Expanding Data Mining Theory

for Industrial Applications

by

Aneeth Anand

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved January 2012 by the
Graduate Supervisory Committee:

Huan Liu, Co-Chair
Karl Kempf, Co-Chair
Arunabha Sen

ARIZONA STATE UNIVERSITY

May 2012

ABSTRACT

The field of Data Mining is widely recognized and accepted for its applications in many business problems to guide decision-making processes based on data. However, in recent times, the scope of these problems has swollen and the methods are under scrutiny for applicability and relevance to real-world circumstances. At the crossroads of innovation and standards, it is important to examine and understand whether the current theoretical methods for industrial applications (which include KDD, SEMMA and CRISP-DM) encompass all possible scenarios that could arise in practical situations. Do the methods require changes or enhancements? As part of the thesis I study the current methods and delineate the ideas of these methods and illuminate their shortcomings which posed challenges during practical implementation. Based on the experiments conducted and the research carried out, I propose an approach which illustrates the business problems with higher accuracy and provides a broader view of the process. It is then applied to different case studies highlighting the different aspects to this approach.

ACKNOWLEDGMENTS

I would like to acknowledge and thank my graduate advisor, Huan Liu and my thesis mentor, Karl Kempf who were very helpful and offered invaluable guidance, assistance and support. I would also like to thank my thesis committee member, Arunabha Sen for his advice. I would like to specially thank Morgan Dempsey without whose knowledge I could not have conducted the experiments of my thesis.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

With the expansion and globalization of the economy, there is a strong need to elevate the art of decision-making and have it behave more like science. This has given rise to a relatively new discipline in data engineering called Data Mining. The goal here is to extract empirical knowledge by sifting through data and developing patterns of the data behavior. The advent of data mining has spun numerous research activities and has resulted in formulating several methods. Several corporations have begun to implement and integrate data mining with their current systems to provide valuable solutions to business problems.

There has been a lot of interest towards mining techniques – of using linear and non-linear mapping techniques that could a) explain the past behavior b) predict the future with higher accuracy. This was very efficient considering that most of the initial problems studied were aimed at understanding the data available on hand. Hence the focus was on defining business problems and devising algorithms that could use the available data to give suggestive results. The data itself was not looked upon as an issue in such cases as it is available and could be trusted. So, most available theories that were formulated for industrial use have certain pre-conceptions about data.

However, with the advances in the field and with businesses trying to act more intelligently, the problems started to move away from ready sources of data, and hence there is growing interest on data gathering and preparation. My thesis attempts to take a closer look and help understand the practical

implementation of the mining process in the industries and the impact of data vis-à-vis results.

1.1. Problem Statement

With the role of data mining and decision modeling for business solutions having generated a lot of interest in the industries recently, it behooves us to understand the process as it is currently being implemented in the industries. Though there are some popular methodologies for the mining process, focus on understanding how these are executed drives the thesis. Primarily, my thesis focuses on answering the following questions. Do the conditions and criteria listed in the theoretical description of the mining process in the existing literature hold true in current industrial practice? What are the key differences between theory and practice? What could be expected and improved in current theoretical methods?

1.2. Challenges and Observations

When I initiate a practical data mining and modeling process to a defined business problem I encounter certain challenges. Some of them arise due to the differences between the theory and practice while some are part of the blind spots that the theoretical descriptions fail to cover. These are listed below:

- *Data Availability.* When one starts looking for the data pertaining to business problem, one could be surprised as the data may not be available, even if the requirement is for internal data. And so there is a need to look into data requirements from the beginning. Some of the reasons why data might not be available:

- Confidentiality – The data could be classified material, or supposed to be contained within specific departments and hence the access could be restricted.
- Distributed availability – The available data may be so disparate and may not have any direct correlation and hence it might be hard to understand it.
- Source Bias – The data obtained from a single source, especially if it is internal data, could run the risk of being biased or inaccurate. There could two cases of bias:
    - There is a lot of data, and one might need to select a subset of it to work on and the selection of subset could cause a bias
    - There is lack of certain data, and one might need to manipulate/extrapolate data to fill the needs, which could again lead to bias
- Quality - The data obtained internally could be of poor quality, i.e. sometimes with non-standard metrics and units, inconsistent arrangement of data etc., which would lead to intense rework on getting the data cleaned up for further use.

- *Data Vitality.* Defining and understanding a business problem is the first step in initiating a mining based solution, and modeling and deployment constitutes the final step these are often considered the most critical. However, based on my observations the process is highly dependent on the data and here is why:

- Not often does the data completely match the original business problem. But the business problem is then tweaked and redefined to the match the data available.
- Though it is perceived that the choice of the algorithm to be implemented is dependent only on the business problem, it actually has high dependence on the features that could be extracted from the available data.
- Data plays a vital role in determining the success or failure to solve a business problem.

- *Data Assurance.* When the data is gathered, one has few challenges:
  - How could one determine if all available data pertaining to the problem was gathered, and if not how does one measure the degree of completeness? This challenge leads us to answer three essential questions:
    - How much data is available?
    - How much does one need to solve a problem?
    - When to stop gleaning data?
  - Are the missing data as well as impact on the accuracy of the solution been accounted for? Typically this can be answered by triangulating the data from multiple independent sources. And is there a way to assess independence?
- *People.* One of the biggest entities that is absent in current theory is the people. Though mining is considered purely as a technical exercise, people involved constitute a variable in the process:

- A business person with ability to clearly define and tweak the business problem.
- A modeling person who can simulate and deploy a successful model
- A data person to access, mine, and organize data - Often enough, this cannot be performed single-handedly. In complex problems, this could be shared by a group of people, who may have to work as a unit and also coordinate with the bigger team.

Also, there are certain factors that might be needed to be taken into account. Some of them include:

- Location of the people – Where is each person(s) located and are they accessible to each other.
- Mode of communication – Are they able to frequently communicate with each other and what is the mode of communication.
- Level of expertise/knowledge – How many years of experience do each of them possess. And how much knowledge or understanding do they have of their area and other areas of the problem.

1.3. Contributions

The key contributions of my thesis include:

- The theoretical mining process is missing a way to establish confidence on the quantity of the data obtained. In order to achieve this I have

devised a measure that gives a degree of completeness. The idea behind this measure is when the data is gathered from two independent sources, the probability that data could be missing is inversely based on the commonality of data between the two sources. If data from source 1 is 'A' and data from source 2 is 'B', and 'C' be the data that is common then the total missing data is

$$M = \frac{(A - C)(B - C)}{C}$$

- I have developed a representation of the data mining process that is more in line with the actual implementation in industries. The key idea of this approach is the loopback communication between the three entities (Business unit, Data unit and Algorithm unit). The approach is described in detail in Chapter 3. Some of the objectives that were achieved through this representation are
    - Create a practical flow diagram between the different entities and states.
    - Define the conditions when the flow should go in a loop.
    - Define the conditions when the flow should stop and reach an end state.

CHAPTER 2

BACKGROUND AND RELATED WORK

The area of data mining is relatively new to the field of business intelligence, and hence lot of the methods and concepts have not been widely tested and implemented in the industries. However, there are some popular methods which have pioneered the use of data mining in industries. Out of several methods, three have been accepted and applied in the industries. Understanding these methods forms the base of data mining, but our approach looks at the same methods but putting a different perspective. In this chapter, I have explained the three methods in detail.

2.1. Overview of KDD

The Knowledge Discovery in Databases or KDD is one of the first method or process that was developed to make sense of data that were being stored. It is widely used when the requirement is to mine the data for pattern finding and extraction. The successful use of KDD in print media and scientific applications to help customers provide interesting perspectives based on data could be attributed for spawning the growing interest in data mining. Currently this method is used in Marketing, Investments and Fraud detection among other avenues.

The figure below gives an overview of the process. The KDD process is interactive and iterative involving user in several steps. Let's briefly go through these steps.

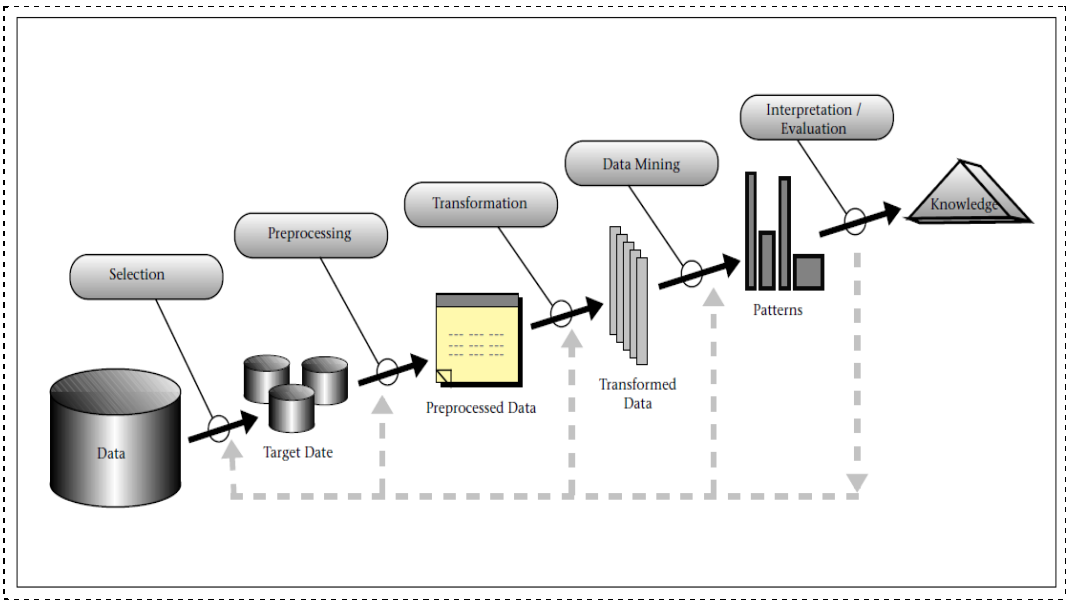


Figure 1. Illustration of the KDD process [7]

- *Selection.* This step comprises of the following sub steps
    - Understanding the application and identifying with the target customer's point of view, and
    - Selecting a target data set, on which the discovery is to be performed.
- *Preprocessing.* This step involves understanding the available data and making it process-friendly. It could be:
    - Cleaning up the data which involves noise removal,
    - Gathering the requisite information,
    - Handling the missing data features and records.
- *Transformation.* Here, the data is transformed by executing transforming methods, which include dimensionality reduction to eliminate unwanted features in the data.
- *Data Mining.* This is a critical step in the process. It involves

    - Matching our goals with a particular data mining method such as summarization, classification, regression or clustering;
    - Identifying models and parameters that would be appropriate for the data;
    - Searching for the patterns of interest and form a model (for classification or clustering)
- *Interpretation/Evaluation.* Understanding the pattern or model and loopback to any of the previous steps in case of searching for other possibilities.

2.2. Overview of SEMMA

An acronym that stands for "Sample, Explore, Modify, Model and Assess", SEMMA is an SAS based data mining solution package. Since it comes as a package backed by an industry leading statistical analysis tool, SEMMA could be used for visualization of data, select and transform the most significant features, model and qualify them. A depiction of the SEMMA process is shown below. Lets study the 5 steps of SEMMA in brief.

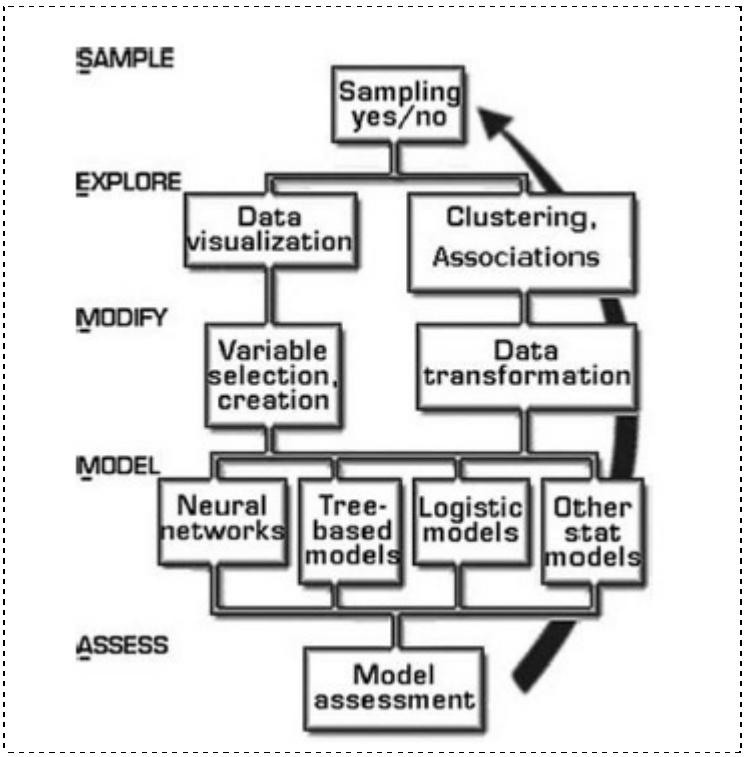


Figure 2. Illustration of SEMMA Process [15]

- *Sample.* This part of the process is to identify and possibly extract a subset of the original dataset that would be smaller yet should not to lose any significant information or features. This helps in reducing the processing time for the modeling process. This phase may also involve partitioning of data in case one choses to model using trained classification/clustering. The data could be partitioned into 3 parts: Training - for model fitting; Validation - to prevent over-fitting of model; Test - for honest assessment of model classification/clustering
- *Explore.* This phase is to explore the data by searching for unanticipated patterns or anomalies in order to gain better understanding of the data. This could be performed by plotting and visualization, if not by clustering the data. Especially useful in marketing campaigns.

- *Modify.* This phase is similar to the Transform stage of KDD. Here variables are selected from data source; eliminate some by transforming the data to model more efficiently.
- *Model.* This phase consists of modeling the data by having software select a modeling strategy that reliably model the data and provide maximum accuracy (proximity to desired values)
- *Assess.* This phase is to assess the pattern, model and data and study the findings, and estimate the performance of the model. This is where the test data could be more useful to find the accuracy and efficiency.

2.3. Overview of CRISP-DM

The CRISP-DM is the most popular method in the knowledge discovery process. Stands for CRoss Industry Standard Process for Data Mining, it is one of the most well-defined, documented, standardized methods for implementing data mining strategies in the industry. It is described in 6 basic steps and the diagram below is an illustration of the process.

Figure 3. Illustration of the CRISP-DM method [18]

The model consists of 6 steps which are summarized below:

- *Business understanding.* This stage focuses on understanding the objectives and view point from the business end. It is then formulated into a problem definition. It has several sub-steps:
    - Determination of business objectives
    - Determination of the data mining goals
    - Formulating a project plan
    - Putting together a team
- *Data understanding.* This stage starts with familiarization with the data and understanding problems with data. This can be subdivided into
    - Collection of data
    - Description of data
    - Exploration of data
    - Verification of data quality

- *Data preparation.* This stage does the preprocessing of the data that would be fed into the modeling tool. It includes
    - Selecting of data
    - Data clean-up
    - Structuring the data
    - Integration of data
    - Formatting of data
- *Modeling.* This stage is when the model is to be applied on the data. This stage could be subdivided into
    - Selection of modeling technique
    - Generating test design
    - Creating models
    - Assessment of modeling performance
- *Evaluation.* Once the model is analyzed for the data, it is then matched to the business requirements and evaluated for the results. At the end of this phase a decision of whether to use the model or not is made. It can contain the following steps
    - Evaluation of the match with business requirements
    - Process review
    - Determine whether to start from the first step or deploy the model
- *Deployment.* At this stage one is convinced the model could be deployed and make it available for the customer to use. This stage could contain
    - Deployment plan
    - Generating final reports

- Plan for monitoring and maintenance

2.4. Analysis of existing methods

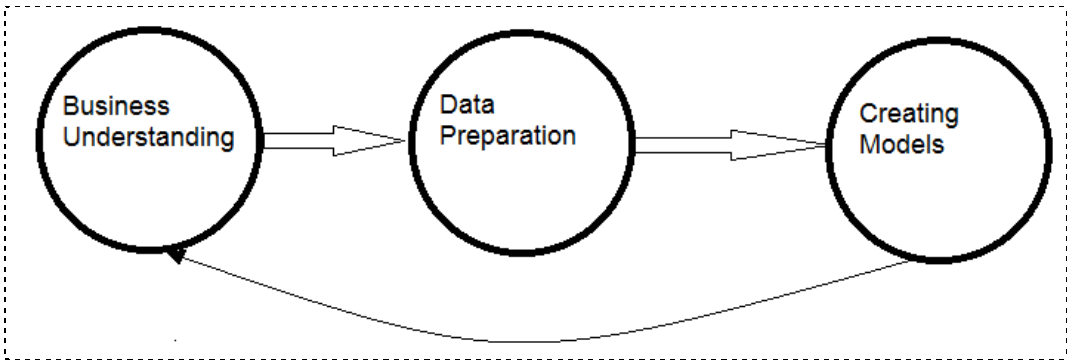


Figure 4. Generic illustration of the current methods

When the existing methods are studied, one could make a few observations:

- The current methods are modeled around the assumption that the data to solve a business problem is already available (though not readily) and those that are available are suitable to create models. Hence the data phase is focused mostly on grouping and organizing data.
- The present methods do not have an indication of failure in the process flow, which could be a limitation in practical implementation.
- There are a lot of common steps with few differences. In general, the existing theory revolves around three major phases and the deciding factor being the third phase, which is creating models. This pushes the idea that the success or failure in the decision making process depends on formulating a mathematical or statistical solution. The loopback on the process typically occurs only in the case of failure in the modeling phase.

However, my observation indicates that there are differences in the process illustration which drives this thesis.

CHAPTER 3

OUR APPROACH

The existing methods are fairly accurate in implementing a data mining model, but since they assume that the data is available and the available data is suitable, the existing methods cover only a subset of the real-world scenarios. They have a streamlined flow from the business problem to creating a model. However, they do not address to cases of failure in obtaining requisite data. One of the primary contributions of this thesis is to create a generalized model of data mining as applicable to the industries and providing a paradigm shift to how the flow of control is perceived. There could be three possible results based on data phase:

(1) *Immediate success.* Where a complete match between data requirement and availability/suitability is obtained. In this case, the flow is the same as detailed in the existing methods and proceeds to the modeling phase.

(2) *Immediate failure.* Where no match is found between the data requirement and availability/suitability. In this case the flow is halted.

(3) *Loopback.* Where a complete match between the requirement and availability of data is not found, but the available data warrants a look into the business problem and/or the modeling algorithms to see if they could be revised to accommodate the data. In this case, the flow loops back to the business problem.

Unlike in the existing approach where the flow is discussed more as a transition between different phases of the process, I envisioned my approach as a flow

between entities and states based on events. This facilitates in understanding some important aspects about the process:

- The flow is almost always in a loop and only reaches the end states upon termination
- The flow between different entities is also an indicator on the interaction between people
- The data plays a very major role in determining the success or failure of the problem

The illustration of the process and control flow in our approach is shown below:

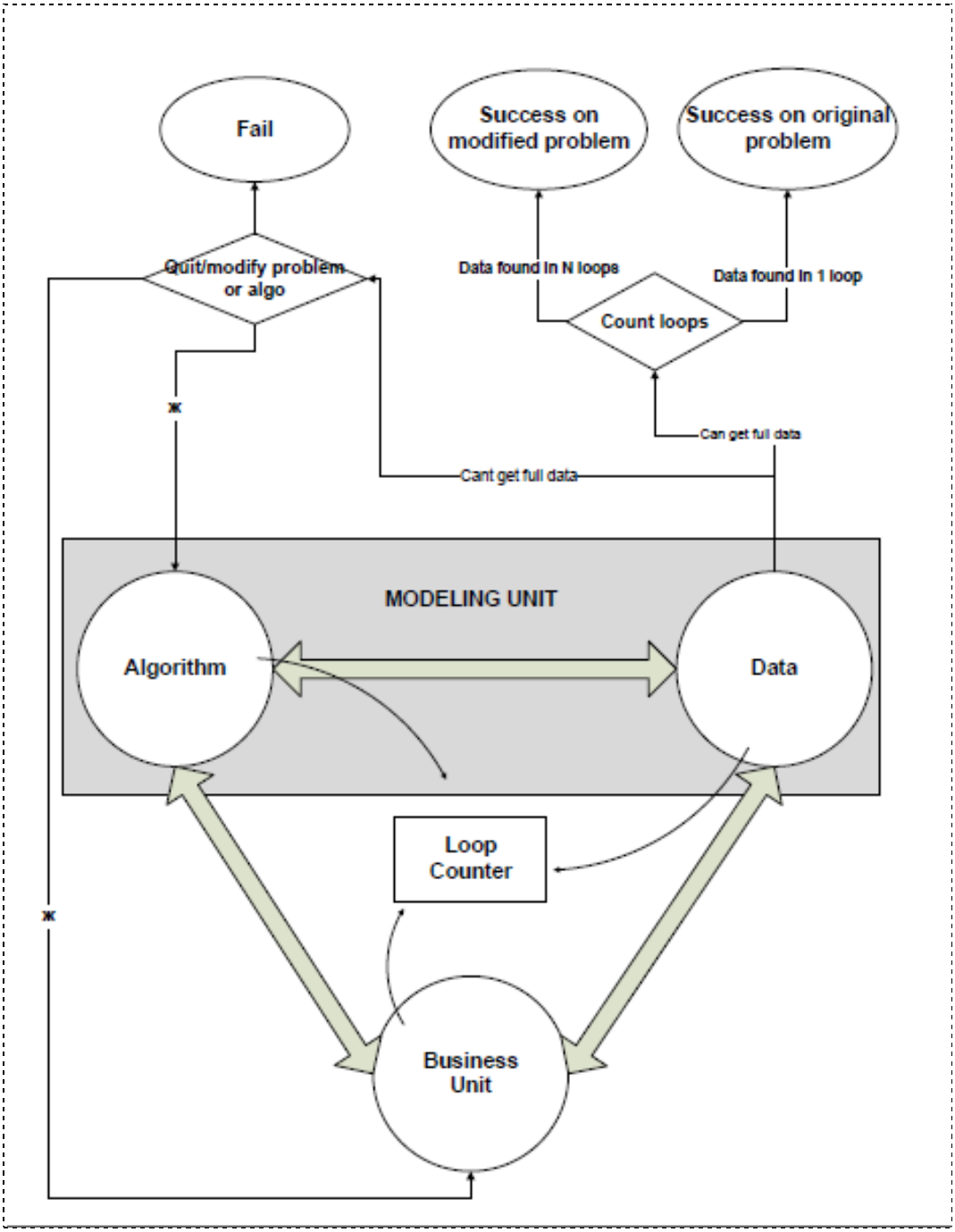


Figure 5. Illustration of process and control flow in our approach

3.1. Description:

In our model, there are 3 different entities which are basically a cluster formed based on the respective activities performed by people in the process. So, the entities comprise of the people and their respective activities. The 3 entities are:

- *Business unit.* This unit comprises of the business person(s) and the primary task would be to
    - Clearly define the business problem.
    - Constantly communicate with Data and Algorithm units to make sure they understand the requirements.
    - Redefine the business problem in case the prior attempt to find viable data fails.
- *Data unit.* This unit comprises of the people who have to transact on the data for the problem. The responsibilities include:
    - Study and Understand the data requirement of the problem, which would include research on
        - What data sources should be selected and access them?
        - What features in the data are of higher interest for Business/Algorithm unit?
        - How to prepare the data for modeling?
    - Extract, Transform and Load (ETL) the data
    - Scale and integrate the data
    - Perform Data qualification for quality of data, quantification for establishing confidence on the entirety of the data, and triangulation for validation of data.

    - Determine the degree of match between the business requirements of the problem to the obtainable data
- *Algorithm unit.* This unit comprises of people who are modeling experts and their task list would include:
    - Understanding the input and output features that would be provided for the data based on discussion with Business and Data Unit
    - Determining the right model/algorithm to be implemented on the data
    - Modify the algorithm in case the prior attempt to obtain the exact data for the problem fails.

The Data and Algorithm unit together make up the modeling unit.

The method also consists of 3 states:

- Success on original business problem – The data was successfully obtained for the initial problem defined.
- Success on modified business problem – the data was obtained that matches the business requirement, but after more than 1 iteration of revising the business requirement/algorithm.
- Fail – The data could not be obtained or was considered insufficient to proceed further.

The existing methods highlight only the flow towards Success on the original business problem.

3.2. Process description:

The business unit would be responsible for the finding and describing the problem. Team formation and establishing communication between the three entities initiates the first step. The communication loop is imperative in every step of the process. Once the business problem is defined, the data unit works to get the data through to the algorithm unit which then creates a model that could predict a desirable pattern from the data which would possibly help in the business decisions. The process would typically follow the various phases as described in the CRISP-DM model.

However, a decision on the success or failure of the process is determined at the end of the data phase. This is based on the degree of match between the obtained data and the business requirement.

- If the match is complete (‘1’), i.e. the complete data is obtained for the problem, and then the problem is deemed to be successfully solved. It is then classified either " Success on original business problem" or " Success on modified business problem" based on the number of iterations in the process
- If the match is poor (‘0’), i.e. there is no data available for the problem, then the problem is considered as "Failed"
- If the match is moderate (~0.01 to 0.99), i.e. there is some available data for the problem, then the problem falls into a fuzzy state. The questions that leads into the fuzzy state:
    - Is the data that is available inappropriate to address the business question? Does it have to be deemed a failure?

(or)

- Does one have enough data so that the algorithm could be modified to get a complete match between the business requirements and the available data? If so, then loopback (ж).

(or)

- Does one have enough data so the business problem could be tweaked which could provide a complete match? If so, proceed to loopback (ж).

The resolution of the fuzzy state is not performed by automated tools but by people from all the three entities. Hence I do not have a deterministic model to resolve the fuzzy state, but this is closer to explaining the practical implementation of the process.

3.3. Significance

- As mentioned earlier, the existing methods work with a presumption about data availability, and our model is more generalized. This mean that though the existing models are accurate in describing the modeling process, they just work on a subset of the bigger data problem. When I look at the generalized view, I can easily see that the existing model works with just one output state "Success on original business problem".
- It is fairly accepted and conclusive based on the experiments conducted and discussions with business and algorithm units, that very few times does one have all available data that could solve a practical business decision problem. The current data mining practices are useful in understanding and analyzing the available data, but most business

problems have moved away from these tasks and focus on large unseen pockets of data. Hence these existing methods may be directly applicable to 1/5th of the real world business problems and a large portion of it could be understood using the generalized approach.

CHAPTER 4

CASE STUDIES

To prove the basis of our approach, I worked on 3 separate cases each highlighting a different result based on the data phase, through my internship at Intel Corp.

Case 1 – *"To find the price-performance measure for Intel and AMD products"*

Case 2 – *"To determine the sales volume of Intel and AMD and provide a comparison of their market standings"*

Case 3 – *"To predict mergers and acquisitions among suppliers of Intel and its impact on its revenue"*

The first two case studies were two sub-problems to a bigger business problem - *"To see if the future behavior of Intel customers could be forecasted based on the past behavior".*

This problem is basically an endeavor to infer what the customers of Intel had done when a new product was introduced in the past. Could one find a pattern based on the data? If so, then it may be used to predict their behavior towards new product introduction, to a great extent.

The third case study was a separate business problem on its own where I attempt to model a prediction algorithm based on data from past mergers and acquisitions, and the financial standing of the companies.

Approach towards Case1 and Case2 –

Of the many aspects of this problem, for the purpose of study, I looked at two aspects:

1. Gather the price-performance points for all the products shipped by Intel and AMD

   – This would ideally give a performance trend of the Intel processors and its biggest competitor AMD in the market over a certain period. This could be juxtaposed with the customer buying trends of these products to get a correlation between the price-performance and purchases

2. Estimate the sales volume of Intel and AMD to determine the number of units sold

   – this could provide us with a sales trend of Intel and AMD processors over several quarters and when correlated with the OEM trends and market type to understand the behavior of different sales pockets over a certain period.

4.1. To predict Price-Performance for Intel & AMD (Case 1)

Problem Statement

To find the price-performance measure for Intel and AMD products

This is a data intensive problem, and though the task has sufficient history to work upon and perceived to have better sources, it is a non-trivial task and involves a lot of regression. Listed below are the steps taken to unravel the problem.

Data Source Selection

The first task in gathering data is selecting the sources for the data. For a problem such as this, where the data required is pertaining to the company I worked for, one would have believed that the data could be available internally.

However, I learnt that it was more prudent to look at outside sources because of the following reasons:

1. The access to internal data is not available because it is difficult to find the people who might control this data, and would have restrictions of disclosure even for internal observation.
2. The data could be biased or non-standard if the data is obtained from an internal source.

Hence, after discussing with the business unit, it was determined that the best move would be to take data from independent sources on the internet.
There were 2 sets of data sources which were required for this problem

(a) Data that provides a complete set of all the processors of Intel and AMD.
(b) Data that provides a standardized performance metrics for the processors.

There was a lot of discussion between the three units on what sources should I look for. During such discussions a few decisions were made which narrowed our focal point for data sources. First, the data sources should have freely available information as our thesis is based on mining patterns on data already available for public, and a derived reason being that it would involve a cumbersome procedure and involvement of financial resources to acquire funds to get data from paid services available with many market research companies. Second, I was looking for the highest credibility among the available sources and this is based on ranking the sources based on where they reference the data from. And thirdly, taking inputs from the business unit to acquire knowledge on

data sources selected for previous business problems, which would provide an empirical advantage. Based on all the reasons, I was able to filter the list of data sources and was able to pick the most appropriate ones.

For the first set of data, Wikipedia was chosen as the best source as it is unbiased since it is completely transparent, accurate since it references to publicly released information, and has free access to it as they are non-profit. For the second set of data, I chose data from 2 separate performance evaluation organizations, SPEC and TPC. The reasons why these two were chosen were:

- Both are independent and non-profit (not affiliated to a particular company)
- They publish performance measures that are based on standardized tests, and make it freely available
- TPC certifies the results and so it is more reliable
- SPEC has a room for flexibility in their validation but only reasonable and accurate data on performance are made available

Data Collection

Once the data sources were selected, the next step is to retrieve the data from these sites. The data retrieval is easier on paid websites, as they would provide API which offer data in XML format that could subsequently be read into database using x-query. However, since the data sources that were selected are free to access, they are not endowed with user-friendliness towards data access. This is especially true in the case of Wikipedia which is used for as the base to collect all processor related information. Hence, I implemented a PERL-based web crawler which uses regular expression (regex) matching of HTML tags to

retrieve the requisite data. The code sample that was used to retrieve the Intel Xeon Processor data is available in Appendix A.

At first glance of the site visually, the data looks tabulated and structured. However, upon looking at the source code it shows that there are several inconsistencies in the structure either due to the features of a particular product family or incoherent data or metric format. Hence I had to develop several permutations of the regex match criteria which obfuscated the crawler script. Once the data is retrieved it is exported to Excel spreadsheet.
For the performance data from SPEC and TPC, the data is available in the CSV format which is extracted into Excel spreadsheet.

Once the data was made available in spreadsheet, it needed to be filtered for duplicate records and the units of the measures needed to be standardized (for e.g., the frequency of some of the processors which were in GHz needed to be converted to MHz) to ease the process of integration. This was accomplished by developing VBA Macro routines that automates such activities in Excel
While data collection is being discussed, one should also heed to the features being collected. In other words, though there might be several features for each record of data, one could save considerably if the significance of each feature is understood in relation to the big picture. It will save a lot of time and effort if the focus was on the data feature selection. This is a key place where the role of communication between the data unit, business and algorithm unit becomes very critical. The importance of each data feature could be determined after formulating an algorithm, which would mean additional effort from both data unit and algorithm unit. However, discussion with the business unit helped remove

trivial data features which would have a minimal effect in the final solution, and hence helped save substantial effort at the data collection phase itself.

The table below shows the list of features collected from the data sources:

Table 1. List of features collected from different data sources

Feature	Wikipedia	SPEC	TPC
Model Name	X	X	X
Frequency	X	X	X
Cache size	X	X	
Release date	X		
Release price	X		X
Benchmarks		X	X
Cores		X	X
Chips/Codename[1]	X	X	
Base performance		X	X
Peak performance		X	

Data Integration

It is good to have data from several sources; however it provides a challenge when the data had to be integrated to form a unified data set. To integrate the data from these sources, I used SQL views and queries and used a SQL database as the storage entity for all data.

[1] The Codename feature was extracted from Wikipedia when the crawling and extraction process was repeated after a problem was discovered during the integration process. The codename feature was also an indicator for the number of chips in the product and hence acts as an indirect referencing to chips.

The excel spreadsheets are imported into tables. However the data required clean-up before matching and integration, as not all records are unique. For the data from Wikipedia, I found that several records which had the same model name, cache size and frequency. However, these could not be removed as duplicates as there was a difference between release date and prices. This is another instance where the communication channel between the data and business unit helped save significant effort towards the progress, though it initiated the rework of crawling and extraction from Wikipedia. It was discovered that though the records appear to be duplicates, they fall into different product families, and it could be distinguished by its "Codename". Hence the data collection was repeated and when the codename was included for each record and that helped resolve the issue of duplicate records on Wikipedia tables. The retrieval of Codename also played a major role in matching and scaling which will be discussed later in this section.

Queries were developed to eliminate duplicates from SPEC tables. The records which were exact matches were queried and removed easily. There were several records in the SPEC tables which had the same values for the Model Name and Chips, but had varied values for performance. To eliminate such discrepancy, using a query, I grouped all records based on Model name and number of Chips. And among the common ones, I selected the one which had the highest peak performance, and if it matches, pick the one with the highest base performance.

The next step is to merge these tables, and this presents a different challenge. The data from Wikipedia and SPEC were merged with the Model name as the pivot.

(1) This creates a cross-join of the frequencies from both these tables. When the frequencies match, I take the performance measure as is, but when there is a mismatch, the performance is scaled by interpolating the frequencies

(2) The number of chips in SPEC should also match the Wikipedia. The Wikipedia data do not have the number of chips. However, I was able to decode the market segment (Dual Processors, Multi-Processors or Uni-Processors) from the Codename of the product. And the market segment reflects the number of chips in the product. When the chips don't match, I performed a scaling operation defined by the business unit as described below:

The dual processor should account to 2 chips, multi-processor to 4 and uni-processor to 1 chip. The scaling factor for the number of chips to change by a factor of 2 in SPEC is '1.95'.

For example, when the number of chips in SPEC is 4 for a dual processor record found in Wikipedia, the performance should be scaled down by 1.95 (4 -> 2). If the number of chips in SPEC is 8, then the performance should be scaled down by 1.95^2 (8 -> 4 -> 2). Similarly, when the number of chips in SPEC is 2 for a multi-processor record in Wikipedia, the performance should be scaled up by 1.95 (2 -> 4), and if the number of chips in SPEC is 1, then it should be scaled up by 1.95^2 (1 -> 2 -> 4).

The scaling factor is determined by the people in the business unit. And the communication link with the business unit was critical in this phase.

Data Quantification

One of the key contributions in the thesis is to develop a confidence measure that could provide us a way to determine how much data have I acquired and how much I was missing. This is especially critical as I was working with free and publicly available data, and I needed some reliability measure to show that I have close to a complete dataset.

The idea behind the confidence measure is the use of probability. When one has data from two sources (A & B) of data, and if both are independent of each other, then let's assume the set C be the common data between A and B. Now the missing information M is given by

$$M = \frac{(A - C) * (B - C)}{C}$$

And if the confidence measure is to determined for source A, then

- The percentage of data lost/unattained = (M/A) * 100
- The percentage of data gathered = (A-M)/A * 100 [Confidence measure]

Applying this probabilistic theory on Wikipedia and SPEC datasets for Intel Processors, the missing data M was found to be approximately 51.67, and the confidence measure was found to be 90.95%. This also gave us the total processors T from the union of data from A and B (A∪B) to be 670.67. This was later cross-referenced with the person in the unit to verify the usage of this estimation method. It was found that the total processors in Intel's database was confirmed to be very close (~98%) to the derived number.

For AMD processors the missing data M was found to be 88.59, and gives us a confidence measure of 78.23%.

When these confidence measures were determined, and after discussing with the business and algorithm units, it was decided that the missing AMD processors needed to found elsewhere to increase the confidence measure up to or more than the confidence measure obtained for Intel.

Hence the process was repeated from the data collection phase for AMD processors, and found 20 records of data that were missing from the Wikipedia dataset, and were added to it. The revised value for M is 28.03 and the confidence measure is 93.44%

Data Qualification

The best way to examine the quality of data is to project the obtained data into a plot. When the performance data after integration is plotted in the graph it is verified against empirical knowledge of the business people to check if the trend matches the expectations.

There were four sets of plots generated for each benchmark,

- FP or Floating Point
- Int or Integer

} Measure of speed

- FPRate
- IntRate

} Measure of throughput

Each of these benchmarks would have 3 plots in a set, one for each market segment (Uni, Dual or Multi Processors). The graphs would be a plot the peak performance against the frequency. However, the frequency needs to be

scaled for both chips and cores in the record. Hence the x-axis frequency is scaled by this formula:

$$\text{Scaled\_freq} = \text{Chips} * \text{Cores per chip} * \text{Frequency}$$

Now the Chips values can be determined based on market segment:

- Uni: Chips = 1
- Dual: Chips = 2
- Multi: Chips = 4

At first pass, these plots were generated. However upon inspection, the FP and Int plots were plotting to be higher than expected values. It was after the discussion with the business unit that it was discovered that for FP and Int benchmarks, the number of chips do not affect the performance measures. Hence the Scaled_freq parameter was computed without scaling for number of chips. Only for FP and Int plots, the scaling formula was revised to:

$$\text{Scaled\_freq} = \text{Cores per chip} * \text{Frequency}$$

The plots used for qualification are shown below.

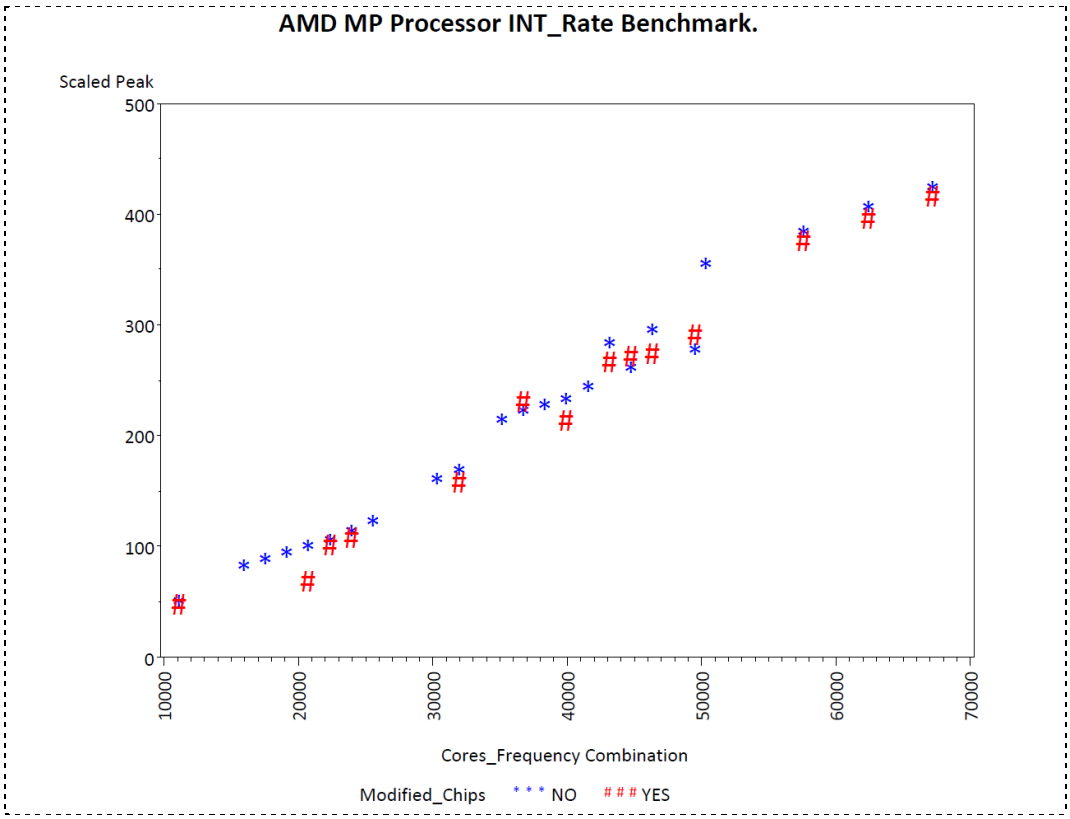


Figure 6. Core Frequency vs. Performance for IntRate benchmark

The processors with the performance value greater than 300 has 24 cores with 6 cores per chip falling under the family name 'Istanbul' as compared to the group 'Shanghai' that has 16 cores in them.

Table 2. List of all features for the family 'Istanbul'

wiki_processor	wiki_CodeName	wiki_frequency	Wiki_L2cache	wiki_core	Frequency_MHz	L2Cache_KB	Chips	Cores	Cores Per Chip	Scaled_Peak	Modified_Chips	xaxis
Opteron 8425 HE	Istanbul	2100	3072	6	2100	512	4	24	6	351	NO	50400
Opteron 8431	Istanbul	2400	3072	6	2400	512	4	24	6	380	NO	57600
Opteron 8435	Istanbul	2600	3072	6	2600	512	4	24	6	402	NO	62400
Opteron 8439 SE	Istanbul	2800	3072	6	2800	512	4	24	6	420	NO	67200

wiki_processor	wiki_CodeName	wiki_frequency	Wiki_L2cache	wiki_core	Frequency_MHz	L2Cache_KB	Chips	Cores	Cores Per Chip	Scaled_Peak	Modified_Chips	xaxis
Opteron 8374 HE	Shanghai	2200	2048	4	2200	512	4	16	4	211	NO	35200
Opteron 8376 HE	Shanghai	2300	2048	4	2300	512	4	16	4	219	NO	36800
Opteron 8378	Shanghai	2400	2048	4	2400	512	4	16	4	223	NO	38400
Opteron 8379 HE	Shanghai	2400	2048	4	2400	512	4	16	4	224	NO	38400
Opteron 8380	Shanghai	2500	2048	4	2500	512	4	16	4	229	NO	40000
Opteron 8381 HE	Shanghai	2500	2048	4	2500	512	4	16	4	229	NO	40000
Opteron 8382	Shanghai	2600	2048	4	2600	512	4	16	4	240	NO	41600
Opteron 8384	Shanghai	2700	2048	4	2700	512	4	16	4	280	NO	43200
Opteron 8386 SE	Shanghai	2800	2048	4	2800	512	4	16	4	257	NO	44800
Opteron 8387	Shanghai	2800	2048	4	2800	512	4	16	4	253	NO	44800
Opteron 8389	Shanghai	2900	2048	4	2900	512	4	16	4	292	NO	46400
Opteron 8393 SE	Shanghai	3100	2048	4	3100	512	4	16	4	274	NO	49600

Similarly,

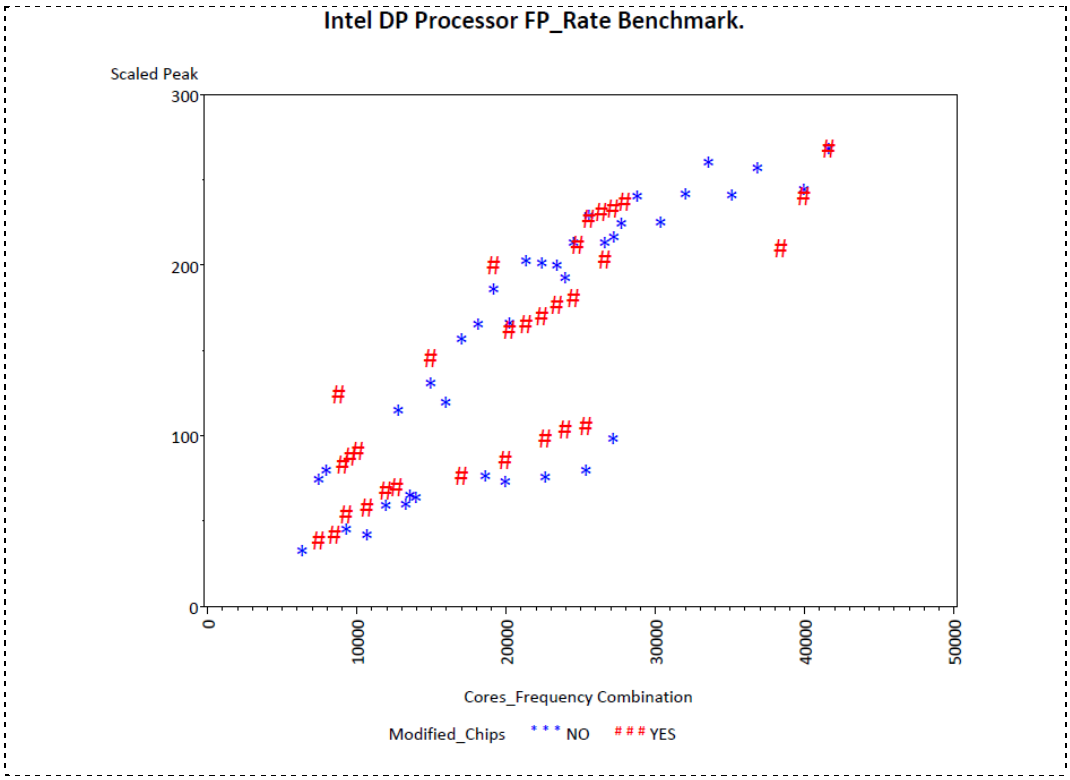


Figure 7. Core Frequency vs. Performance for FPRate benchmark

The plot above shows a distinct differentiation between two different architectural groups 'Gulftown' & 'Gainestown' and others.

Table 3. List of all features for the family 'Gulftown' and 'Gainestown'

wiki_processor	wiki_CodeName	wiki_frequency	Date	Scaled_Peak	xaxis
Xeon E5603	Gulftown	1600	12/01/2010	113	12800
Xeon L5609	Gulftown	1870	01/01/2011	128.27438371	14960
Xeon E5504	Gainestown	2000	07/01/2010	117	16000
Xeon E5506	Gainestown	2130	04/01/2010	119.83122363	17040
Xeon E5606	Gulftown	2130	05/01/2011	136.80731364	17040
Xeon L5506	Gainestown	2130	07/01/2010	119.83122363	17040
Xeon L5518	Gainestown	2130	05/01/2010	154.78199719	17040
Xeon L5630	Gulftown	2130	03/01/2010	124	17040
Xeon E5507	Gainestown	2270	02/01/2011	131.17335686	18160
Xeon E5520	Gainestown	2270	01/01/2010	159	18160
Xeon E5607	Gulftown	2270	04/01/2011	138.61061947	18160
Xeon L5520	Gainestown	2270	06/01/2010	163.21570357	18160
Xeon E5530	Gainestown	2400	06/01/2010	167	19200
Xeon E5620	Gulftown	2400	05/01/2011	184	19200
Xeon L5530	Gainestown	2400	06/01/2010	167	19200
Xeon E5540	Gainestown	2530	08/01/2009	163.74112076	20240
Xeon E5630	Gulftown	2530	06/01/2010	160	20240
Xeon E5640	Gulftown	2670	03/01/2010	177	21360
Xeon X5550	Gainestown	2670	01/01/2010	200.22497188	21360
Xeon X5560	Gainestown	2800	04/01/2010	199	22400
Xeon X5570	Gainestown	2930	04/01/2009	197.73006135	23440
Xeon X5647	Gulftown	2930	02/01/2011	195.79952267	23440
Xeon L5638	Gulftown	2000	08/01/2010	190	24000
Xeon X5667	Gulftown	3070	04/01/2010	211.27527723	24560
Xeon W5580	Gainestown	3200	04/01/2009	207	25600
Xeon X5672	Gulftown	3200	04/01/2011	227	25600
Xeon W5590	Gainestown	3330	10/01/2009	210.81008101	26640
Xeon L5640	Gulftown	2270	03/01/2011	214.28319365	27240
Xeon X5677	Gulftown	3470	04/01/2010	222.19209691	27760
Xeon E5645	Gulftown	2400	12/01/2010	217	28800
Xeon X5687	Gulftown	3600	02/01/2011	238	28800
Xeon E5649	Gulftown	2530	05/01/2011	222.7358863	30360
Xeon X5650	Gulftown	2670	05/01/2010	239.35858965	32040
Xeon X5660	Gulftown	2800	05/01/2011	258.38596491	33600
Xeon X5670	Gulftown	2930	12/01/2010	238.86401604	35160
Xeon X5675	Gulftown	3070	03/01/2011	254.33137639	36840
Xeon X5680	Gulftown	3330	04/01/2010	242	39960
Xeon X5690	Gulftown	3470	03/01/2011	266.30698211	41640

Table 4. List of all features for other families

wiki_processor	wiki_CodeName	wiki_frequency	Date	Scaled_Peak	xaxis
Xeon 5110	Woodcrest	1600	11/01/2007	30.1	6400
Xeon 5120	Woodcrest	1870	05/01/2007	32.351901446	7480
Xeon E5205	Wolfdale-DP	1870	02/01/2008	38.083288555	7480
Xeon L5215	Wolfdale-DP	1870	01/01/2009	40.565077665	7480
Xeon 5130	Woodcrest	2000	11/01/2007	40.1	8000
Xeon 5140	Woodcrest	2330	11/01/2007	42.5	9320
Xeon 5150	Woodcrest	2670	01/01/2007	39.84924812	10680
Xeon 5160	Woodcrest	3000	11/01/2007	48.8	12000
Xeon L5240	Wolfdale-DP	3000	11/01/2008	56.7	12000
Xeon E5310	Clovertown	1600	11/01/2007	43.217286915	12800
Xeon L5310	Clovertown	1600	12/01/2007	46.7	12800
Xeon X5260	Wolfdale-DP	3330	02/01/2008	57.286015038	13320
Xeon X5272	Wolfdale-DP	3400	05/01/2008	62.465349454	13600
Xeon X5270	Wolfdale-DP	3500	11/01/2008	61.5	14000
Xeon E5320	Clovertown	1870	11/01/2007	47.776647027	14960
Xeon L5320	Clovertown	1870	03/01/2007	45.67327263	14960
Xeon E5335	Clovertown	2000	02/01/2008	57.2	16000
Xeon E5405	Harpertown	2000	01/01/2008	63.495242864	16000
Xeon L5335	Clovertown	2000	12/01/2007	58.4	16000
Xeon L5408	Harpertown	2130	07/01/2008	57.019690577	17040
Xeon E5345	Clovertown	2330	06/01/2007	57.2	18640
Xeon E5410	Harpertown	2330	05/01/2008	71.8	18640
Xeon L5410	Harpertown	2330	04/01/2009	74.104586369	18640
Xeon E5420	Harpertown	2500	01/01/2008	69.995988769	20000
Xeon L5420	Harpertown	2500	11/01/2008	71.1	20000
Xeon E5430	Harpertown	2670	01/01/2008	71.880764904	21360
Xeon L5430	Harpertown	2670	11/01/2008	75.985376828	21360
Xeon X5355	Clovertown	2670	06/01/2007	60.727443609	21360
Xeon E5462	Harpertown	2800	05/01/2008	84.4	22400
Xeon E5440	Harpertown	2830	01/01/2008	73.503892427	22640
Xeon E5450	Harpertown	3000	11/01/2008	78.4	24000
Xeon E5472	Harpertown	3000	05/01/2008	88.235294118	24000
Xeon X5365	Clovertown	3000	11/01/2007	67.7	24000
Xeon X5450	Harpertown	3000	12/01/2007	77.6	24000
Xeon X5472	Harpertown	3000	11/01/2008	87.6	24000
Xeon X5460	Harpertown	3170	01/01/2008	77.292590247	25360
Xeon X5482	Harpertown	3200	12/01/2008	94	25600
Xeon X5470	Harpertown	3330	11/01/2008	84.523852385	26640
Xeon X5492	Harpertown	3400	11/01/2008	96.4	27200

4.2. To predict sales volume for Intel & AMD (Case 2)

Problem Statement

"To determine the sales volume of Intel and AMD and provide a comparison of their market standings"

At the outset, the problem looks straight forward and looks like a simple task to complete. However, it is not a trivial problem, as the sales volume of the companies is not easily accessible, and requires strong communication and understanding of the various requirements at several stages.

The steps taken towards achieving our solution are described below.

The procedure

Step 1. Determine the sales volume for Intel and AMD through the market share information that are released quarterly and annually. However, upon research it is determined that the market share information is released as a percentage of the total volume of sales in the microprocessors industry, and there are no direct ways to obtain this information which leads to the next step.

Step 2. Determine the sales volume for all major OEMs (Original Equipment Manufacturers), as both percentage of market share and as number of units sold on a yearly and quarterly basis. This could enable us to estimate the number of units sold by Intel and AMD to their customers (OEMs) during this period.

Step 3. Determine the sales volume of Intel and AMD per market segment (i.e. Desktop, Mobile or Server). This would entail collecting percentage market share of Intel or AMD per segment. Then the number of units sold by OEMs per segment should be found. This data again would be again followed on a per-quarter and per-year basis.

Step 4. Determine which processor architectures (or families) of both Intel and AMD are contributing for their sales of each segment.

The process

Listed below are the steps in the process of obtaining the data to solve the problem.

Data source selection

One of the first challenges to the problem is obtain the sales volume of data and make sure the sources have highest credibility and reliability and has sufficient history in publishing sales data as it would provide consistency when accessing data over longer period (in our case, > 5 years). The reason why source selection is critical is that it sales data are easily susceptible to:

- Non-standard metrics – The way sales figures are calculated could be radically different between companies and there could be deficiencies in methodology.
- Misrepresentation – There may be cases where sales figure could reflect the billing volume for the quarter and not the actual sold volume.
- Bias – Data from certain forums could be biased towards one of the supplier companies which could affect reliability.

Hence, keeping all these points in consideration the sources were selected after research and discussions with the business unit.

To obtain data on market share of Intel and AMD (Step 1 in the procedure), I selected two companies

- IDC - market research and analysis firm specializing in information technology, telecommunications and consumer technology.
- Mercury Research - The PC industry's primary source for market information on PC Microprocessors, System Logic Chip Sets, and Graphics Components.

To acquire the OEM market share information both on an overall and per-segment basis (Step 2 and 3), I used several sources:

- Gartner – is an information technology research and advisory firm and has set a standard for information research over a long period.
- HP, Dell, Acer – press notes and releases on their quarterly performances.

All the sources provide market share and sales volume information as a paid service. However, since I require freely available data, I relied on their quarterly and annual releases of the market share report for the microprocessors.

Establishing Loopback

However, there was a setback when I was unable to identify a source to obtain the contribution of processor architecture towards sales volume of the Intel and AMD products by segment.

At this juncture, it was discussed with the business and algorithm unit, and a decision was to be made on whether to continue with the process or halt it due to insufficient data. Since there is no control metric that could guide us to a choice, it was replaced by several rounds of discussion by the people in the three units to figure the next step and make a call on whether the requirement for the solution should be changed or should the problem be dropped as a failure. It was well established that a high percentage of data can been retrieved that warrants a look into the algorithm and/or revision to the business problem to achieve results.

The algorithm could not support the initial business requirement based on the available data. However, upon further investigation it was determined that though I might not get the actual distribution on the architecture information, it was possible to arrive at an approximate estimation on the data combining:

- Data collected in the previous case study - the codename (product family name) and the release date obtained from the previous data set could be used
- Wikipedia source that enables correlating product family name with their architecture
- Heuristic model that explains the market size distribution from the time of the product launch. This was obtained from prior business knowledge.

Hence, it was possible to go back and change the business requirements from finding the actual sales volume distribution based on architecture, to having an estimation of the same. And the data obtained found to match this requirement.

This whole exercise, though might look as a simple procedure, was instrumental in defining the generic model that was described in Chapter 3. Also, it made us realize that there exists a fuzzy state in the decision making when not all required data could be collected.

Data Preparation

The most rigorous part of the data collection process is to identify web pages of press releases from the sources. The reason this could be tricky is that several articles try to forecast the trend and are not the actual results. Once these pages are identified, then the data is imported into a spreadsheet for analysis.

Data Triangulation

One of the key aspects to the process is the triangulation of the data. This needs to be achieved for the following reasons:

- Reliability – When one looks at a data from a single source, there are chances that the accuracy of the data could be compromised. Hence when data from more than one source is selected it could be used as a cross-reference for our findings from first one. It gives a better sense of accuracy and precision
- Packing – The data that is obtained could be missing certain information for a particular period (e.g. the release information for a particular quarter or a year could be missing), or could miss on providing granularity (e.g. the information could be given annually, but might miss the quarterly data). In such cases, data from other sources could be used to fill those blanks, and help us analyze the trend without fear of any missing links.

Data Qualification

To perform data qualification, I did a simplistic data modeling where the number of units sold by Intel and AMD was calculated over each quarter from the percentage market share obtained and the total number of units sold by OEMs during those quarters. This is then verified with the business unit to check if the data reflects the historical and empirical facts.

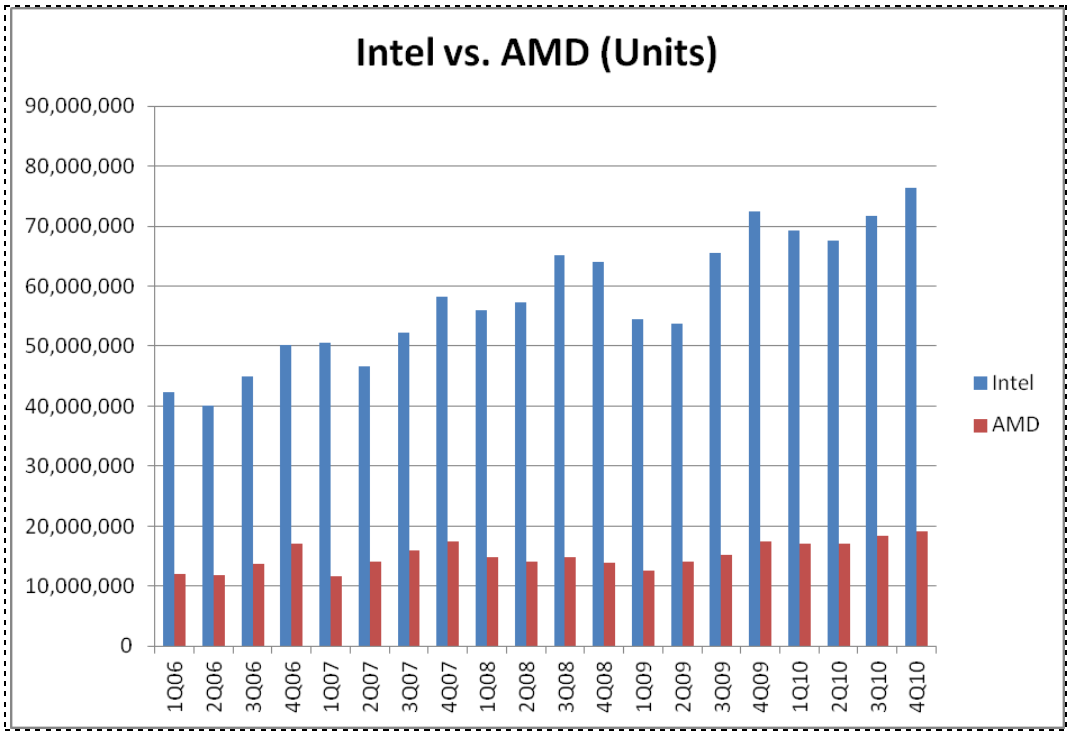


Figure 8. Intel and AMD Sales per quarter

The data obtained was verified based on 2 quarters which were lagging during the global economic crisis in the fiscal 2008-09 calendar. Several other models were created and plotted to analyze the quality of data and is included in the Appendix.

Inference

The result was that the data was obtained to the satisfaction of the business requirements albeit modified in the process to accommodate the gathered data. And the model was found to reach a state of "Success on modified business problem"

4.3. To predict Mergers & Acquisitions among suppliers (Case 3)

Problem Statement

"To predict mergers and acquisitions among suppliers of Intel and its impact on its revenue"

The problem aims at developing a successful predictive model aided by the historical data on all companies that are/were supplying to Intel and acquire data on their success or failure of merging or acquiring one another and their impact concerning Intel's cost of operation and revenues.

The procedure

A look into the factors that could drive mergers or acquisitions, there were 4 major aspects that come to light:

1. Need – Does one see the need for a company to merge or acquire another one? Basically this is estimated by financial standings of the company and their projected 5-year revenue growth.
2. Ability – Do the companies have the clout to stand in the market and lead the next few years? This could be obtained from the spending habit of the company in research and development (R&D) in the past 10 year period.
3. Rewards – Do the companies have a reward to increasing their market standing through the merger or acquisition? For this, one would require the market share information of the companies.
4. Opportunity – Is there a possibility that a company could seek an opportunity to enter a new market place? And for which one would

require data on how has the company handled their investment portfolios.

The process

The course of data phase started off with research on data sources for all four aspects discussed above. However, through the search it was found that

a) The data about financial stature and expenditures for publicly trading companies are extremely difficult to obtain and market research companies seldom publish financial data of companies freely to public
b) The financial data are impossible to acquire if the company is privately held. And since some of the supplier companies are privately held, it's a setback.

Discussion and Outcome

At this stage, there was plenty of discussions between the three units to resolve the inevitable fuzzy state. It was determined that the data that could be acquired does not match the business requirements. On the corollary, the business and algorithm requirements (successful prediction of mergers and acquired) could not be compromised to accommodate the data. Hence it was determined that the process could not be taken forward and was decided to scrap the business problem. One of the major reasons to this outcome could be attributed to the people involved in this case study as the people in the three units were separated geographically, which would have resulted in lack of frequent communication. And the priority of this study was set low owing to focus on more pertinent problem to the company.

4.4. Experimental Results

Here is a summary of all the findings from the three case studies

4.4.1 Case 1

- The data is retrieved completely and was scaled and integrated
- The task was significant but achievable
- The business and modeling people's requirement matched with the effort put in by data engineering.
- The 3 entities (Business unit, Data Unit, Algorithm unit) were in constant communication on every step in the process.

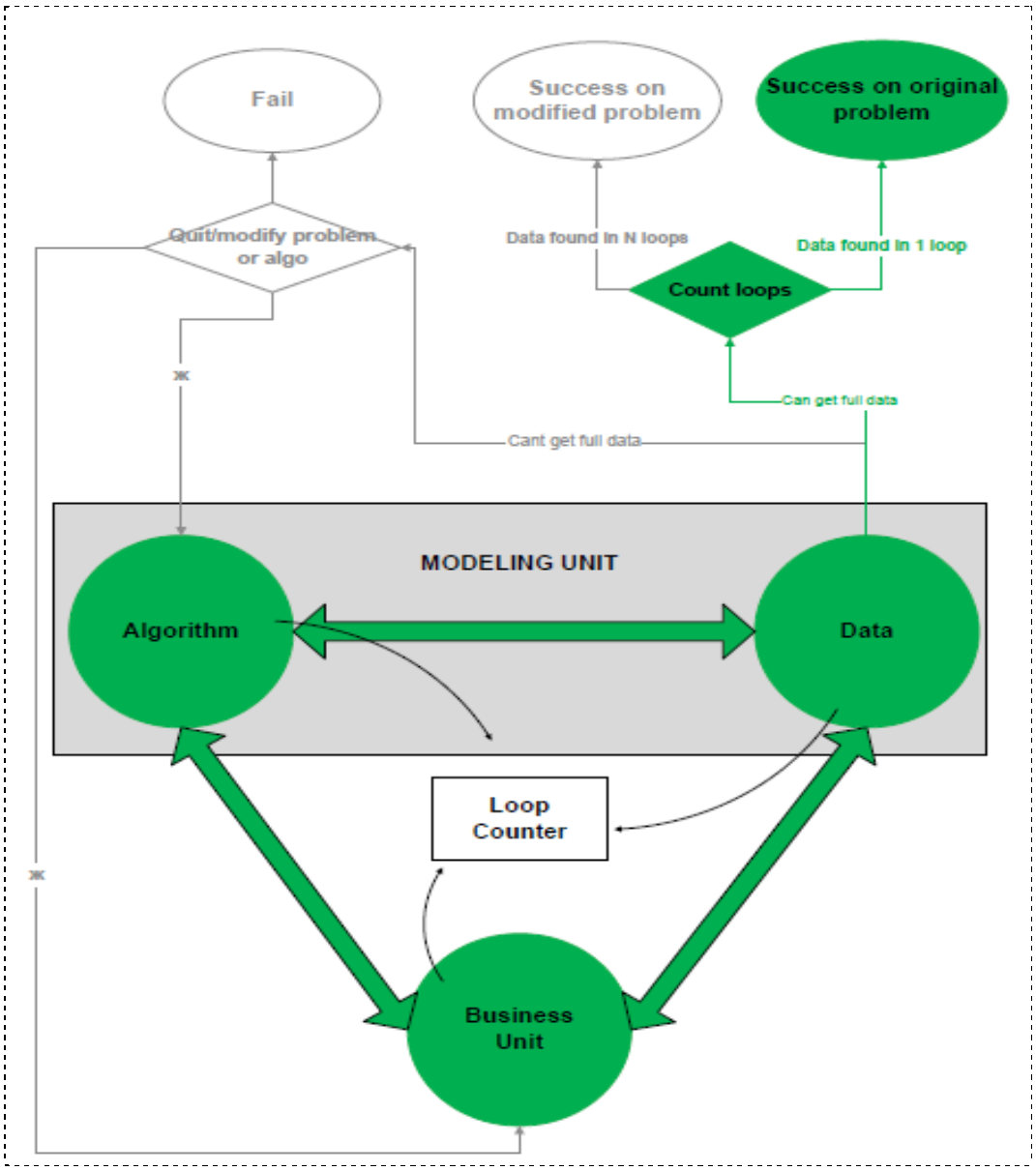


Figure 9. The highlighted flow on "Case 1" – Success on original problem

The green path highlights the process flow between the entities and the end state in this case study.

4.4.2 Case 2

- The data could not be retrieved completely but enough to tweak the business problem.
- The task was significant but achievable only upon changing requirements
- The requirements of the business and modeling people could not be met by data engineer's effort as the data is not available freely.
- The 3 entities (Business unit, Data engineering, Modeling unit) were in constant communication on every step in the process, enabling a change of requirements

NOTE The process loops back to change business requirement and algorithm accordingly, and finally reaches the end state at the end of the subsequent loop.

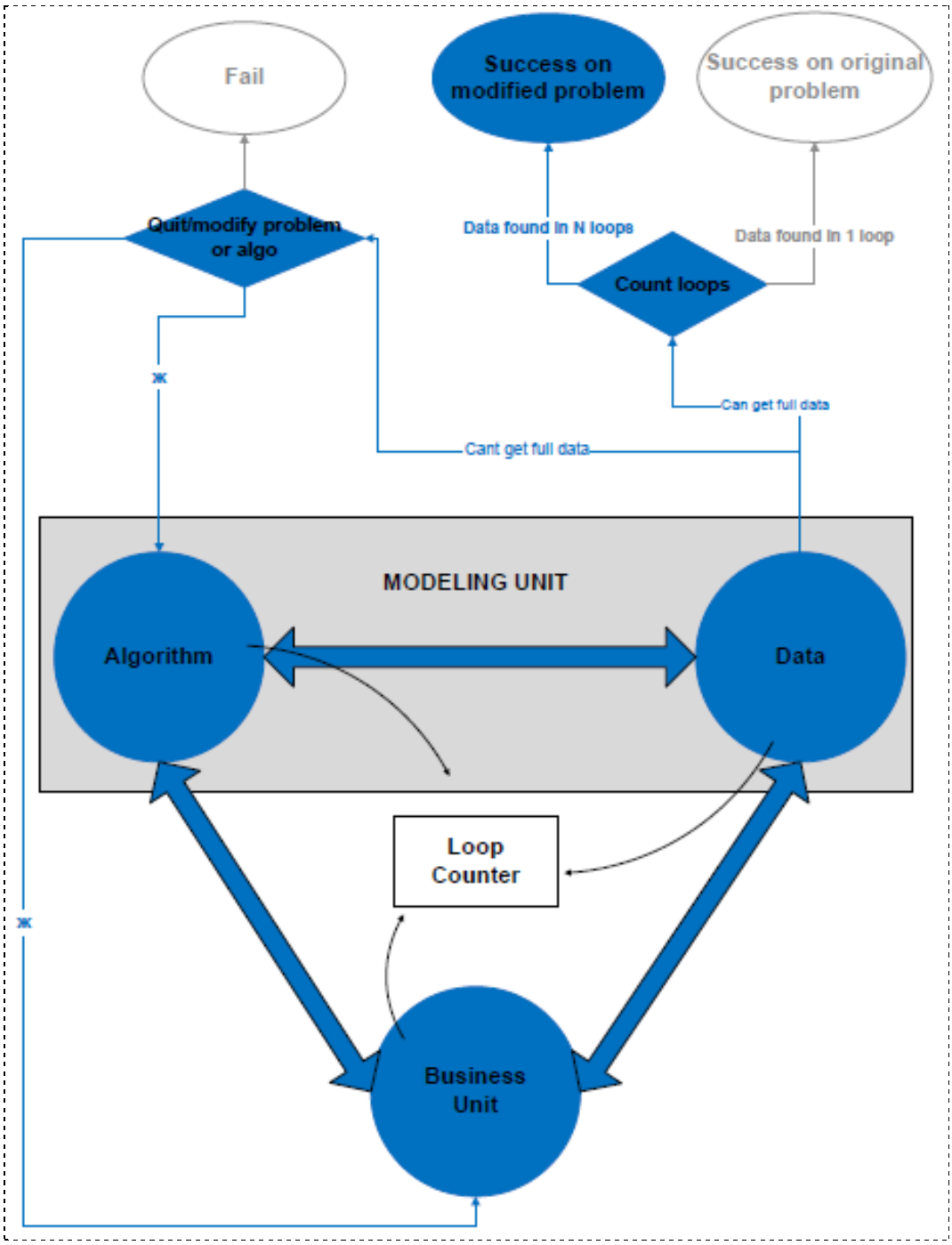


Figure 10. The highlighted flow on "Case 2" – Success on modified problem

The blue path highlights the process flow between the entities and the end state in this case study.

4.4.3 Case 3

- The data could not be retrieved that could be sufficient to make progress
- The task was almost impossible to complete with lack of data being a big setback
- The 3 entities (Business unit, Data engineering, Modeling unit) were lacking frequent communication to go through several loops in the process, and could've had an impact in the final result of the task
- The case was deemed as failed.

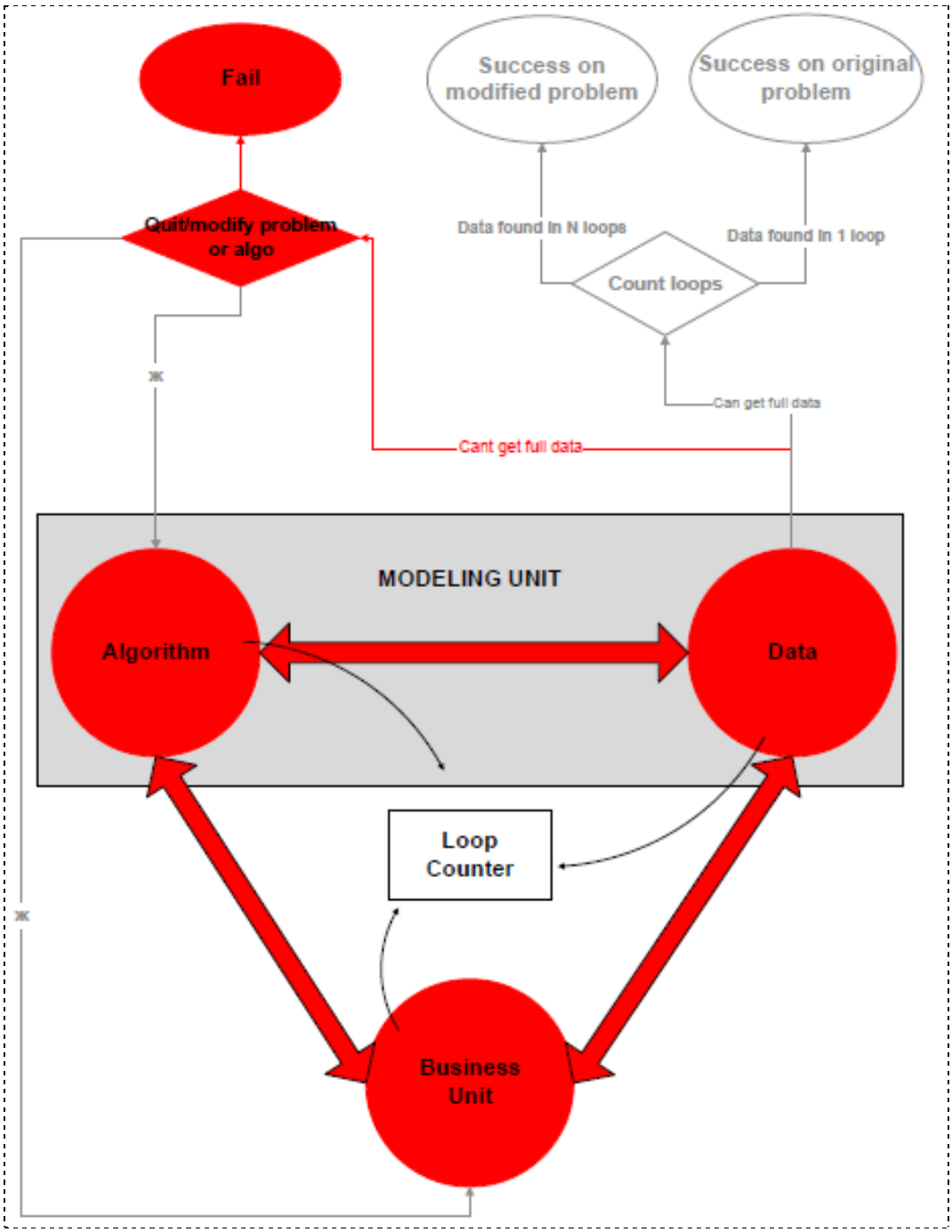


Figure 11. The highlighted flow on "Case 3" – Fail

The red path highlights the process flow between the entities and the end state in this case study.

CHAPTER 5

OBSERVATIONS

1. The current theories only look at a subset of the holistic approach. This mean that though the existing models they just work on a subset of the bigger data problem. When I look at the complete view, I can easily see that the existing model works with just one output state "Success on original business problem".
2. The current theories usually make an assumption of data availability and suitability. They focus on developing a statistical decision modeling based on some available data and only if the data looks insufficient in quantity or features to develop successful models, one looks back into the data preparation phase again. Hence the focus is more on tweaking the model to get the desired results. But realistically one should look back into the data and figure out what data will resolve it. Our approach was to look in for "what data" should be collected that will resolve the issue.
3. As highlighted several times during the experiment descriptions, the communication between different units is vital. The current methods show the transition between each phase, but they overlook the factor of communication, which plays a critical role in reducing turnaround time for several processes in the loop and helps in resolving fuzzy states.
4. One of the biggest blind spot in this whole process is the People involved. The success in case1 and case2 experiments and the failure of case3 could easily be attributed to the people involved as much as it is attributed to the process. There are several attributes to the people

involved that could become factors in the final outcome. One, the priority level of the project (case study) in different units. Often times the people in the business, data and modeling units may belong to different groups or divisions which could mean that not everyone involved would have the same priorities or commitments towards arriving at a solution. Two, the skill set and experience of the people in the three units could be varied, and since communication is a major factor in the result, this could be a factor in determining the direction in these communications, especially in resolving conflicts and the fuzzy states. Third, the geographical separation of the people, which could hinder communication and progress, a major factor that contributed to the demise of case3.

5. The data quantification step discussed in case1, was a significant step in knowing how much time and effort need to be spent on data gathering and how many iterations of data acquisition are required to proceed to the next steps. However, this is not a sampling problem, i.e. it is not a concern as to whether I needed to look at other sources of data to bridge the gap of missing data. This is a different issue by itself, since if I start looking at more than the selected sources of data (in our case - Wikipedia and SPEC), a question would start to arise on the authenticity and accuracy of data obtained from the third source. And the missing data metric would become a third order equation, making it more complex. Hence, the focus was maintained in unraveling more data from selected sources which would contribute towards filling the missing data and increase our confidence at the data collection stage, rather on stepping

back to the data source selection and spend time on selecting, comparing and analyzing more sources for the same data.

CHAPTER 6

CONCLUSION

At the tipping point of usage of the data mining techniques in industries, it is imperative to understand the implications of these techniques in real-time. And through the work that was involved in the thesis, I was able to make a significant leap in our perception of how data mining applies in industries.

This thesis has made noteworthy contributions to the field. Firstly, I was able to generalize the flow of process and control in the existing methods. The new flow gives a better insight into the data mining process and gives better coverage of possible real world scenarios. The proposed flow makes for a better understanding when it's seen as a flow of control between entities and states as opposed to flow between phases in the existing methods. It also provides a systematic approach to handle failure in the process. Secondly, I was able to demonstrate the importance of data availability and usability as the most important criteria in determining the success or failure of the data mining solution to a business problem. Unlike the existing methods, which evaluate the possibility of failure after the modeling phase, I was able to show how the decision is actually dependent on the success or failure of data availability. Apart from these, I also contributed a way of quantifying the data, which gives a way to learn how much data is missing and establish a confidence measure of how much data has been collected.

Moving forward, the work presented here provides scope for further studies and research in the area of data mining application in industries. One of the major areas that could be pondered over is the comprehension of the fuzzy

state in the proposed flow. The fuzzy state controls critical decision making for each solution and stands between looping back/reworking on the issue and deeming it a failed endeavor. More experiments could be conducted to model the fuzzy state resolution. Another possible area would be the study of branching ratios in the control flow and understand how often one reworks on a problem which could pave way for improving the success ratios.

REFERENCES

[1] Fayyad, U. M. et al. 1996. From data mining to knowledge discovery: an overview. In Fayyad, U. M.et al (Eds.), Advances *in knowledge discovery and data mining*. AAAI Press / The MIT Press.

[2] Benoît, G., 2002. Data Mining. Annual Review of Information Science and Technology, Vol. 36, No. 1, pp 265-310.

[3] Brachman, R. J. & Anand, T., 1996. The process of knowledge discovery in databases. In Fayyad, U. M. et al. (Eds.), *Advances in knowledge discovery* and data mining. AAAI Press / The MIT Press.

[4] Chen, M. et al, 1996. Data Mining: An Overview from a Database Perspective. IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 6, pp 866-883.

[5] Simoudis, E., 1996. Reality check for data mining. IEEE Expert, Vol. 11, No. 5, pp 26-33.

[6] Fayyad, U. M., 1996. Data mining and knowledge discovery: making sense out of data. IEEE Expert, Vol. 11 No. 5, pp20-25.

[7] Accessed from http://www.decisionstats.com/wp-content/uploads/2011/10/fayyad1996.png on July 2011.

[8] Dzeroski, S., 2006. Towards a General Framework for Data Mining.. In Dzeroski, S and Struyf, J (Eds.), Knowledge Discovery in Inductive Databases. LNCS 47474. Springer-Verlag.

[9] Han, J. et al, 1996. DMQL: A Data Mining Query Language for Relational Databases. In proceedings of DMKD-96 (SIGMOD-96 Workshop on KDD). Montreal. Canada.

[10] Meo, R. e tal, 1998. An Extension to SQL for Mining Association Rules. Data Mining and Knowledge Discovery Vol. 2, pp 195-224. Kluwer Academic Publishers.

[11] Imielinski, T.; Virmani, A., 1999. MSQL: A Query Language for Database Mining. Data Mining and Knowledge Discovery Vol. 3, pp 373-408. Kluwer Academic Publishers.

[12] Sarawagi, S. et al, 2000. Integrating Association Rule Mining with Relational Database Systems: Alternatives and Implications. Data Mining and Knowledge Discovery, Vol. 4, pp 89–125.

[13] Botta, Marco, et al, 2004. Query Languages Supporting Descriptive Rule Mining: A Comparative Study. Database Support for Data Mining Applications. LNAI 2682, pp 24-51.

[14] SAS Enterprise Miner – SEMMA. SAS Institute.

[15] Accessed from http://www.sas.com/technologies/analytics/datamining/miner/semma.html, on July 2011

[16] Espinosa, Roberto, 2011. Visual representation of SEMMA.

[17] Accessed from http://www.decisionstats.com/wp-content/uploads/2011/10/metodo-semma.jpg on Oct 2011.

[18] Santos, M & Azevedo, C (2005). Data Mining – Descoberta de Conhecimento em Bases de Dados. FCA Publisher.

[19] Chapman, P. et al, 2000. CRISP-DM 1.0 - Step-by-step data mining guide.

[20] Accessed from http://www.crisp-dm.org/CRISPWP-0800.pdf on July 2011

[21] CRISP-DM for InfoTech Marketing.

[22] Accessed from http://www.infotechmarketing.net/data_mining.htm on July 14, 2011.

[23] Barrow, John D.,2009 - 100 Essential Things You Din't Know You Din't Know.

APPENDIX A

CODE SNIPPETS AND EXAMPLES

1. Perl regular expression match patterns for retrieving wiki dataset

```
use strict;
use warnings;
use LWP::Simple;

my $url =
"http://en.wikipedia.org/wiki/List_of_Intel_Xeon_microproce
ssors/";

my $html1;
my @html;
my $processor;
my $frequency;
my $gpufreq;
my $l2cache;
my $l3cache;
my $reldate;
my $relprice;
my $match;
my $match1;
my $match2;
my $match3;
my $match4;
my $match5;
my $match6;
my $match7;
my $match8;
my $match9;
my $match10;
my $match11;
my $match12;
my $match13;
my $l2cacheval;
my $l3cacheval;
my $procval;
my $freqval;
my $gpufreqval;
my $reldateval;
my $relpriceval;

open (FILE, ">output.txt") or die $!;
print FILE "Processor\tFrequency\tGPU
Frequency\tL2Cache\tL3Cache\tRel Date\tRel Price\n";

{
    local $/ = undef;
   $html1 = <HTML>;
}
```

```
$processor = "<td>(?<proc>[^<]*?)<\/td>";
$frequency = "<td>(?<freq>.*(MHz|GHz))<\/td>";
$gpufreq = "<td>(?<gpufreq>.*(MHz|GHz))<\/td>";
$l2cache = "<td>(?<l2cache>.*(<a.*)?(Ki?B|MB)).*?<\/td>";
$l3cache = "<td>(?<l3cache>.*(<a.*)?(Ki?B|MB)).*?<\/td>";
$reldate =
"<td>(?<reldate>(Jan(uary)?|Feb(ruary)?|Mar(ch)?|Apr(il)?|M
ay?|June?|July?|Aug(ust)?|Sep(tember)?|Oct(ober)?|Nov(ember
)?|Dec(ember)?)\\s*[\\d,]*\\s\\d{4})<\/td>";
$relprice = "<td>(?<relprice>\\$\\d+)<\/td>";

#13 Different Match options to retrieve the data
$match = $processor . "(\n|.)*" . $frequency  . "(\n|.)*" .
$gpufreq  . "(\n|.)*" .  $l2cache . "(\n|.)*" . $l3cache .
"(\n|.)*" . $reldate . "(\n|.)*" . $relprice;
$match1 = $processor . "(\n|.)*" . $frequency  . "(\n|.)*"
. $gpufreq  . "(\n|.)*" .  $l2cache . "(\n|.)*" . $l3cache
. "(\n|.)*" . $reldate;
$match2 = $processor . "(\n|.)*" . $frequency  . "(\n|.)*"
. $gpufreq  . "(\n|.)*" .  $l2cache . "(\n|.)*" . $reldate
. "(\n|.)*" . $relprice;
$match3 = $processor . "(\n|.)*" . $frequency  . "(\n|.)*"
. $gpufreq  . "(\n|.)*" .  $l2cache . "(\n|.)*" . $reldate;
$match4 = $processor . "(\n|.)*" . $frequency  . "(\n|.)*"
. $gpufreq  . "(\n|.)*" .  $reldate . "(\n|.)*" .
$relprice;
$match5 = $processor . "(\n|.)*" . $frequency  . "(\n|.)*"
. $gpufreq  . "(\n|.)*" .  $l2cache . "(\n|.)*" . $l3cache;
$match6 = $processor . "(\n|.)*" . $frequency  . "(\n|.)*"
. $gpufreq  . "(\n|.)*" .  $l2cache;
$match7 = $processor . "(\n|.)*" . $frequency  . "(\n|.)*"
.  $l2cache . "(\n|.)*" . $l3cache . "(\n|.)*" . $reldate .
"(\n|.)*" . $relprice;
$match8 = $processor . "(\n|.)*" . $frequency  . "(\n|.)*"
.  $l2cache . "(\n|.)*" . $l3cache . "(\n|.)*" . $reldate;
$match9 = $processor . "(\n|.)*" . $frequency  . "(\n|.)*"
.  $l2cache . "(\n|.)*" . $reldate . "(\n|.)*" . $relprice;
$match10 = $processor . "(\n|.)*" . $frequency  . "(\n|.)*"
.  $l2cache . "(\n|.)*" . $reldate;
$match11 = $processor . "(\n|.)*" . $frequency  . "(\n|.)*"
.  $reldate . "(\n|.)*" . $relprice;
$match12 = $processor . "(\n|.)*" . $frequency  . "(\n|.)*"
.  $l2cache . "(\n|.)*" . $l3cache;
$match13 = $processor . "(\n|.)*" . $frequency  . "(\n|.)*"
.  $l2cache;

@html = ($html1 =~ /<tr>\n(<td(.|\n)*?)\n<\/tr>/mg);

my $records = @html;
```

```
print "Length = $records\n";

for (my $i=0; $i<$records; $i = $i+2) {
  if($html[$i] =~ /$match/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $gpufreqval = $+{gpufreq};
    $reldateval = $+{reldate};
    $relpriceval = $+{relprice};
    $l2cacheval = $+{l2cache};
    $l3cacheval = $+{l3cache};
    $l2cacheval =~ s/<a href="\/wiki.*">//g;
    $l3cacheval =~ s/<a href="\/wiki.*">//g;
    print FILE
"$procval\t$freqval\t$gpufreqval\t$l2cacheval\t$l3cacheval\
t$reldateval\t$relpriceval\n";
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=$gpufreqval\tL2Cache=$l2cacheval\tL3Cache=$l3cacheval\
tRel_Date=$reldateval\tRel_Price=$relpriceval\n";
  }
  elsif($html[$i] =~ /$match1/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $gpufreqval = $+{gpufreq};
    $reldateval = $+{reldate};
    $l2cacheval = $+{l2cache};
    $l3cacheval = $+{l3cache};
    $l2cacheval =~ s/<a href="\/wiki.*">//g;
    $l3cacheval =~ s/<a href="\/wiki.*">//g;
    print FILE
"$procval\t$freqval\t$gpufreqval\t$l2cacheval\t$l3cacheval\
t$reldateval\tN/A\n";
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=$gpufreqval\tL2Cache=$l2cacheval\tL3Cache=$l3cacheval\
tRel_Date=$reldateval\tRel_Price=N/A\n";
  }
  elsif($html[$i] =~ /$match2/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $gpufreqval = $+{gpufreq};
    $reldateval = $+{reldate};
    $relpriceval = $+{relprice};
    $l2cacheval = $+{l2cache};
    $l2cacheval =~ s/<a href="\/wiki.*">//g;
    print FILE
"$procval\t$freqval\t$gpufreqval\t$l2cacheval\tN/A\t$reldat
eval\t$relpriceval\n";
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=$gpufreqval\tL2Cache=$l2cacheval\tL3Cache=N/A\tRel_Dat
e=$reldateval\tRel_Price=$relpriceval\n";
```

```
  }
  elsif($html[$i] =~ /$match3/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $gpufreqval = $+{gpufreq};
    $reldateval = $+{reldate};
    $l2cacheval = $+{l2cache};
    $l2cacheval =~ s/<a href="\/wiki.*">//g;
    print FILE
"$procval\t$freqval\t$gpufreqval\t$l2cacheval\tN/A\t$reldat
eval\tN/A\n";
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=$gpufreqval\tL2Cache=$l2cacheval\tL3Cache=N/A\tRel_Dat
e=$reldateval\tRel_Price=N/A\n";
  }
  elsif($html[$i] =~ /$match4/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $gpufreqval = $+{gpufreq};
    $reldateval = $+{reldate};
    print FILE
"$procval\t$freqval\t$gpufreqval\tN/A\tN/A\t$reldateval\tN/
A\n";
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=$gpufreqval\tL2Cache=N/A\tL3Cache=N/A\tRel_Date=$relda
teval\tRel_Price=N/A\n";
  }
  elsif($html[$i] =~ /$match5/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $gpufreqval = $+{gpufreq};
    $l2cacheval = $+{l2cache};
    $l3cacheval = $+{l3cache};
    $l2cacheval =~ s/<a href="\/wiki.*">//g;
    $l3cacheval =~ s/<a href="\/wiki.*">//g;
    print FILE
"$procval\t$freqval\t$gpufreqval\t$l2cacheval\t$l3cacheval\
tN/A\tN/A\n";
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=$gpufreqval\tL2Cache=$l2cacheval\tL3Cache=$l3cacheval\
tRel_Date=N/A\tRel_Price=N/A\n";
  }
  elsif($html[$i] =~ /$match6/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $gpufreqval = $+{gpufreq};
    $l2cacheval = $+{l2cache};
    $l2cacheval =~ s/<a href="\/wiki.*">//g;
```

```
    print FILE
"$procval\t$freqval\t$gpufreqval\t$l2cacheval\tN/A\tN/A\tN/
A\n";
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=$gpufreqval\tL2Cache=$l2cacheval\tL3Cache=N/A\tRel_Dat
e=N/A\tRel_Price=N/A\n";
  }
  elsif($html[$i] =~ /$match7/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $reldateval = $+{reldate};
    $relpriceval = $+{relprice};
    $l2cacheval = $+{l2cache};
    $l3cacheval = $+{l3cache};
    $l2cacheval =~ s/<a href="\/wiki.*">//g;
    $l3cacheval =~ s/<a href="\/wiki.*">//g;
    print FILE
"$procval\t$freqval\tN/A\t$l2cacheval\t$l3cacheval\t$reldat
eval\t$relpriceval\n";
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=N/A\tL2Cache=$l2cacheval\tL3Cache=$l3cacheval\tRel_Dat
e=$reldateval\tRel_Price=$relpriceval\n";
  }
  elsif($html[$i] =~ /$match8/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $reldateval = $+{reldate};
    $l2cacheval = $+{l2cache};
    $l3cacheval = $+{l3cache};
    $l2cacheval =~ s/<a href="\/wiki.*">//g;
    $l3cacheval =~ s/<a href="\/wiki.*">//g;
    print FILE
"$procval\t$freqval\tN/A\t$l2cacheval\t$l3cacheval\t$reldat
eval\tN/A\n";
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=N/A\tL2Cache=$l2cacheval\tL3Cache=$l3cacheval\tRel_Dat
e=$reldateval\tRel_Price=N/A\n";
  }
  elsif($html[$i] =~ /$match9/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $reldateval = $+{reldate};
    $relpriceval = $+{relprice};
    $l2cacheval = $+{l2cache};
    $l2cacheval =~ s/<a href="\/wiki.*">//g;
    print FILE
"$procval\t$freqval\tN/A\t$l2cacheval\tN/A\t$reldateval\t$r
elpriceval\n";
```

```
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=N/A\tL2Cache=$l2cacheval\tL3Cache=N/A\tRel_Date=$relda
teval\tRel_Price=$relpriceval\n";
  }
  elsif($html[$i] =~ /$match10/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $reldateval = $+{reldate};
    $l2cacheval = $+{l2cache};
    $l2cacheval =~ s/<a href="\/wiki.*">//g;
    print FILE
"$procval\t$freqval\tN/A\t$l2cacheval\tN/A\t$reldateval\tN/
A\n";
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=N/A\tL2Cache=$l2cacheval\tL3Cache=N/A\tRel_Date=$relda
teval\tRel_Price=N/A\n";
  }
  elsif($html[$i] =~ /$match11/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $reldateval = $+{reldate};
    print FILE
"$procval\t$freqval\tN/A\tN/A\tN/A\t$reldateval\tN/A\n";
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=N/A\tL2Cache=N/A\tL3Cache=N/A\tRel_Date=$reldateval\tR
el_Price=N/A\n";
  }
  elsif($html[$i] =~ /$match12/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $l2cacheval = $+{l2cache};
    $l3cacheval = $+{l3cache};
    $l2cacheval =~ s/<a href="\/wiki.*">//g;
    $l3cacheval =~ s/<a href="\/wiki.*">//g;
    print FILE
"$procval\t$freqval\tN/A\t$l2cacheval\t$l3cacheval\tN/A\tN/
A\n";
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=N/A\tL2Cache=$l2cacheval\tL3Cache=$l3cacheval\tRel_Dat
e=N/A\tRel_Price=N/A\n";
  }
  elsif($html[$i] =~ /$match13/mg) {
    $procval =  $+{proc};
    $freqval = $+{freq};
    $l2cacheval = $+{l2cache};
    $l2cacheval =~ s/<a href="\/wiki.*">//g;
    print FILE
"$procval\t$freqval\tN/A\t$l2cacheval\tN/A\tN/A\tN/A\n";
```

```
    print "Processor=$procval\tFrequency=$freqval\tGPU
Freq=N/A\tL2Cache=$l2cacheval\tL3Cache=N/A\tRel_Date=N/A\tR
el_Price=N/A\n";
  }
  else {
    print FILE "No information retrieved of record number
$i \n\n $html[$i]\n\n";
    print "No information retrieved of record number $i
\n\n $html[$i]\n\n";
  }

}
```

2. SAS code to manipulate and qualify data

```
libname intel 'C:\SAS\Data\Intel';

options helpbrowser=sas;

%let dataLoc=C:\SAS\Data;
proc import dbms=xls out=intel.specRate
            datafile="&dataLoc\Latest_INTEL_SPEC2006.xls"
replace;
   sheet='Sheet1';
run;
/*************************************************************
*********************/
/* FP_INT_RATES*/
/*************************************************************
*********************/
data intel.specRate_intrate;
  set intel.specRate;
  if compress(Benchmark) = 'CINT2006rate';
run;
data intel.specRate_fprate;
  set intel.specRate;
  if compress(Benchmark) = 'CFP2006rate';
run;
/*************************************************************
*********************/
/* INT_RATES_MP_DP_UP*/
/*************************************************************
*********************/
data intel.specRate_intrate_DP;
  set intel.specRate_intrate;
  if UPCASE(compress(wiki_MarketSegment)) = 'DP';
run;
data intel.specRate_intrate_MP;
  set intel.specRate_intrate;
  if UPCASE(compress(wiki_MarketSegment)) = 'MP';
run;
data intel.specRate_intrate_UP;
  set intel.specRate_intrate;
  if UPCASE(compress(wiki_MarketSegment)) = 'DESKTOP' or
UPCASE(compress(wiki_MarketSegment)) = 'MOBILE';
run;
/*************************************************************
*******************/
/* INT_RATES_DP*/
/*************************************************************
*******************/
data intel.specRate_intrate_DP;
```

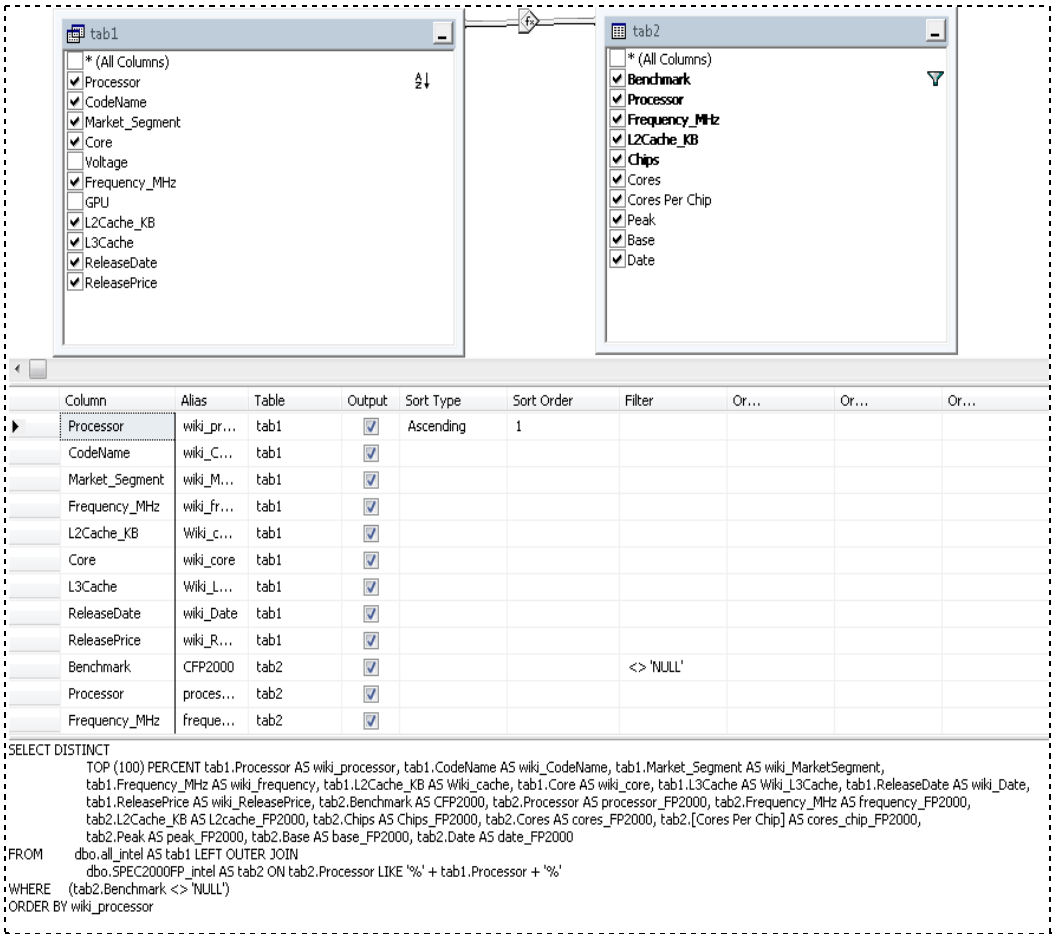
```
  set intel.specRate_intrate_DP;
  xaxis = 2 * Cores_Per_Chip * wiki_frequency;
run;

%ODSOn(path=c:\SAS\Output, name=specrate, style=listing,
ODSFormat=PDF);
    goptions xmax=30 inches ymax=10 inches hsize=7 inches
vsize=6 inches device=png ftext='Calibri'
ftitle='Calibri/bold' htitle=3.5 pct htext=2.5 pct;

    axis1 label=('Cores_Frequency Combination')
value=(angle=90 rotate=0);  /* only use the group axis for
value/bar text */
    axis2 label=('Scaled Peak') minor=(number=1)
offset=(0,0);
    axis3 value=(angle=90 rotate=0);

    legend1 label=none position=(top right inside)
cframe=white mode=protect
     shape=bar(3,3) cborder=black across=1;

    /* pattern v=solid color=red; */
    pattern1 v=solid color=cxbd0026;  /* reddish color */
    pattern2 v=solid color=cx43a2ca;  /* this is the hex
rgb color for mild blue */

    title "Intel DP Processor INT_Rate Benchmark.";
    proc sort data=intel.specRate_intrate_DP;
      by Modified_Chips xaxis;

proc means data=intel.specRate_intrate_DP noprint;
  by Modified_Chips xaxis;
  var Scaled_Peak;
  output out=intel.specrate_intrate_DP_overall
max=Scaled_Peak;
run;
%resetSymbols(i=none);

proc gplot data=intel.specrate_intrate_DP_overall;
  plot Scaled_Peak*xaxis = Modified_Chips / haxis=axis1
vaxis=axis2;
  run;
quit;
proc gplot data=intel.specrate_intrate_DP_overall;
  by Modified_Chips;
  plot Scaled_Peak*xaxis = Modified_Chips / haxis=axis1
vaxis=axis2;
  run;
quit;
%ODSOff;
```

```
/*************************************************************
******************/
/* INT_RATES_MP*/
/*************************************************************
******************/
data intel.specRate_intrate_MP;
  set intel.specRate_intrate_MP;
  xaxis = 4 * Cores_Per_Chip * wiki_frequency;
run;

%ODSOn(path=c:\SAS\Output, name=specrate, style=listing,
ODSFormat=PDF);
    goptions xmax=30 inches ymax=10 inches hsize=7 inches
vsize=6 inches device=png ftext='Calibri'
ftitle='Calibri/bold' htitle=3.5 pct htext=2.5 pct;

    axis1 label=('Cores_Frequency Combination')
value=(angle=90 rotate=0);  /* only use the group axis for
value/bar text */
    axis2 label=('Scaled Peak') minor=(number=1)
offset=(0,0);
    axis3 value=(angle=90 rotate=0);

    legend1 label=none position=(top right inside)
cframe=white mode=protect
     shape=bar(3,3) cborder=black across=1;

    /* pattern v=solid color=red; */
    pattern1 v=solid color=cxbd0026;  /* reddish color */
    pattern2 v=solid color=cx43a2ca;  /* this is the hex
rgb color for mild blue */

    title "Intel MP Processor INT_Rate Benchmark.";
    proc sort data=intel.specRate_intrate_MP;
      by Modified_Chips xaxis;

proc means data=intel.specRate_intrate_MP noprint;
  by Modified_Chips xaxis;
  var Scaled_Peak;
  output out=intel.specrate_intrate_MP_overall
max=Scaled_Peak;
run;
%resetSymbols(i=none);

proc gplot data=intel.specrate_intrate_MP_overall;
  plot Scaled_Peak*xaxis = Modified_Chips / haxis=axis1
vaxis=axis2;
  run;
quit;
```

```
proc gplot data=intel.specrate_intrate_MP_overall;
  by Modified_Chips;
  plot Scaled_Peak*xaxis = Modified_Chips / haxis=axis1
vaxis=axis2;
  run;
quit;
%ODSOff;
/*************************************************************
******************/
/* INT_RATES_UP*/
/*************************************************************
******************/
data intel.specRate_intrate_UP;
  set intel.specRate_intrate_UP;
  xaxis = 1 * Cores_Per_Chip * wiki_frequency;
run;

%ODSOn(path=c:\SAS\Output, name=specrate, style=listing,
ODSFormat=PDF);
    goptions xmax=30 inches ymax=10 inches hsize=7 inches
vsize=6 inches device=png ftext='Calibri'
ftitle='Calibri/bold' htitle=3.5 pct htext=2.5 pct;

    axis1 label=('Cores_Frequency Combination')
value=(angle=90 rotate=0);  /* only use the group axis for
value/bar text */
    axis2 label=('Scaled Peak') minor=(number=1)
offset=(0,0);
    axis3 value=(angle=90 rotate=0);

    legend1 label=none position=(top right inside)
cframe=white mode=protect
     shape=bar(3,3) cborder=black across=1;

    /* pattern v=solid color=red; */
    pattern1 v=solid color=cxbd0026;  /* reddish color */
    pattern2 v=solid color=cx43a2ca;  /* this is the hex
rgb color for mild blue */

    title "Intel UP Processor INT_Rate Benchmark.";
    proc sort data=intel.specRate_intrate_UP;
      by Modified_Chips xaxis;

proc means data=intel.specRate_intrate_UP noprint;
  by Modified_Chips xaxis;
  var Scaled_Peak;
  output out=intel.specrate_intrate_UP_overall
max=Scaled_Peak;
run;
%resetSymbols(i=none);
```

```
proc gplot data=intel.specrate_intrate_UP_overall;
  plot Scaled_Peak*xaxis = Modified_Chips / haxis=axis1
vaxis=axis2;
  run;
quit;
proc gplot data=intel.specrate_intrate_UP_overall;
  by Modified_Chips;
  plot Scaled_Peak*xaxis = Modified_Chips / haxis=axis1
vaxis=axis2;
  run;
quit;
%ODSOff;
```

3. SQL view created to merge Intel data from Wikipedia and SPEC



Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...	Or...
Processor	wiki_pr...	tab1	☑	Ascending	1				
CodeName	wiki_C...	tab1	☑						
Market_Segment	wiki_M...	tab1	☑						
Frequency_MHz	wiki_fr...	tab1	☑						
L2Cache_KB	Wiki_c...	tab1	☑						
Core	wiki_core	tab1	☑						
L3Cache	Wiki_L...	tab1	☑						
ReleaseDate	wiki_Date	tab1	☑						
ReleasePrice	wiki_R...	tab1	☑						
Benchmark	CFP2000	tab2	☑			<> 'NULL'			
Processor	proces...	tab2	☑						
Frequency_MHz	freque...	tab2	☑						

```
SELECT DISTINCT
                TOP (100) PERCENT tab1.Processor AS wiki_processor, tab1.CodeName AS wiki_CodeName, tab1.Market_Segment AS wiki_MarketSegment,
                tab1.Frequency_MHz AS wiki_frequency, tab1.L2Cache_KB AS Wiki_cache, tab1.Core AS wiki_core, tab1.L3Cache AS Wiki_L3Cache, tab1.ReleaseDate AS wiki_Date,
                tab1.ReleasePrice AS wiki_ReleasePrice, tab2.Benchmark AS CFP2000, tab2.Processor AS processor_FP2000, tab2.Frequency_MHz AS frequency_FP2000,
                tab2.L2Cache_KB AS L2cache_FP2000, tab2.Chips AS Chips_FP2000, tab2.Cores AS cores_FP2000, tab2.[Cores Per Chip] AS cores_chip_FP2000,
                tab2.Peak AS peak_FP2000, tab2.Base AS base_FP2000, tab2.Date AS date_FP2000
FROM        dbo.all_intel AS tab1 LEFT OUTER JOIN
                dbo.SPEC2000FP_intel AS tab2 ON tab2.Processor LIKE '%' + tab1.Processor + '%'
WHERE     (tab2.Benchmark <> 'NULL')
ORDER BY wiki_processor
```

4. SQL query for data quantification (AMD)

4.1. Finding the Minimum test date from the Spec2006 dataset

```
SELECT MIN([Date])
FROM [Test_amd_Sales].[dbo].[SPEC2006FP_amd]
union
SELECT MIN([Date])
FROM [Test_amd_Sales].[dbo].[SPEC2006FPRATE_amd]
union
SELECT MIN([Date])
FROM [Test_amd_Sales].[dbo].[SPEC2006INT_amd]
union
SELECT MIN([Date])
FROM [Test_amd_Sales].[dbo].[SPEC2006INTRATE_amd]
```

Results | Messages

	(No column name)
1	2006-04-01

4.2. Finding the count of data from the Wiki AMD dataset that have releases greater than the above found date

```
SELECT count(distinct Processor)
FROM [Test_AMD_Sales].[dbo].[all_AMD_Processors]
where  [ReleaseDate] >= '2006-04-01'
```

Results | Messages

	(No column name)
1	407

4.3. Total count of data retrieved from SPEC performance data.

```
select count(*) from
(
(SELECT distinct [Processor]
FROM [Test_amd_Sales].[dbo].[SPEC2006FP_amd])
union
(SELECT distinct [Processor] from
[Test_amd_Sales].[dbo].[SPEC2006FPRATE_amd])
union
(SELECT distinct [Processor] from
[Test_amd_Sales].[dbo].[SPEC2006INT_amd])
union
(SELECT distinct [Processor] from
[Test_amd_Sales].[dbo].[SPEC2006INTRATE_amd])) as a
```

Results | Messages

	(No column name)
1	140

4.4. Count of dataset after merge

```
select count(*) from (
(SELECT distinct [wiki_processor]
FROM [Test_amd_Sales].[dbo].[amd_CFP2006])
union
(SELECT distinct [wiki_processor] from
[Test_amd_Sales].[dbo].[amd_CFP2006rate])
union
(SELECT distinct [wiki_processor] from
[Test_amd_Sales].[dbo].[amd_CINT2006])
union
(SELECT distinct [wiki_processor] from
[Test_amd_Sales].[dbo].[amd_CINT2006rate])) as a
```

Results | Messages

	(No column name)
1	108

4.5. Data that was missing from the collected wiki but found in the SPEC Performance data.

```
SELECT distinct [Processor]
  FROM [Test_AMD_Sales].[dbo].[proc_not_in_wiki_in_spec]
```

Results | Messages

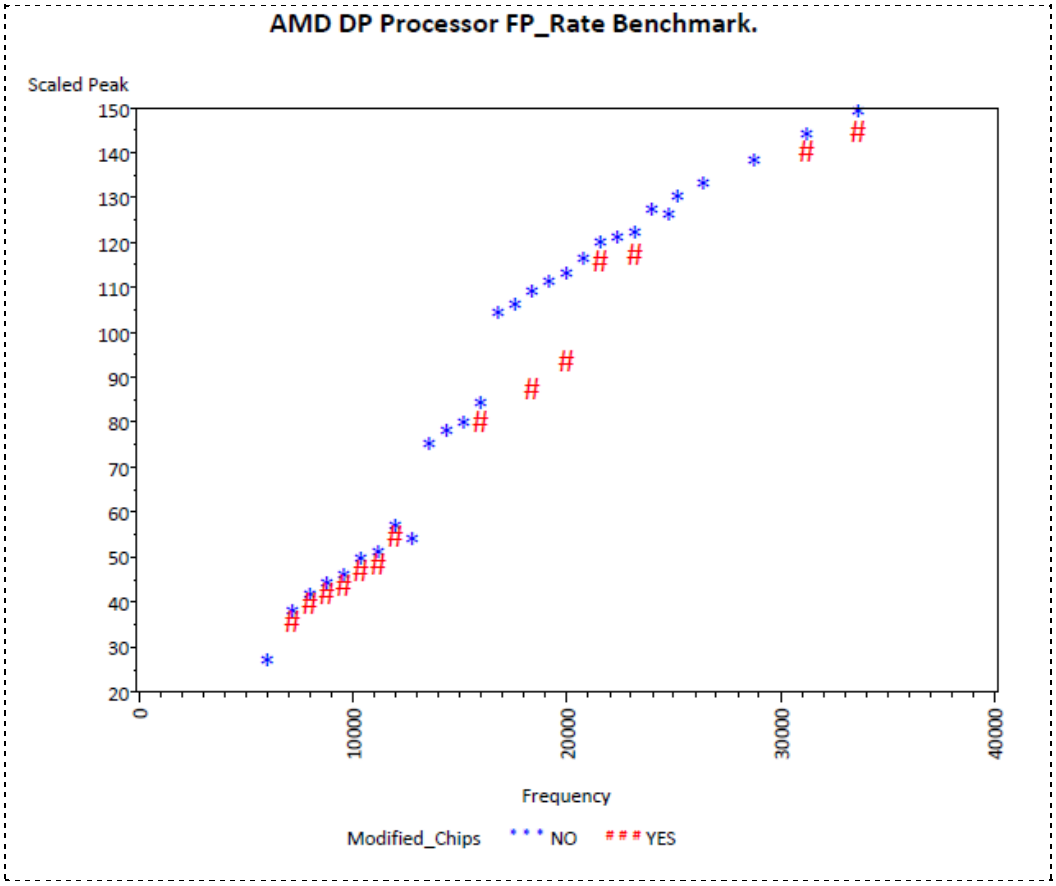
	Processor
1	AMD Opteron 4164 EE
2	AMD Phenom II X4 940
3	AMD Phenom II X3 720
4	AMD Opteron 4170 HE
5	AMD Athlon X2 6000+
6	AMD Opteron 6168
7	AMD Opteron 6134
8	AMD Opteron 6164 HE
9	AMD Opteron 6180 SE
10	AMD Opteron 4174 HE
11	AMD Athlon X2 7750
12	AMD Opteron 6136
13	AMD Opteron 6132 HE
14	AMD Opteron 4180
15	AMD Phenom II X2 555
16	AMD Phenom X3 8750
17	AMD Phenom II X2 570
18	AMD Opteron 6174
19	AMD Opteron 6124 HE
20	AMD Opteron 6176
21	AMD Athlon 64 2100+
22	AMD Opteron 6140
23	AMD Phenom II X2 560
24	AMD Opteron 6172
25	AMD Opteron 6128 HE
26	AMD Athlon 64 X2 3000+
27	AMD Opteron 4176 HE
28	AMD Opteron 6166 HE
29	AMD Opteron 6128
30	AMD Athlon X2 6400+
31	AMD Opteron 6176 SE
32	AMD Opteron 4184

APPENDIX B

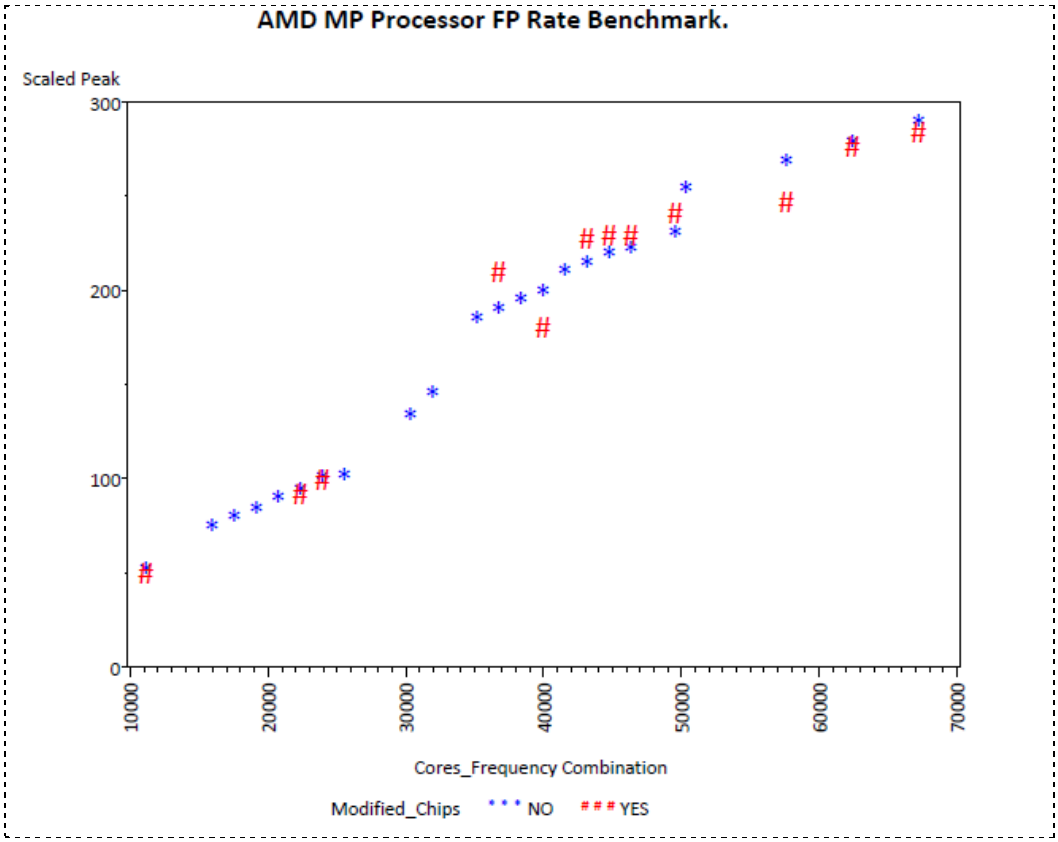
DATA QUALIFICATION PLOTS

1. AMD DP/MP/UP processors performance for FP_RATE benchmarks

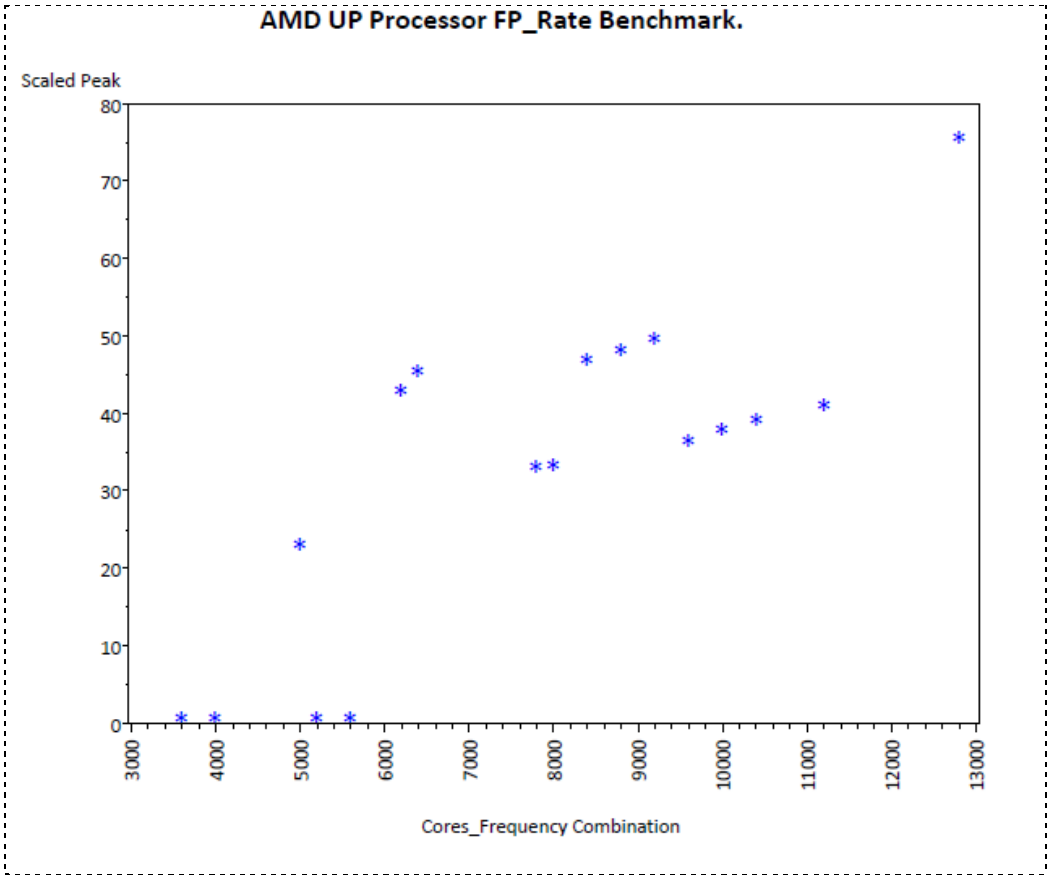
1.1. AMD DP processors performance for FP_RATE benchmarks



1.2. AMD MP processors performance for FP_RATE benchmarks

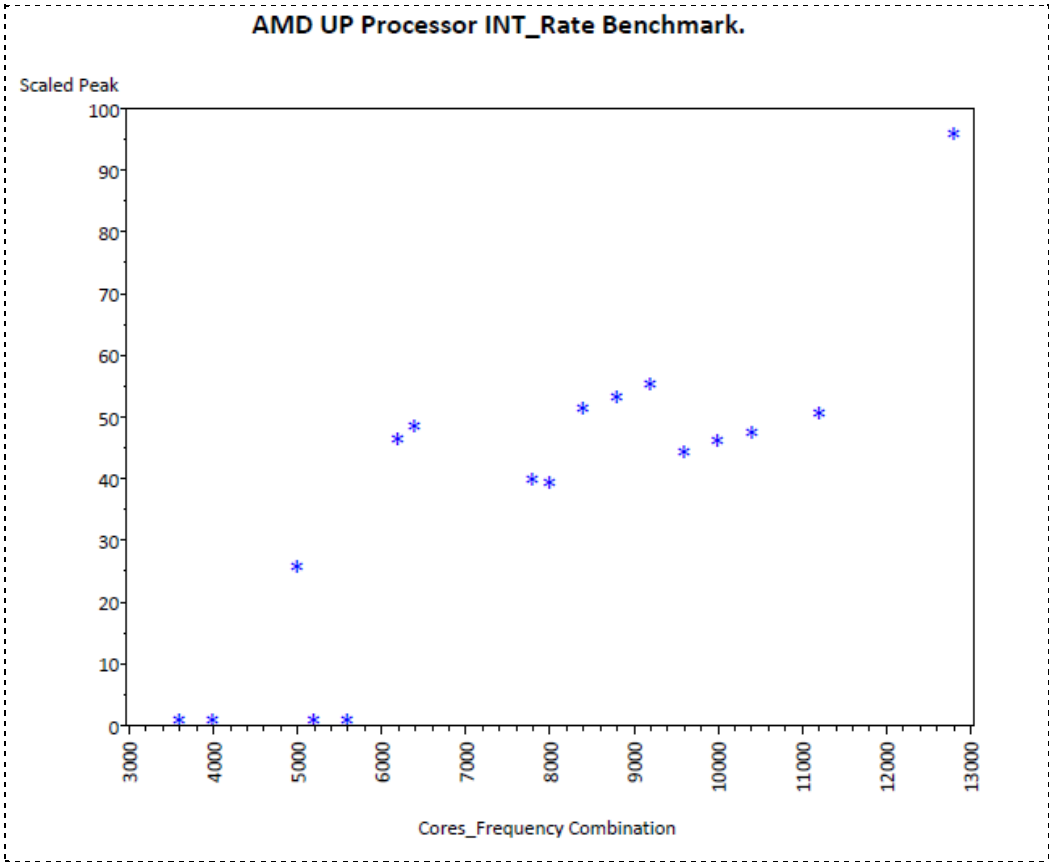


1.3. AMD UP processors performance for FP_RATE benchmarks

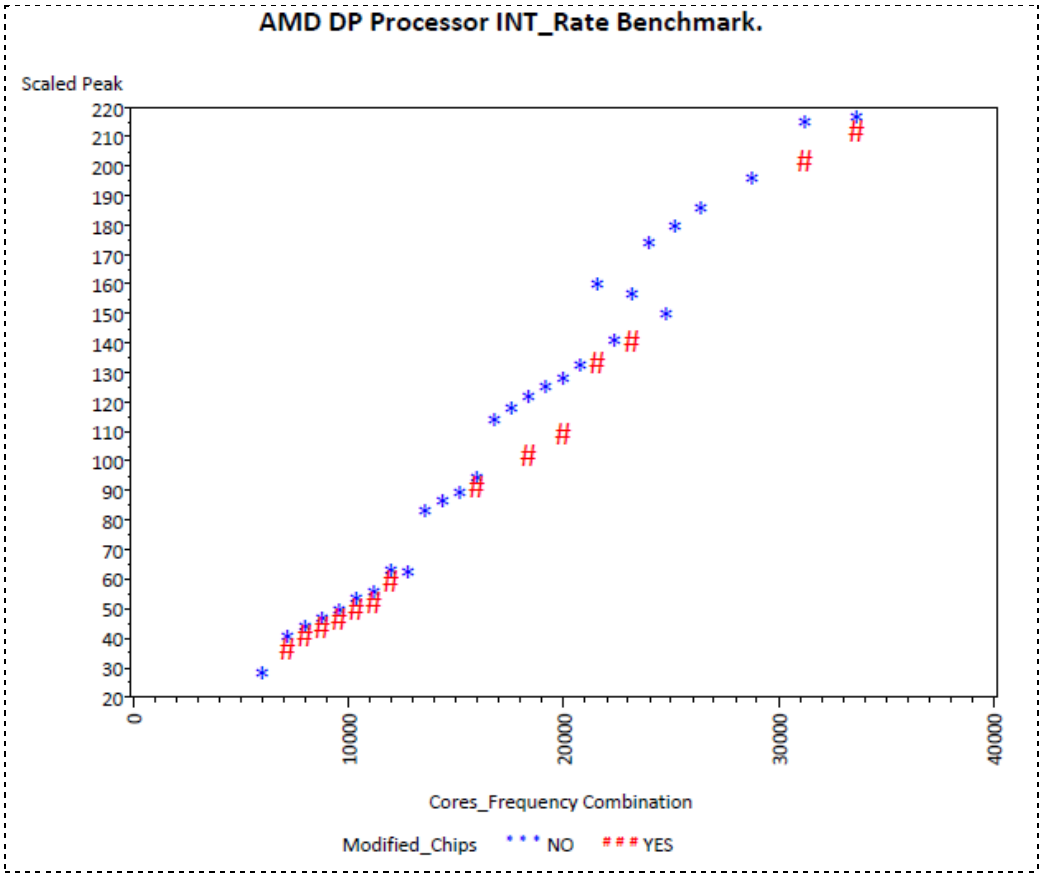


2. AMD DP/MP/UP processors performance for INT_RATE benchmarks

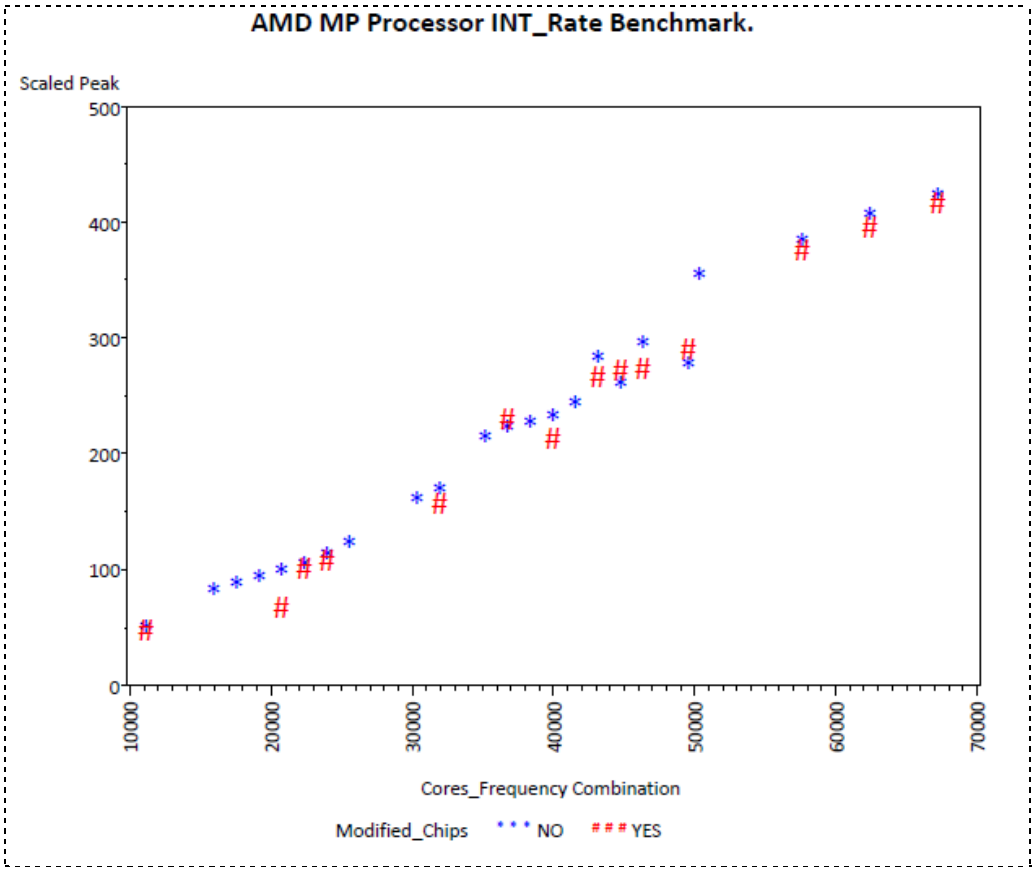
2.1. AMD UP processors performance for INT_RATE benchmarks



2.2. AMD DP processors performance for INT_RATE benchmarks

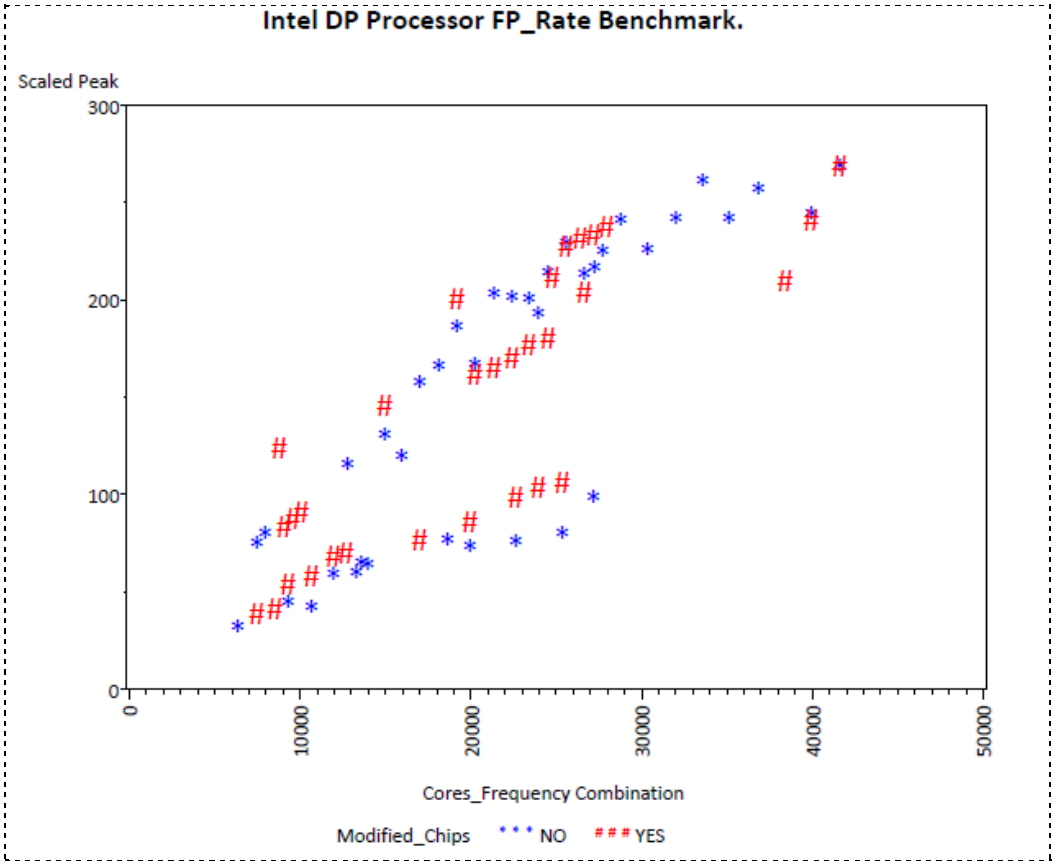


2.3. AMD MP processors performance for INT_RATE benchmarks

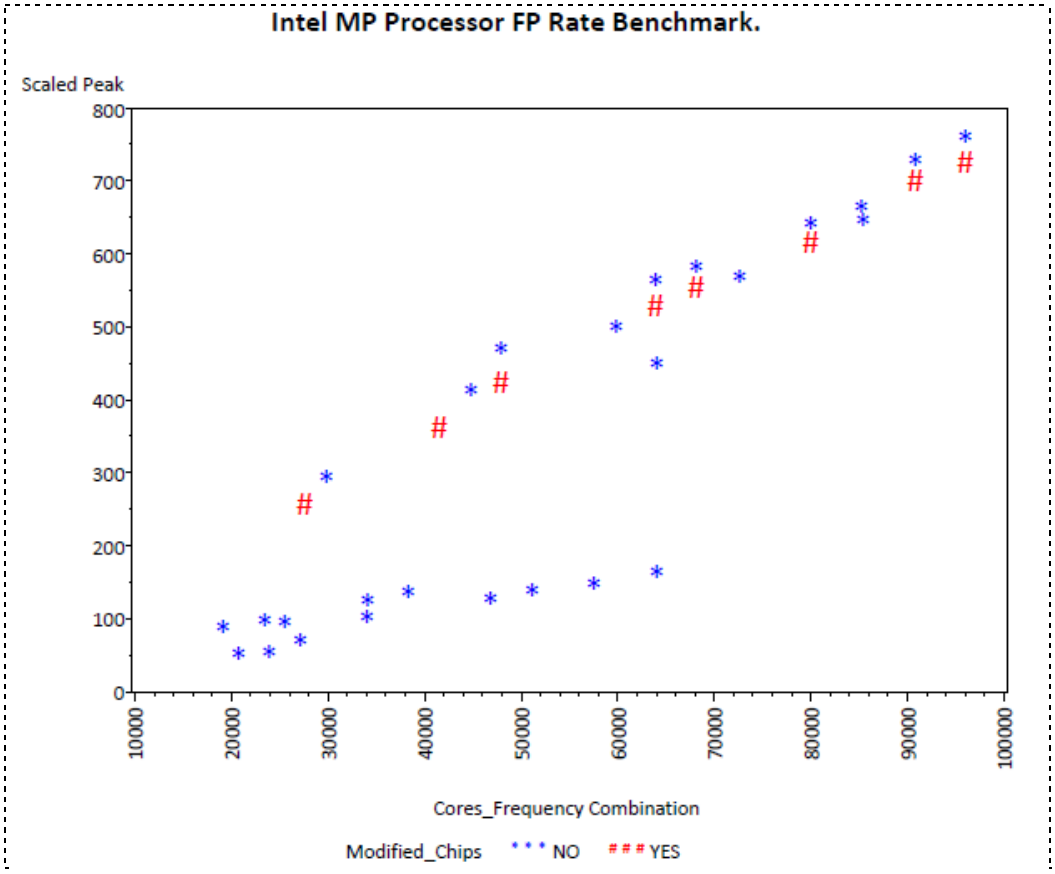


3. Intel DP/MP/UP processors performance for FP_RATE benchmarks

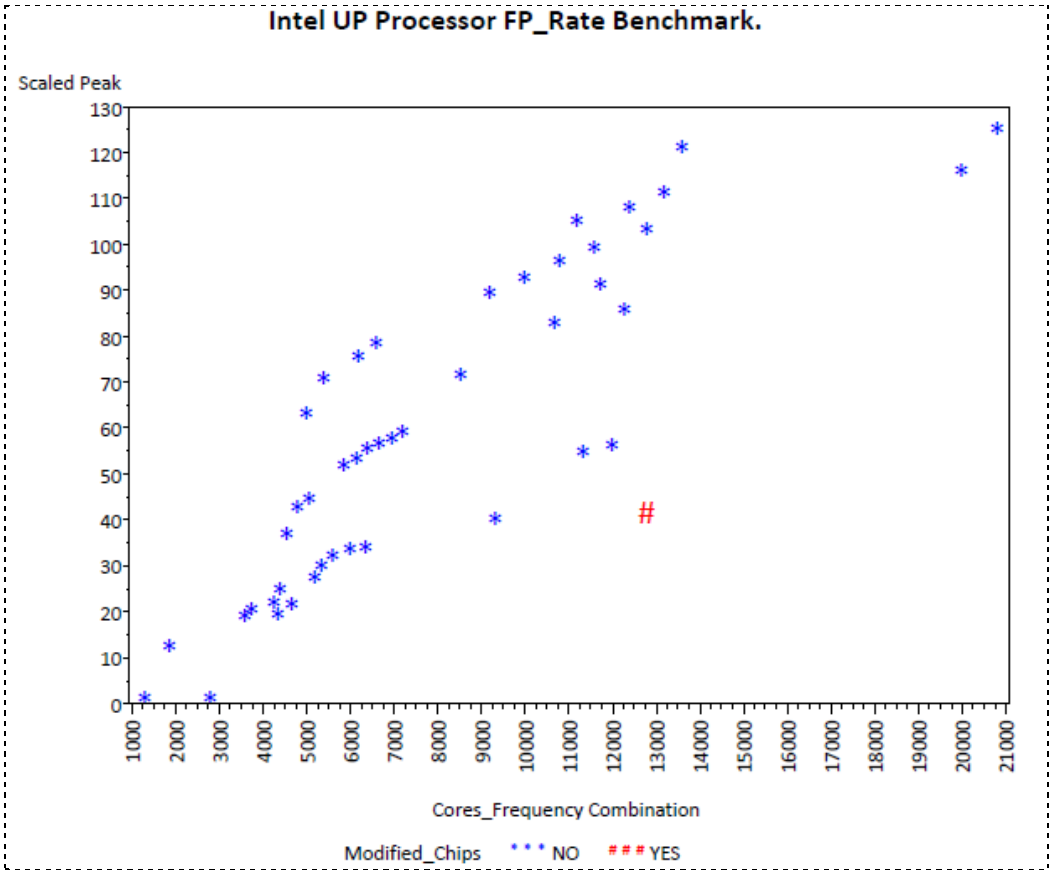
3.1. Intel DP processors performance for FP_RATE benchmarks



3.2. Intel MP processors performance for FP_RATE benchmarks

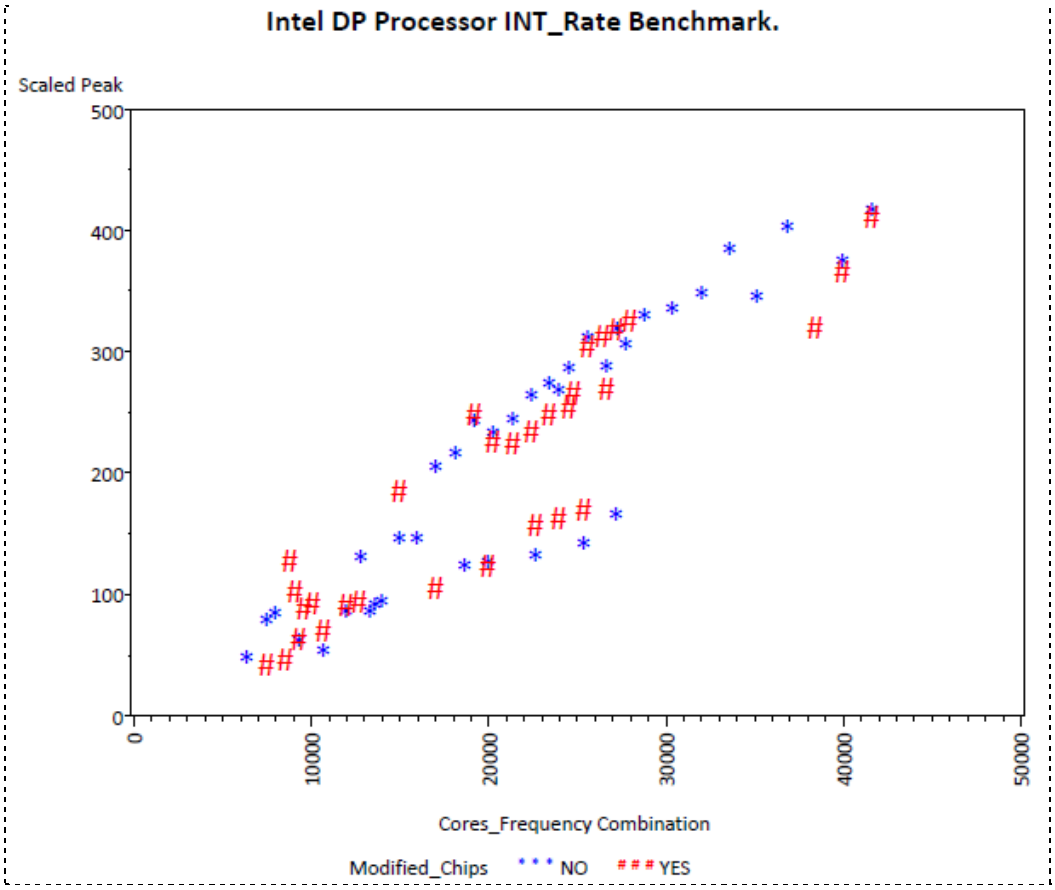


3.3. Intel UP processors performance for FP_RATE benchmarks

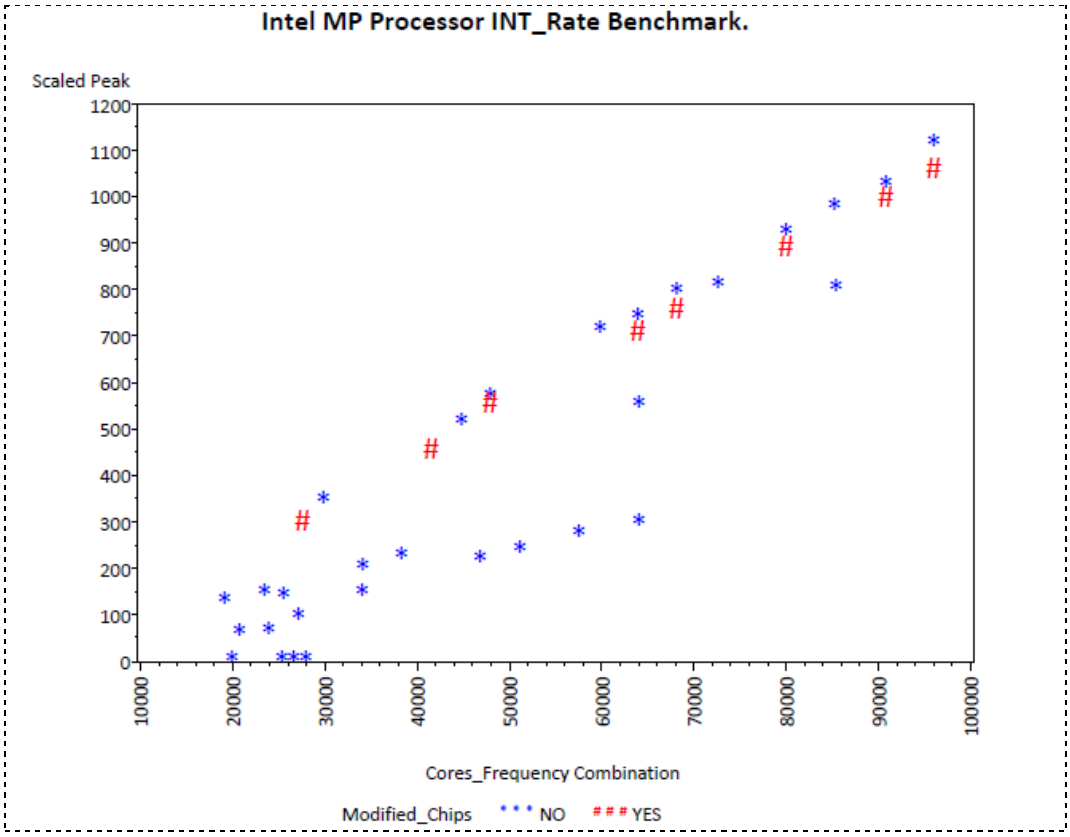


4. Intel DP/MP/UP processors performance for INT_RATE benchmarks

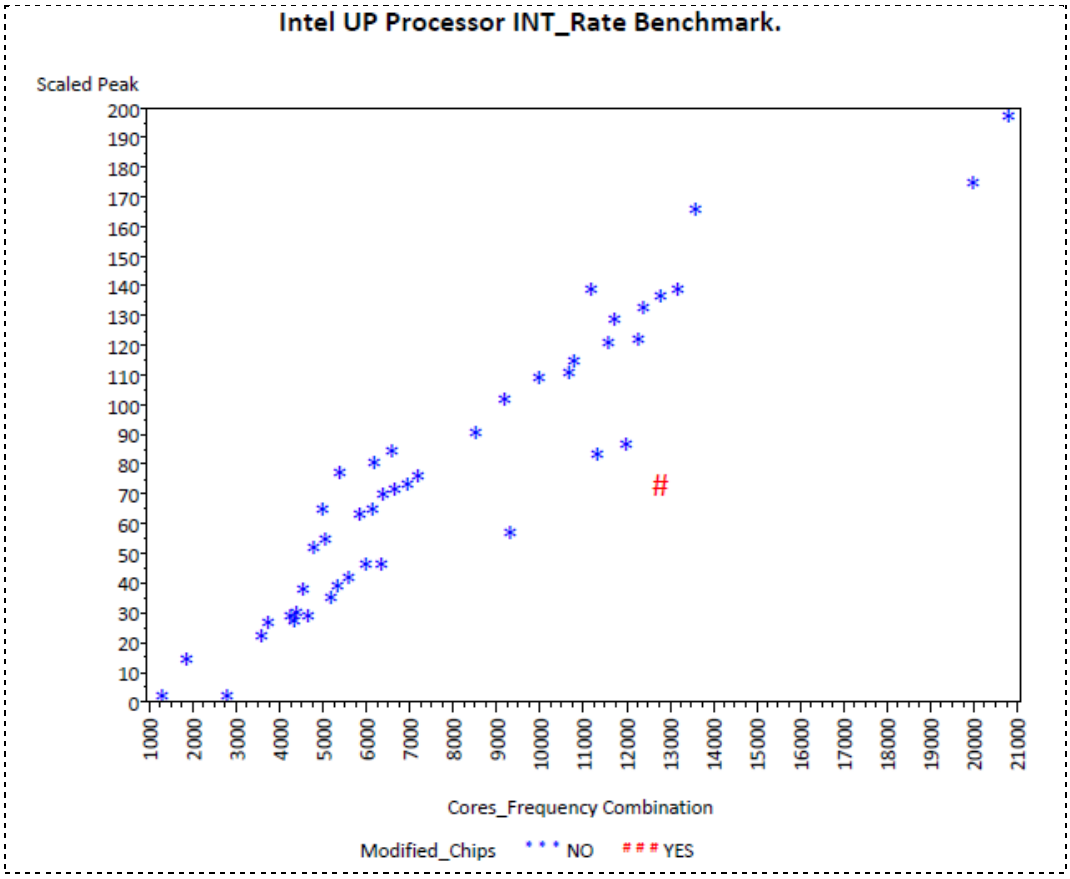
4.1. Intel DP processors performance for INT_RATE benchmarks



4.2. Intel MP processors performance for INT_RATE benchmarks

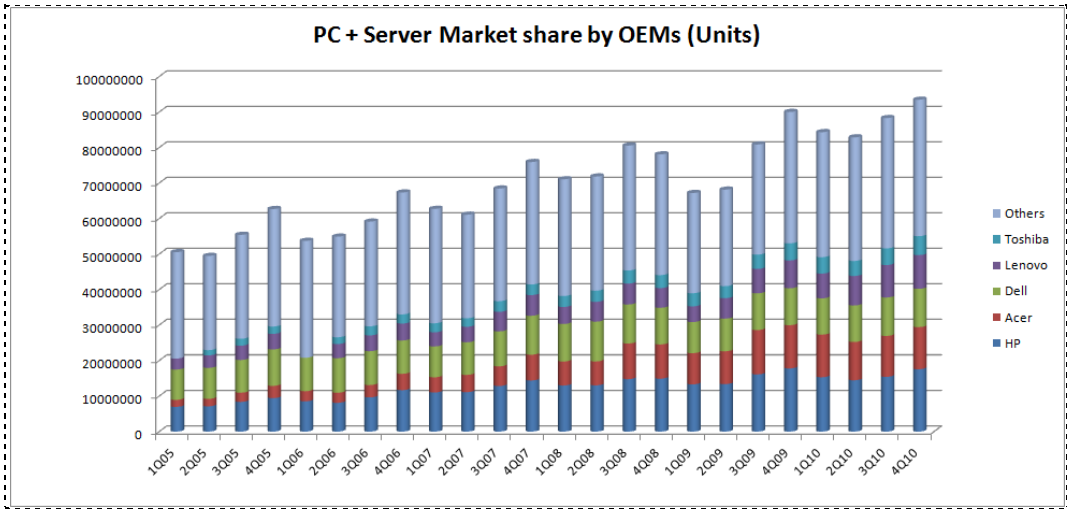


4.3. Intel UP processors performance for INT_RATE benchmarks

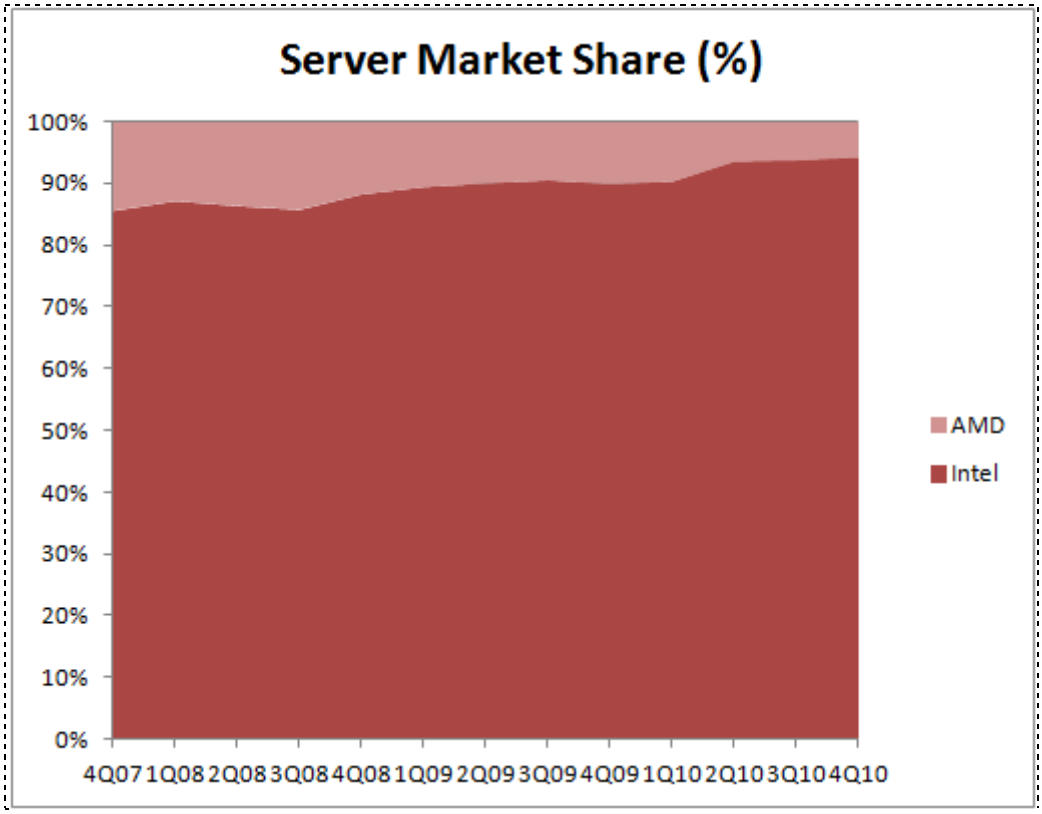


5. Market Segment Share

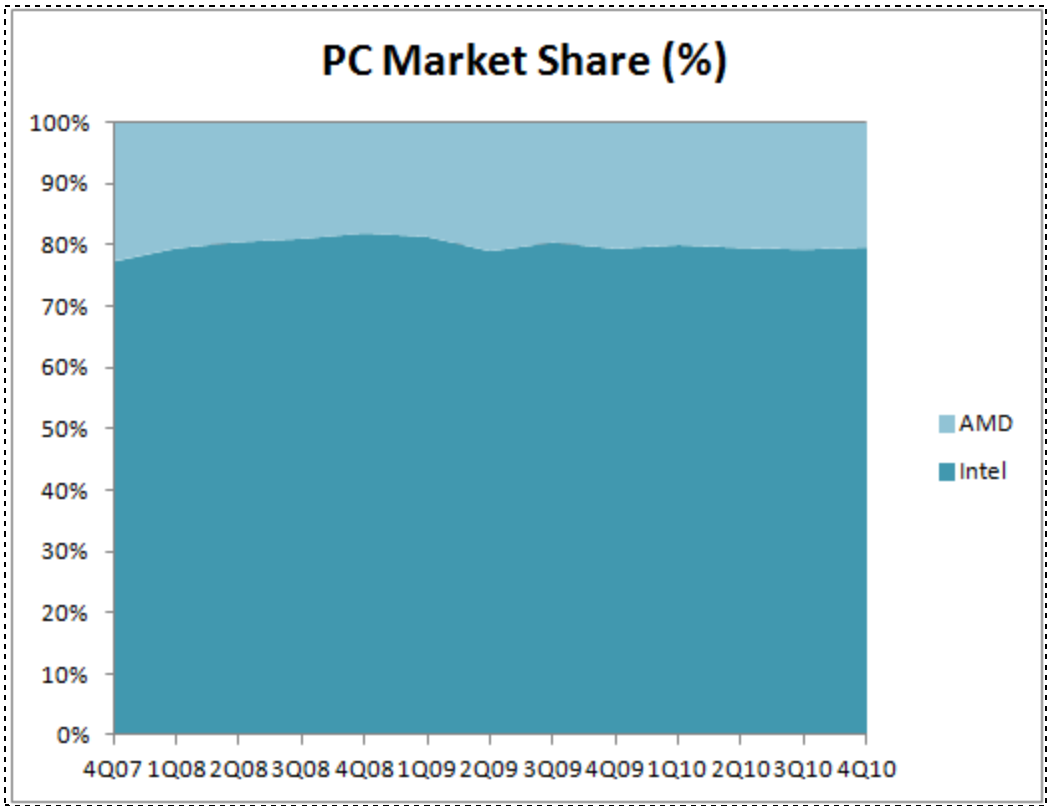
5.1. Market share of OEMs in PC+Server segments



5.2. Market share of Intel and AMD in Server segment



5.3. Market share of Intel and AMD in PC segment



5.4. Market share of Intel in PC and Server segments