IT UNIVERSITY OF COPENHAGEN

Clustering Player Behaviors in Data Streams using K-means in MapReduce

by

Sigurdur Karl Magnusson

A thesis submitted in partial fulfillment for the
degree of Master of Science in IT

in the
Computer Science
Software Development and Technology

Supervisors
Julian Togelius, Rasmus Pagh

April 2013

"Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius – and a lot of courage – to move in the opposite direction."

Albert Einstein

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor...

Contents

Abstract ii

Acknowledgements iii

List of Figures vi

List of Tables vii

Abbreviations viii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Problem Statement . . . 3
- 1.3 Method . . . 4
- 1.4 Contributions . . . 4
- 1.5 Project Outline . . . 5

2 Background Theory 6
- 2.1 Player Behavior Profiles . . . 6
  - 2.1.1 Game Metric . . . 7
- 2.2 Clustering . . . 7
  - 2.2.1 K-means clustering method . . . 8
  - 2.2.2 Player Behaviors . . . 9
- 2.3 MapReduce and Large-Scale Data . . . 10

3 Related Work 12
- 3.1 Clustering Player Behaviors . . . 12
- 3.2 Clustering Large Data . . . 14
- 3.3 Our study . . . 17

4 Methodology 18
- 4.1 Data and Preprocessing . . . 18
  - 4.1.1 Feature selection and behavioral variables . . . 19
- 4.2 K-means algorithm in MapReduce . . . 20
  - 4.2.1 Map . . . 21
  - 4.2.2 Combine . . . 22
  - 4.2.3 Reduce . . . 22
  - 4.2.4 Distance measure . . . 23

4.3 Experiment Set-Up 23

5 Results and Discussion **25**
5.1 Results 25
5.2 Discussion 26

6 Conclusions **28**
6.1 Conclusions 28
6.2 Summary of Contributions 29
6.3 Future Research 29

A Appendix Title Here **31**

Bibliography **32**

List of Figures

List of Tables

Abbreviations

LAH **L**ist **A**bbreviations **H**ere

Chapter 1

Introduction

1.1 Motivation

We live in a world where data is being generated at an amazing rate everywhere around us. Data can describe characteristics of e.g. Internet activities, social interactions, user behaviors in games, mobile phone activities, scientific experiments and measurements from different devices and sensor equipments. Amount of data that is being registered and stored around us is growing massively in volume and complexity. Organizations are storing more and more historic data than before, moving from large databases towards more commonly online distributed file systems or storage web services providing scalability and high availability at commodity costs where *petabytes* (10^{15} = 1.000 *terabytes*) of information can be stored. We live in a world of *Big Data*, exploring unknown patterns and structures without knowing where it will lead us in the future.

"Information is the oil of the 21st century, and analytics is the combustion engine."

Peter Sondergaard, senior VP at Gartner

In digital games, data about in-game user interactions have been logged and behaviors analyzed since the first game came out. Analyzing the user experience and behaviors of players have mostly been done in laboratories in the past, both during game development and after game launch to see if the game was played as designed. Game designs

have becoming increasingly complex in the recent years offering much more freedom to the players by increasing the number of actions available, items to interact with and massively multiplayer online (MMO) persistent worlds that continue to exist after a player exits a game [1, 2]. This complexity generates much more user-centric data than before and is increasingly challenging when evaluating game designs [3, 4]. The user interactions being registered is called *user telemetry* and is translated to *game metrics* as referred in game development, providing detailed and objective numbers, e.g. total playtime, monsters killed, puzzles solved.

Collecting user's telemetry can give very detailed quantitative information on player behavior and using data mining techniques can supplement traditional qualitative approaches with large-scale behavioral analysis [5], for example show where users are getting stuck and finding actionable behavioral profiles [1, 2, 6]. In the recent years user behavior analysis have in part been driven by the emergence of MMO games and Free-to-Play (F2P) games which can have millions of users and objects that can form highly complex interactions. These game models, especially of persistent nature, are monitoring users actions and their behaviors to drive their revenue with subscriptions or offer players to buy virtual items via micro transactions [1, 2, 4, 7].

One way of doing a behavioral analysis is use an unsupervised machine learning technique called clustering. Cluster analysis is a popular exploratory data mining technique that groups set of data objects together in a cluster that are more similar to each other than data objects in other groups [8]. Human beings categorizes or classifies a new object or a phenomenon based on similarity or dissimilarity of the object's descriptive features and is one of most primitive activies of humans [9]. Clustering explores the unknown patterns of the data and provide compressed data representation for large-scale data. In computer games cluster analysis or behavioral categorization can find behavioral profiles that are actionable and give high valuable insights into the game development as well as increasing the monetization [10, 11].

Most clustering algorithms are designed for modern sizes of datasets where the whole data can fit into memory or allows few passes into a database (where each data object is read more than once). It can be very expensive analyzing large-scale datasets and to get answers efficiently then one needs to reduce the set of data to be analyzed, e.g. sample fewer players and have fewer features (dimensions) to be compared. Computations

for large-scale data takes time and needs to be distributed to be able to complete in reasonable amount of time. Google's MapReduce programming model was introduced in 2004 [12] and allows automatic parallelization and distribution of computations on large clusters of commodity computers. Allowing programmers and researchers to easily implement highly scalable algorithms to process large amount of data using the MapReduce model without worrying about handling failures and distributing the data with a large amount of complex code.

1.2 Problem Statement

> How can clustering using incremental k-means find general player behaviors in large-scale behavioral game data in reasonable time?

Considering the massive size of user telemetry data being logged and processed, and the complexity of game designs. There is a knowledge gap when it comes to analyzing such large-scale data efficiently. Number of players are increasing and the complexity of player-game and player-player interactions grows exponentially. One of the largest massively multiplayer online role-playing game (MMORPG) *World of Warcraft* has a population of around 12 million users where players live in a persistent world that can create many millions of different interactions in the game.

User telemetry from games can arrive in daily chunks and need to be processed incrementally (in mini batches). There is a need for algorithms that can process massive amount of data that doesn't fit in a computer memory to extracts knowledge in a reasonable time. The k-means algorithm can find clusters that represent general game behaviors. When implemented in the MapReduce framework, k-means can cluster the data in parallel and is highly scalable with running time increasing linearly with the size of the input.

Our goal with this project is to implement a scalable clustering algorithm to find the general behaviors in a specific real life game dataset in collaboration with GameAnalytics (GA) [13]. The goal is not to implement a complete product but a scalable algorithm that provides information about the general player behavioral profiles for a specific game. Also the algorithm should allow GA to easily develop a product that can cluster general player behaviors in large-scale games.

The success criteria of this project is:

- A scalable k-means clustering algorithm finding clusters describing the general behaviors of a real life game dataset provided by GameAnalytics.
- The general behaviors found must be intuitively interpretable and actionable to game developers.
- The algorithm must be able to process incrementally cluster daily arriving chunks of game metric data.

GameAnalytics is Software as a Service (SaaS) start-up, a data and analytics engine for game studios with its headquarters located in Copenhagen. Analyzing large quantities of game metric data that needs to be processed efficiently returning actionable results to aid game design and development.

1.3 Method

TODO A short description of the methods

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1.4 Contributions

TODO Description of the contributions made in the project

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis,

molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

1.5 Project Outline

References are cited by index in the bibliography and are in order of appearance, e.g. [2] is a citation number two that is referenced in the thesis. Referring to other sections is by the number of that section, e.g. 4.1.2.

The organization of the thesis is as follows:

- **Chapter 2 - Background** Describes a short background theory about clustering player behaviors and the MapReduce framework for large-scale data parallel processing.
- **Chapter 3 - Related Work** Overview of related and recent work regarding clustering player behaviors and large-scale data with k-means as focus.
- **Chapter 4 - Methodology** Our design and implementation work is described; Description of the real game dataset and selection of features, the k-means algorithm in MapReduce and the experimental set-up.
- **Chapter 5 - Results** Results and observations from experiments are explained.
- **Chapter 6 - Conclusions** Conclusions are drawn from the study including future research.

Chapter 2

Background Theory

2.1 Player Behavior Profiles

TODO Describe player behavior

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.1.1 Game Metric

TODO Describe User Telemetry and Game Metric. Features and behavioral variables

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

2.2 Clustering

The goal of clustering is to categories or groups similar objects together into so called clusters (hidden data structures) while different objects belong to other clusters. A cluster are set of objects that are similar to each other, while objects in different clusters are dissimilar to each other. Identifying descriptive features of an object one can compare these features to a known object based on their similarity or dissimilarity based on some criteria. Cluster analysis can be achieved by various algorithms and is a common technique in statistical data analysis that is used in many fields, e.g. machine learning, pattern recognition, image analysis and bioinformatics. In this thesis we focus on a popular clustering algorithm called k-means that is a centroid based clustering algorithm where a cluster is represented by a central vector called centroid.

2.2.1 K-means clustering method

Many clustering methods exists but one of the most popular ones is called *k-means* [14, 15], also known as the Lloyd algorithm [16] which was further generalized for vector quantization [17]. K-means seeks to group similar data points into k partitions or hyperspherical clusters using a distance measure and giving insights into the general distributions in the dataset. A cluster is represented by a centroid that characterize the geometric center of the cluster that is calculated as the mean of all data instances belonging to that cluster. The objective function in k-means is to minimizes the squared error between a cluster's centroid (mean) and its assigned points and over all set of clusters minimizing the Sum of Squared Error (SSE). Let a set of data points $x_i \in \Re^d, i = 1, ..., N$, where each point x is a real number d-dimensional vector and we want to partition them into K clusters $C = \{c_1, ..., c_K\} \in \Re^d$, then the objective function is defined as

$$SSE = \sum_{k=1}^{K} \sum_{x_i \in c_k} \|x_i - \mu_k\|^2$$

where $\|x_i - \mu_k\|$ is a chosen distance measure between a data point x_i and its cluster centroid μ_k (mean). The centroid is calculated as

$$\mu_k = \frac{1}{n_k} \sum_{x_i \in c_k} x_i,$$

where c_k is a cluster number k and its corresponding n_k data points $i = 1, ..., n_k$. The algorithm for the k-means starts by initializing K centroids by choosing random data points from the dataset or according some heuristic procedure. In each iteration of the algorithm it assigns data points to it's nearest centroid by calculating the minimum distance to the K centroids for each instance. After assigning all the data points to clusters the centroids are updated so they represent the mean value of all the points in the corresponding cluster. The algorithm stops the iteration when the centroids do not change from the previous iteration or the error (SSE) is below some specific threshold. Stopping can also be done by predefine a maximum number of iterations to be run.

TODO INSERT PICTURE, showing some simple iterations...

One of the weaknesses of k-means is that it is sensitive of the initial selection of the centroids which can lead to local optimum, that is the algorithm converges and fails

to find the global optimum. One solution is to run the algorithm n times and pick the initialization that gave the lowest SSE result. Another weakness of k-means is that a user has to predefine the number of centers k-means need to cluster, this is most often not known in advance. Many methods exist and most popular one is to run the algorithm with by increasing the k number of clusters to some K and pick a good k candidate using the popular Scree plots [18]. Noise in data and outliers can dramatically increase the squared error and centroids shifting from data distribution in question towards outliers far away, thus representing skewed distributions. Solutions involve removing these noise in preprocessing or normalize the data with the zero mean normalization [8, 18].

The solution to the optimal partition can be found by checking all possibilities using a brute force method but that is a NP-hard problem [19] and cannot be solved in a reasonable time. The k-means algorithm is a heuristic approach for the clustering problem with running time of the algorithm $O(NKdT)$ where T is number of iterations. Usually K, d and T is much less than N, k-means is good for clustering large-scale data because of approximately linear time complexity.

The above implementation of k-means is called the *batch* k-means, where the centroids are updated after all the data points have been assigned. The *online* (incremental) mode of the algorithm processes each data point sequentially. For each data point x the nearest cluster centroid c_{min} is calculated and that centroid is updated right away, defined as

$$c_{min}(old) = \underset{k}{\operatorname{argmin}} \|x - c_k\|$$

$$c_{min}(new) = c_{min}(old) + \eta(x - c_{min}(old))$$

where the cluster centroid c_{min} is updated towards the data point x using the learning rate η which determines the adaptation speed to each data point. The online approach is however highly dependent on the order of which the data points are processed. A variant of this method is used when clustering an endless stream of data where data points arrive one at a time or in chunks.

2.2.2 Player Behaviors

TODO Describe clustering player behaviors with focus using k-means

K-means algorithm have been shown to be very useful in behavioral analysis to give good insights in the general behaviors found in a game [CITE].

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

2.3 MapReduce and Large-Scale Data

TODO Describe MapReduce and Large-Scale data

MapReduce is a programming model introduced by Google in 2004 [12], built on the divide-and-conquer paradigm, dividing a massive task into smaller chunks and process them in parallel. MapReduce enables fault-tolerant distributed computing on large-scale datasets and is a new way to interact with *Big Data* where as old techniques are more complicated, costly and time consuming [12]. Google also introduced along with MapReduce a powerful distributed file system called *Google File System* (GFS) that could hold massive amount of data. This led to a new open source software framework called Hadoop [CITE], which is a end-to-end solution for organizations that want to apply MapReduce. Hadoop builds on the MapReduce and GFS foundation, designed to abstract away much of the complexity of distributed processing running on large clusters of commodity computers.

...

TODO Draw picture of the MapReduce framework

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus

ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Chapter 3

Related Work

Clustering player behaviors and processing large data sets have been researched actively in the recent years. In our work we find the general player behaviors in a the case study using k-means clustering algorithm in the MapReduce framework for high scalability and parallel processing of large-scale data. The behavior of the data can change from day to day like when dealing with an endless stream of data is also discussed. In subsequent sections some of the recent and related work are given a short introduction.

3.1 Clustering Player Behaviors

Many researches have been done on clustering and predicting player behaviors over the last years to get a better understanding which kind of user behaviors are to be found when playing a game that can be actionable for game developers [2, 6, 20–22]. User behavior analysis has becoming increasingly popular in the recent years because of rise of the free-to-play (F2P) genre games in *Facebook* and *Google Play* where populations can be in millions creating complex game interactions [1, 2]. Playing these games are free and many are of persistent nature where the world in the game continues when a player exits. To be profitable these games drive their revenue via micro transactions, e.g. players buying upgrades or virtual items in game for real money [1, 2, 4, 7]. Major game publishers have also been collecting and analyzing large scale of behavior telemetry data but details of their methods are kept confidential [5, 23]. Most available research work

is case-based where a specific algorithm is applied to a specific game and commercial game data sets has only become accessible recently for academic researchers [5].

Predicting player behavior in a major commercial game Tomb Raider: Underworld (TRU) was presented in a study of Drachen et al. [10]. Authors classified 1365 players in a moderate data set into four user behavioral groups using six statistical gameplay features based on core game design as inputs of an emergent self-organizing map to identify dissimilar behavior clusters. Behavior profiles covering 90 percent of the users in the dataset were labeled in game terminology usable for game designers. Mahlmann et al. [11] did a follow up on the research using eight gameplay features and classified behavior of 10,000 players. The authors presented also how to predict behavior based on early play analysis, a popular topic which can be used to prevent churn (attrition) [7].

Analyzing social groups in the highly popular Massively Multiplayer Online Role-Playing Game (MMORPG) World of Warcraft was done by Thurau and Bauckhage [24]. They analyzed how groups (guilds) evolve over time from both American and European based guilds. Their paper is the first study analyzing such amount of data in a MMORPG, analyzing large-scale data gathered on-line from 18 million players belonging in 1.4 million groups over a period of 4 years. Convex-Hull Non Negative Matrix Factorization (CH-NMF) [22] technique was applied to the data to find the extremes rather than averages and the results show no significant cultural difference in formation processes of guilds from either the US or the EU. Interpretability of CH-NMF was more distinguishable and representing archetypal guilds than the more conventional clustering method k-means that represent the cluster centroids with similar characteristic.

Drachen et al. [6] did a clustering analysis for two major commercial games applied to large-scale of high-dimensionality player behavior telemetry. K-means and Simplex Volume Maximization (SIVM) clustering were applied to the MMORPG *Tera* and the multi-player first-person shooter strategy game *Battlefield: Bad Company 2*. SIVM clustering is an adaption of Archetype Analysis (AA) for large-scale data sets to find extreme player behaviors profiles [22, 25]. The authors show the contribution differences from the two algorithms where k-means gives insights into the general distribution of behaviors vs. SIVM showing players with extreme behaviors. The selection of the most important features from the data set were followed by a method suggested by Drachen

et al. [10], behavioral profiles were extracted and interpreted in terms of design language [10, 23].

In a recent study by Drachen et al. [26] the authors compare four different popular methods with purpose of clustering player behaviors and develop profiles from large-scale game metric data set from the highly popular commercial MMORPG World of Warcraft. The data set was collected from mining the Warcraft Realms site, recordings of on-line time and what level each player reached for each day in the years 2005-2010 for approx. 70 thousands of players. The authors selected playtime and leveling speed as their behavioral variables to show a measure of the overall player engagement in the game, where playtime is one of the most important measure for calculating the churn rate [4, 7]. Interpretable behaviors profiles where only generated by the k-means and the SIVM algorithm. The SIVM Archetype Analysis algorithm produces however significantly different behaviors that result in easier interpretation of behavior profiles compared to the k-means algorithm where the centroids are overall similar.

3.2 Clustering Large Data

K-means [14–16] is one of the most studied clustering algorithm out there and is still actively researched. It's a simple algorithm that partition the data into k partitions by minimizing its objective function sum of squared error (SSQ). From its appearance it has been one of the most popular clustering algorithm to research because its ease of interpretation and simplicity [8, 27]. One of the problems with k-means it is a heuristic algorithm and has no guarantee to converge to a global optimum (optimal solution). Many work have been done in researching approximation guarantees (guarantee a approximated solution which is a within a constant-factor of the optimal solution) for k-means both in non-streaming and streaming versions [28–32]. In recent years k-means has also been very popular algorithm to study in the MapReduce framework where the algorithm can easily be applied to cluster large amount of data sets in parallel [12, 33–36].

Guha et al. [37] designed an algorithm in 2003 called STREAM that is based on the divide and conquer strategy to solve the k-median problem (a k-means variant), authors guarantee a approximated guaranteed solution. The algorithm divides the dataset into m pieces of similar sizes, where each of the pieces are independently clustered sequentially

and all the centers from all the pieces are then clustered further. They show a new k-median algorithm called LSEARCH that is used by the STREAM algorithm and is based on a local search algorithm solving the facility location problem [38] to solve the k-median problem. Results show that STREAM LSEARCH produced near optimal quality clusters and better than STREAM k-means also with smaller variance. Compared to the the hierarchical algorithm BIRCH [39] it took 2-3 times longer to run but produced 2-3 times better quality clusters (SSQ), showing superior strength when the goal is to minimize the SSQ like detecting intrusions in networks [40].

In 2009 Ailon et al. [30] extended the the work of Guha et al. mentioned above by introducing a new single pass streaming algorithm for k-means, first of its kind with approximation guarantees. Achieving this they also designed a new algorithm called k-means# that is based on a randomized seeding procedure from the non-streaming algorithm k-means++ by Arthur and Vassilvitskii [29]. The k-means++ chooses k centers non-uniformly whereas k-means# selects $O(k \log k)$ centers and achieves a constant approximation guarantee. In the streaming algorithm they run the k-means# independently on each divided piece of the data to achieve $O(k \log k)$ random centers non-uniformly and use the k-means++ algorithm to find k centers from the intermediate centers from all the pieces of the data set.

Another approach using a *coreset* by selecting a weighted subset from the original dataset such that by running any k-means algorithm on the subset will give near similar results to running k-means on the whole dataset. Ackermann et al. [41] introduced a streaming algorithm called StreamKM++ that is a streaming version of k-means++ algorithm from Arthur et al. [29] to solve k-means on the weighted subset and a new data structure called coreset tree, speeding up the time for the sampling in the center initialization. Their approach was shown to produce similar quality of clusters (in terms of SSQ) as the STREAM LSEARCH [37] algorithm but scaling much better with number of clusters centers.

Shindler et al. [32] proposed an algorithm called *Fast streaming k-means* based on the online facility location algorithm [42] and extends the work of Braverman et al. [31]. Authors prove that their algorithm has a much faster running time and often better cluster quality than the divide and conquer algorithm introduced by Ailon et al.

[30]. The algorithm however show similar average results as StreamKM++ [41] in both running time and quality tho with better accuracy.

Clustering data streams of an unknown length, evolving over time [43, 44] are challenging where it is not possible to access historic data points because of the amount of data arriving continuously. Aggarwal et al. [45] proposed a well-known stream clustering framework called CluStream for clustering large evolving data streams and is guided by application-centered requirements. CluStream has an online component that maintains snapshots of statistical information about micro-clusters (a.k.a. *cluster feature vector* [39]) in a pyramidal time window and an offline component that uses the compact intermediate summary statistics from the micro-clusters to find higher level k clusters using k-means, in a time horizon defined by an analyst. A new high-dimensional highly scalable data stream clustering algorithm called HPStream was also proposed [46]. HPStream uses projected clustering [47], which can determine clusters for a subset of dimensions, to data streams and a new data structure called *fading cluster structure* that allows historical and current data to integrate nicely with a user-specified fading factor.

Another approach clustering evolving streams Zhou et al. [48] presented a algorithm called SWClustering to cluster evolving data streams over so called sliding windows, where the most recent records are considered to be more critical than historic data [49]. Allowing to analyze the evolution of the individual clusters by eliminating influence by outdated historic data points when new data points arrive. Authors show that the CluStream algorithm [45] is much more sensitive to influences of outdated data and is less efficient.

Processing large of amount of data efficiently using parallel processing is an active research and gain much of popularity when the Google's MapReduce programming model was introduced by Dean and Ghemawat [12]. Zhao et al. [33] implemented a parallel version of k-means (PKMeans) in the MapReduce framework and showing their algorithm can be effectively run on large data sets. The Map function calculates the distance to the closest cluster centroid for each data point at a time, after assigning to the clusters a combiner sums up all the data points dimensions for each cluster from that Map function and outputs the key and value $<$*cluster centroid, [sum for all dimensions, number of points]* $>$. The Reduce function then sums up all the intermediate sum values for each cluster and calculates the mean for the new centroids.

Li et al. [50] implemented the algorithm MBK-means in MapReduce using the Bagging ensemble learning method [51], using replacement sampling to generate k new data sets from the original data. K-means algorithm clusters each new data set using the MapReduce framework until convergence and in the end all the centroids from the k sets are merged to form the final k centroids.

Many extensions on the traditional MapReduce framework have been proposed to support efficient algorithms running iteratively [52–57] and incrementally [57–59]. The incremental MapReduce frameworks are interesting and relates to our work since we are incrementally clustering chunks of data but are not under study in this thesis.

3.3 Our study

The algorithm in this thesis is most similar to PKMeans [33] described above, a parallel k-means implementation in MapReduce using a Combine function to reduce the intermediate data sent between the mappers and the reducers. We however implement a parallel k-means algorithm in MapReduce so that each Map function efficiently calculates the distance to nearest cluster centers by calculating the distance matrix between all data points and the cluster centers instead of processing each data point separately. We apply the algorithm to incrementally but non-iteratively cluster player behaviors on theoretically large data sets that arrive daily.

In our work we show k-means is a good approach to provide valuable insights into the general of behaviors for a specific real game data provided by GameAnalytics [13]. Work of Drachen et al. [6, 26] relates to ours when selecting and building important behavioral variables from user's telemetry and extracting behavioral profiles by analyzing and interpret the centroids (basis vectors) from the k-means algorithm running in MapReduce. Additionally we perform experiments with a controlled data set where we have different normal distributions of data arriving separately each day with results and future work are discussed.

Chapter 4

Methodology

TODO Describe introduction to the methodology

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4.1 Data and Preprocessing

TODO Describe the real game dataset and preprocessing

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum

urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

4.1.1 Feature selection and behavioral variables

TODO Describe feature selection and behavioral variables

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices

bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

4.2 K-means algorithm in MapReduce

TODO Describe the k-means algorithm implementation in MapReduce and the relevant pseudo codes and pictures

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac,

nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.2.1 Map

TODO Describe the map function

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.2.2 Combine

TODO Describe the Combine function

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

4.2.3 Reduce

TODO Describe the Reduce function

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

4.2.4 Distance measure

TODO Describe the distance measure used in k-means

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4.3 Experiment Set-Up

TODO Describe the set-up for the experiments

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer

sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Chapter 5

Results and Discussion

5.1 Results

TODO Show results with relevant pictures and what they mean

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

5.2 Discussion

TODO Describe the results with more detailed explanations

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis

egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Chapter 6

Conclusions

6.1 Conclusions

TODO Write Conclusions. Convince the reader that the research question was answered/solved. Write what is relevant to the research question. Use short statements directly related to the research question.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes,

nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6.2 Summary of Contributions

TODO Describe the contributions may overlap Conclusions (Note: maybe move into Conclusions section). Short numbered statements.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

6.3 Future Research

TODO What is the future work. What can be done differently, what needs to be addressed?

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Appendix A

Appendix Title Here

Write your Appendix content here.

Bibliography

[1] Jun H. Kim, Daniel V. Gunn, Eric Schuh, Bruce Phillips, Randy J. Pagulayan, and Dennis Wixon. Tracking real-time user experience (true): a comprehensive instrumentation solution for complex systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 443–452, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1. doi: 10.1145/1357054.1357126.

[2] Anders Drachen and Alessandro Canossa. Evaluating motion: Spatial user behaviour in virtual environments. *International Journal of Arts and Technology*, 4 (3):294–314, 2011. doi: 10.1504/IJART.2011.041483.

[3] Randy J. Pagulayan, Kevin Keeker, Dennis Wixon, Ramon L. Romero, and Thomas Fuller. User-centered design in games. In Julie A. Jacko and Andrew Sears, editors, *The human-computer interaction handbook*, chapter User-centered design in games, pages 883–906. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 2003. ISBN 0-8058-3838-4. URL http://dl.acm.org/citation.cfm?id=772072.772128.

[4] M Seif El-Nasr and Canossa A Drachen A. *Game Analytics: Maximizing the Value of Player Data*. Springer, 2013. ISBN 978-1-4471-4768-8.

[5] Geogios N. Yannakakis. Game ai revisited. In *Proceedings of the 9th conference on Computing Frontiers*, CF '12, pages 285–292, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1215-8. doi: 10.1145/2212908.2212954.

[6] A. Drachen, R. Sifa, C. Bauckhage, and C. Thurau. Guns, swords and data: Clustering of player behavior in computer games in the wild. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, pages 163–170, 2012. doi: 10.1109/CIG.2012.6374152.

[7] Tim Fields and Brandon Cotton. *Social Game Design: Monetization Methods and Mechanics*. CRC Press, 12 2011. ISBN 978-0240817668.

[8] Rui Xu and II Wunsch, D. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005. ISSN 1045-9227. doi: 10.1109/TNN.2005.845141.

[9] M. R. Anderberg. *Cluster Analysis for Applications*. Academic Press, 1973.

[10] A. Drachen, A. Canossa, and G.N. Yannakakis. Player modeling using self-organization in tomb raider: Underworld. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, pages 1–8, 2009. doi: 10.1109/CIG.2009.5286500.

[11] T. Mahlmann, A. Drachen, J. Togelius, A. Canossa, and G.N. Yannakakis. Predicting player behavior in tomb raider: Underworld. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pages 178–185, 2010. doi: 10.1109/ITW.2010.5593355.

[12] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. In *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, OSDI'04, pages 10–10, Berkeley, CA, USA, 2004. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1251254.1251264.

[13] GameAnalytics Aps. Data and analytics engine for game studios, 2013. URL http://www.gameanalytics.com.

[14] E. W. FORGY. Cluster analysis of multivariate data : efficiency versus interpretability of classifications. *Biometrics*, 21:768–769, 1965. URL http://ci.nii.ac.jp/naid/10009668881/en/.

[15] James B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. Mathematical Statist. Probability*, pages 281–297, 1967. URL http://www-m9.ma.tum.de/foswiki/pub/WS2010/CombOptSem/kMeans.pdf.

[16] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982. ISSN 0018-9448. doi: 10.1109/TIT.1982.1056489.

[17] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantizer design. *Communications, IEEE Transactions on*, 28(1):84–95, 1980. ISSN 0090-6778. doi: 10.1109/TCOM.1980.1094577.

[18] J. Han, M. Kamber, and J. Pei. *Data Mining, Second Edition: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2006. ISBN 9780080475585. URL http://books.google.dk/books?id=AfL0t-YzOrEC.

[19] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009. ISSN 0885-6125. doi: 10.1007/s10994-009-5103-0.

[20] Tim Marsh, Shamus Smith, Kiyoung Yang, and Cyrus Shahabi. Continuous and unobtrusive capture of User-Player behaviour and experience to assess and inform game design and development. In *1st World Conference for Fun 'n Games*, Preston, England, 2006.

[21] Olana Missura and Thomas Gärtner. Player modeling for intelligent difficulty adjustment. In João Gama, VítorSantos Costa, AlípioMário Jorge, and PavelB. Brazdil, editors, *Discovery Science*, volume 5808 of *Lecture Notes in Computer Science*, pages 197–211. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-04746-6. doi: 10.1007/978-3-642-04747-3˙17. URL http://dx.doi.org/10.1007/978-3-642-04747-3_17.

[22] C. Thurau, K. Kersting, and C. Bauckhage. Convex non-negative matrix factorization in the wild. In *Data Mining, 2009. ICDM '09. Ninth IEEE International Conference on*, pages 523–532, 2009. doi: 10.1109/ICDM.2009.55.

[23] G Zoeller. Game development telemetry. In *Proceedings of the Game Developers Conference*, 2010.

[24] C. Thurau and C. Bauckhage. Analyzing the evolution of social groups in world of warcraft ő. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pages 170–177, 2010. doi: 10.1109/ITW.2010.5593358.

[25] K. Kersting, M. Wahabzada, C. Thurau, and C. Bauckhage. Hierarchical convex nmf for clustering massive data. In Qiang Yang Masashi Sugiyama, editor, *Proceedings of the 2nd Asian Conference on Machine Learning (ACML–10)*, Tokyo, Japan,

Nov 8–10 2010. URL http://www-kd.iai.uni-bonn.de/pubattachments/477/kersting10acml.pdf. draft.

[26] A. Drachen, C. Thurau, R. Sifa, and C. Bauckhage. A comparison of methods for player clustering via behavioral telemetry. In *Foundations of Digital Games 2013*, 2013.

[27] Lior Rokach. A survey of clustering algorithms. In Oded Maimon and Lior Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 269–298. Springer US, 2010. ISBN 978-0-387-09822-7. doi: 10.1007/978-0-387-09823-4˙14.

[28] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. In *Proceedings of the eighteenth annual symposium on Computational geometry*, SCG '02, pages 10–18, New York, NY, USA, 2002. ACM. ISBN 1-58113-504-1. doi: 10.1145/513400.513402.

[29] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 978-0-898716-24-5. URL http://dl.acm.org/citation.cfm?id=1283383.1283494.

[30] Nir Ailon, Ragesh Jaiswal, and Claire Monteleoni. Streaming k-means approximation. *Advances in Neural Information Processing Systems*, 22:10–18, 2009. URL http://www1.cs.columbia.edu/~rjaiswal/ajmNIPS09.pdf.

[31] Vladimir Braverman, Adam Meyerson, Rafail Ostrovsky, Alan Roytman, Michael Shindler, and Brian Tagiku. Streaming k-means on well-clusterable data. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 26–40. SIAM, 2011. URL http://dl.acm.org/citation.cfm?id=2133036.2133039.

[32] Michael Shindler, Alex Wong, and Adam W. Meyerson. Fast and accurate k-means for large datasets. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2375–2383. NIPS, 2011. URL http://books.nips.cc/papers/files/nips24/NIPS2011_1271.pdf.

[33] Weizhong Zhao, Huifang Ma, and Qing He. Parallel k-means clustering based on mapreduce. In *Proceedings of the 1st International Conference on Cloud Computing*, CloudCom '09, pages 674–679, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-10664-4. URL http://dx.doi.org/10.1007/978-3-642-10665-1_71.

[34] Makho Ngazimbi. Data clustering using mapreduce. Master of science in computer science, Boise State University, March 2009. URL http://cs.boisestate.edu/~amit/research/makho_ngazimbi_project.pdf.

[35] Georgios Christopoulos. Fast, parallel stream clustering using hadoop online. Diploma thesis, Technical University of Crete, July 2011. URL http://titan.softnet.tuc.gr:8080/softnet/GetFile?FILE_TYPE=PUB.FILE&FILE_ID=201.

[36] Grace Nila Ramamoorthy. K-means clustering using hadoop mapreduce. Msc advanced software engineering in computer science, University College Dublin, September 2011. URL http://www.resumegrace.appspot.com/pdfs/kmeansCluster.pdf.

[37] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O"Callaghan. Clustering data streams: Theory and practice. *Knowledge and Data Engineering, IEEE Transactions on*, 15(3):515–528, 2003. ISSN 1041-4347. doi: 10.1109/TKDE.2003.1198387.

[38] M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 378–388, 1999. doi: 10.1109/SFFCS.1999.814609.

[39] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *SIGMOD Rec.*, 25(2):103–114, June 1996. ISSN 0163-5808. doi: 10.1145/235968.233324.

[40] David Marchette. A statistical method for profiling network traffic. In *Proceedings of the 1st conference on Workshop on Intrusion Detection and Network Monitoring - Volume 1*, ID'99, pages 13–13, Berkeley, CA, USA, 1999. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1267880.1267893.

[41] Marcel R. Ackermann, Christiane Lammersen, Marcus Märtens, Christoph Raupach, Christian Sohler, and Kamil Swierkot. Streamkm++: A clustering algorithms for data streams. In *ALENEX*, pages 173–187, 2010. URL http://www.siam.org/proceedings/alenex/2010/alx10_016_ackermannm.pdf.

[42] A. Meyerson. Online facility location. In *Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, FOCS '01, pages 426–, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1390-5. URL http://dl.acm.org/citation.cfm?id=874063.875567.

[43] Charu C. Aggarwal. An intuitive framework for understanding changes in evolving data streams. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 261–, 2002. doi: 10.1109/ICDE.2002.994715.

[44] Charu C. Aggarwal. A framework for diagnosing changes in evolving data streams. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, SIGMOD '03, pages 575–586, New York, NY, USA, 2003. ACM. ISBN 1-58113-634-X. doi: 10.1145/872757.872826.

[45] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu. A framework for clustering evolving data streams. In *Proceedings of the 29th international conference on Very large data bases - Volume 29*, VLDB '03, pages 81–92. VLDB Endowment, 2003. ISBN 0-12-722442-4. URL http://dl.acm.org/citation.cfm?id=1315451.1315460.

[46] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu. A framework for projected clustering of high dimensional data streams. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pages 852–863. VLDB Endowment, 2004. ISBN 0-12-088469-0. URL http://dl.acm.org/citation.cfm?id=1316689.1316763.

[47] Charu C. Aggarwal, Joel L. Wolf, Philip S. Yu, Cecilia Procopiuc, and Jong Soo Park. Fast algorithms for projected clustering. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, SIGMOD '99, pages 61–72, New York, NY, USA, 1999. ACM. ISBN 1-58113-084-8. doi: 10.1145/304182.304188.

[48] Aoying Zhou, Feng Cao, Weining Qian, and Cheqing Jin. Tracking clusters in evolving data streams over sliding windows. *Knowl. Inf. Syst.*, 15(2):181–214, May 2008. ISSN 0219-1377. doi: 10.1007/s10115-007-0070-x.

[49] M. Datar, A. Gionis, P. Indyk, and R. Motwani. Maintaining stream statistics over sliding windows. *SIAM Journal on Computing*, 31(6):1794–1813, 2002. doi: 10.1137/S0097539701398363.

[50] Hai-Guang Li, Gong-Qing Wu, Xue-Gang Hu, Jing Zhang, Lian Li, and Xindong Wu. K-means clustering with bagging and mapreduce. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pages 1–8, 2011. doi: 10.1109/HICSS.2011.265.

[51] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. ISSN 0885-6125. doi: 10.1023/A:1018054314350.

[52] Tyson Condie, Neil Conway, Peter Alvaro, Joseph M. Hellerstein, Khaled Elmeleegy, and Russell Sears. Mapreduce online. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, NSDI'10, pages 21–21, Berkeley, CA, USA, 2010. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1855711.1855732.

[53] Jaliya Ekanayake, Hui Li, Bingjing Zhang, Thilina Gunarathne, Seung-Hee Bae, Judy Qiu, and Geoffrey Fox. Twister: a runtime for iterative mapreduce. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 810–818, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-942-8. doi: 10.1145/1851476.1851593.

[54] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, HotCloud'10, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1863103.1863113.

[55] Yingyi Bu, Bill Howe, Magdalena Balazinska, and Michael D. Ernst. Haloop: efficient iterative data processing on large clusters. *Proc. VLDB Endow.*, 3(1-2): 285–296, September 2010. ISSN 2150-8097. URL http://dl.acm.org/citation.cfm?id=1920841.1920881.

[56] Yingyi Bu, Bill Howe, Magdalena Balazinska, and Michael D. Ernst. The haloop approach to large-scale iterative data analysis. *The VLDB Journal*, 21 (2):169–190, April 2012. ISSN 1066-8888. URL http://dx.doi.org/10.1007/s00778-012-0269-7.

[57] Cairong Yan, Xin Yang, Ze Yu, Min Li, and Xiaolin Li. Incmr: Incremental data processing based on mapreduce. In *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*, CLOUD '12, pages 534–541, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-0-7695-4755-8. URL http://dx.doi.org/10.1109/CLOUD.2012.67.

[58] Pramod Bhatotia, Alexander Wieder, Rodrigo Rodrigues, Umut A. Acar, and Rafael Pasquin. Incoop: Mapreduce for incremental computations. In *Proceedings of the 2nd ACM Symposium on Cloud Computing*, SOCC '11, pages 7:1–7:14, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0976-9. URL http://doi.acm.org/10.1145/2038916.2038923.

[59] Pramod Bhatotia, Marcel Dischinger, Rodrigo Rodrigues, and Umut A Acar. Slider: Incremental sliding-window computations for large-scale data analysis. *MPI-SWS-2012-004*, September 2012. URL http://www.mpi-sws.org/tr/2012-004.pdf.