on MOLAP engines still offer important value in terms of performance, multi-dimensional navigation and high-end analytics. But Cube Views make it possible to begin to think of trade-offs between cube size (and therefore base levels) and specialized MQTs serving the same purpose in the warehouse.

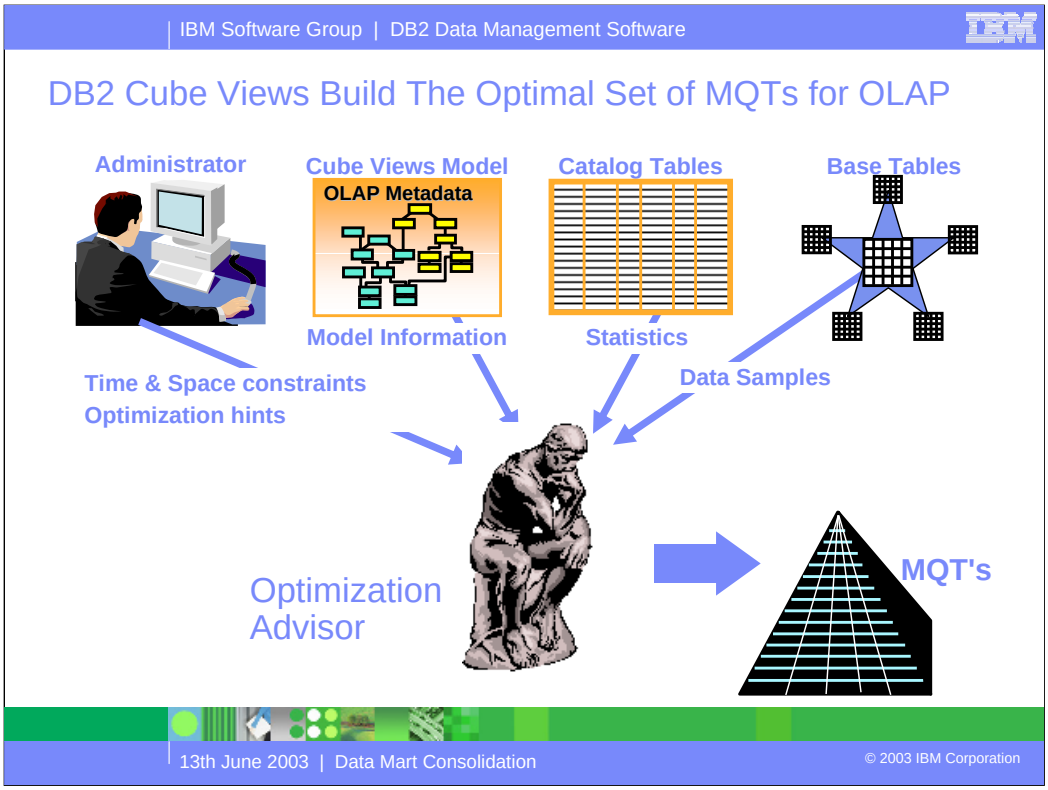
DB2 UDB Data Warehouse Edition



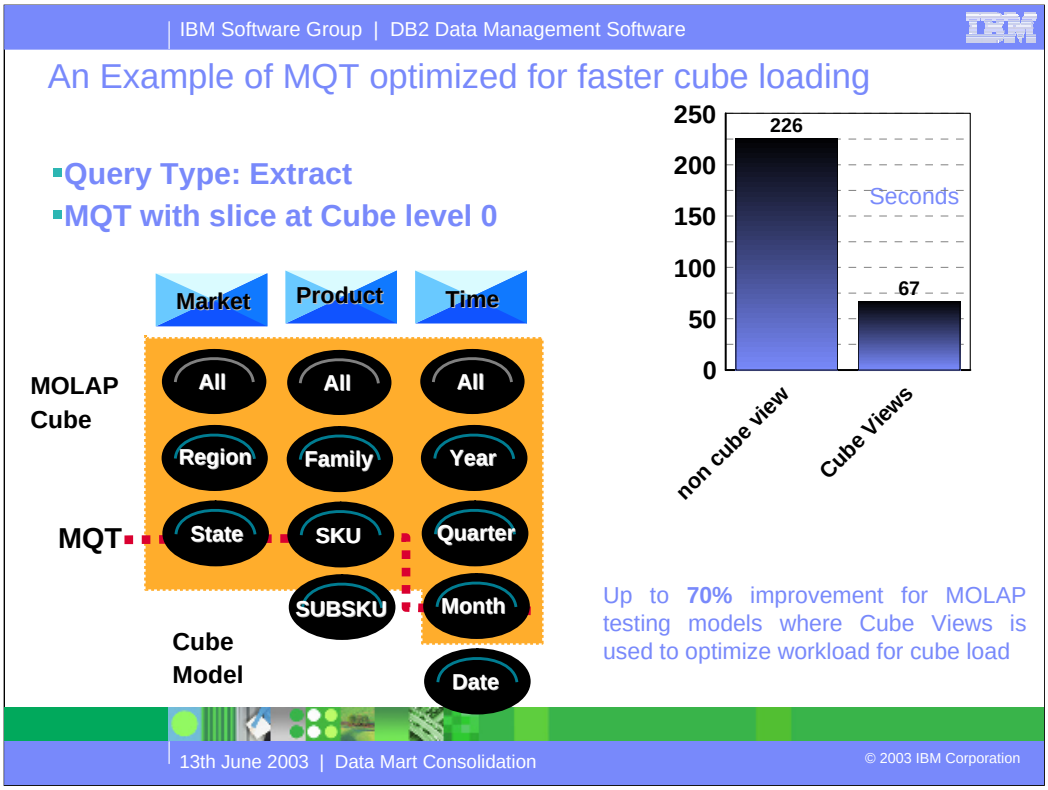
DB2 UDB Data Warehouse Enterprise Edition



this shows that Cube Views is DB2 infrastructure based on MQTs to speed a range of OLAP queries supporting nearly all business partners.



there is a design optimizer in DB2 Cube Views that considers many factors before recommending the optimal set of MQTs for a given OLAP meta-model and query types.

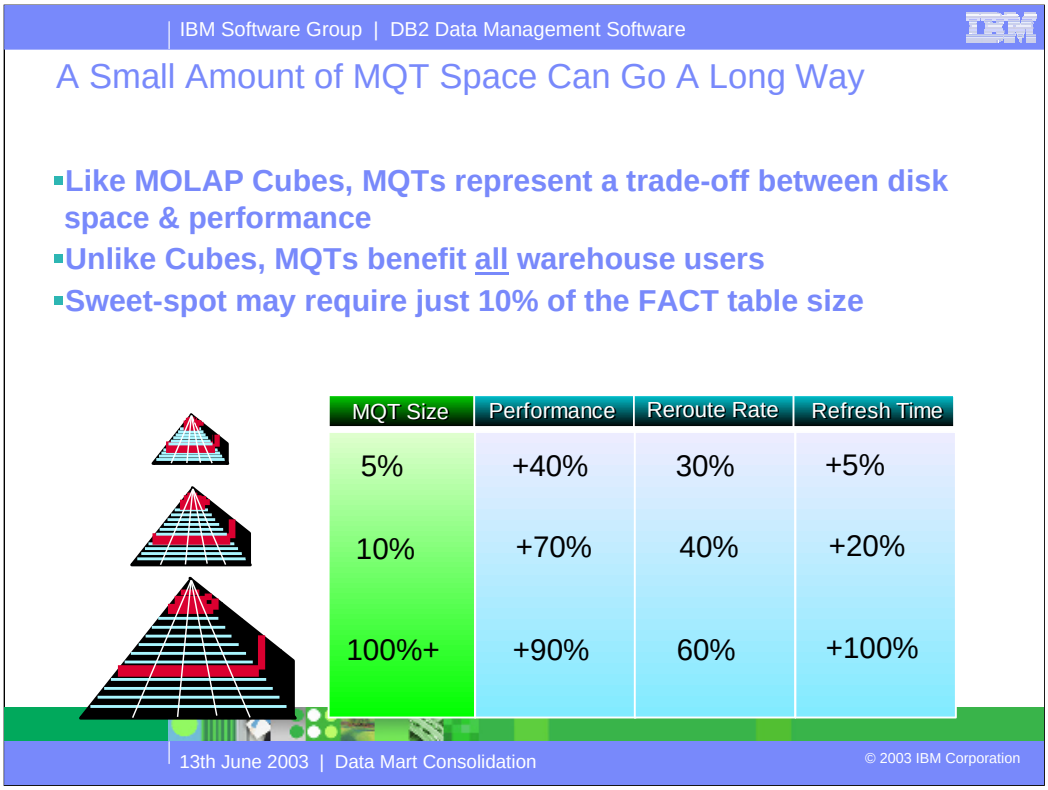


Inside Aurora, we anticipate four types of queries. The first type is to use Aurora to speed up the data extract process to load or refresh pure MOLAP cubes.

Here is an example, the orange area represents the DB2 OLAP Server cube.

Based on the Cube Model, and Cube object representing the Essbase Cube, A MQT will be created at level 0 of the cube.

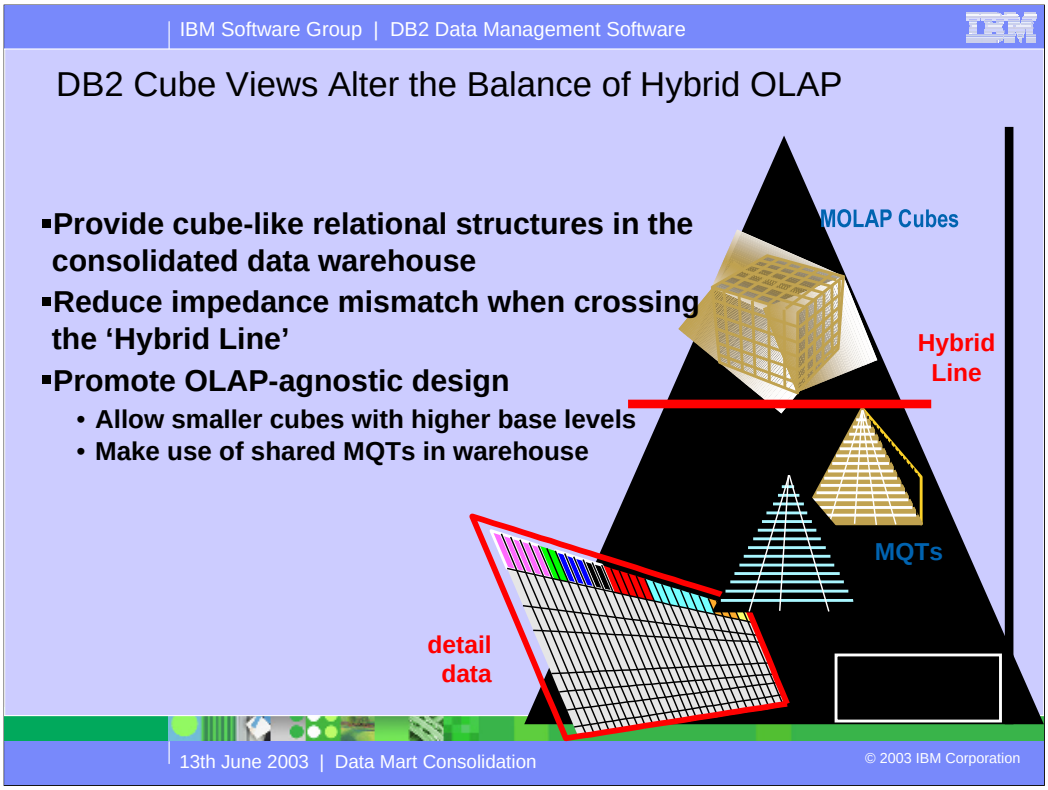
But what if the Essbase cube were designed one level higher in each dimensional hierarchy – at Region, Family, Quarter in this example – and the MQT provided State, SKU, Month as a hybrid drill-through. The cube could be smaller, load faster, and queries crossing the hybrid line would perform well thanks to the MQT. Plus other



Here is a few hints for configuring the disk space allocations.

Obviously, large MQT will have larger coverage but require more space and time to refresh

Based on our testing, it is a good idea to start at roughly 10% of the base fact table size. It provides good performance enhancements and a jump in sub-second queries, while requiring much lest refresh time when comparing with larger MQTs. But remember that disk saved in MOLAP cubes can be applied to equivalent MQTs. Is it possible to begin thinking in terms of a HOLAP continuum, where resources can be allocated between MOLAP marts and warehouse shared structures (MQTs) based on what makes the most sense for the overall warehouse community?



Cube Views takes a first step towards bridging the impedance mismatch when OLAP queries cross the hybrid line, making it possible to begin thinking in terms of a single OLAP design continuum bridging cube marts and the enterprise warehouse.

Summary

- Consolidation makes sense
- Faster response to business requirements
  - Earlier realisation of benefit = more benefit
- More economical support of BI processing
- Less maintenance
- Less machine resource required
- Consolidation is possible
- Costs paid for in savings in
  - Run time
  - Development
  - procurement

IBM Software Group

Thank You

DB2 Data Management Software

e business software

Back Up
Slides Follow