# Міністерство освіти і науки України

# Національний університет "Запорізька політехніка"

## Кафедра програмних засобів

ЗВІТ

З лабораторної роботи № 1

З дисципліни " **Групова динаміка і комунікації** " на тему:

“ОФОРМЛЕННЯ ДОКУМЕНТАЦІЇ ЗА ДОПОМОГОЮ ТЕКСТОВОГО РЕДАКТОРА”

Виконав

Студент групи КНТ-119 I.I.Дрокiн

Прийняв:

к.т.н., доцент А. О. Олійник

**Мета роботи**

1. Вивчити основні можливості сучасних текстових редакторів.

2. Навчитися редагувати документи відповідно до діючих тандартів за допомогою засобів текстового редактору.

**Завдання до роботи**

Особисто обрати документ для редагування (обсягом не менше ніж 30 сторінок). Обраний документ обов’язково повинен бути структурований на розділи (заголовки першого рівня), підрозділи (за- головки другого рівня), пункти (заголовки третього рівня) та підпункти (заголовки четвертого рівня). Крім того документ, що редагується, повинен містити наступні елементи:

– таблиці;

– рисунки;

– формули;

– нумеровані та марковані списки.

У випадку, якщо обраний документ не містить вказаних елементів і не є структурованим, необхідно самостійно додати ці елементи, узгодивши їх з текстом документу, та виконати розбиття документу на розділи, підрозділи, пункти та підпункти для його подальшого редагування відповідно до діючих стандартів.

Ознайомитися зі стандартом підприємства СТП 15-96 «Пояснювальна записка до курсових і дипломних проектів. Вимоги й правила оформлення».

Відредагувати текст відповідно до СТП 15-96, використовуючи засоби текстового редактору, зокрема створені стилі.

**Короткі теоретичні відомості**

Відповідно до СТП 15-96 та діючих норм **основний текст** документу набирається з використанням шрифту Times New Roman розміром 14 пунктів із міжрядковим інтервалом 1,5, розмір абзацу – 1,5 см.Розмір полів документа: верхнє, ліве і нижнє – не менше 20 мм, праве – не менше 10 мм.

**Заголовки.** Текст пояснювальної записки має бути структурований на розділи та підрозділи, які повинні мати заголовки. Пункти і підпункти також можуть мати заголовки.

**Нумерація** рисунків, таблиць, формул має бути наскрізною вмежах розділу; посилання на друковані джерела мають подаватися вквадратних дужках. Бібліографічні описи в переліку посилань наво-дять за порядком першого згадування в тексті.

**Списки.** За необхідністю можуть бути використані переліки. Перед переліком ставлять двокрапку. Кожна позиція переліку повинна мати абзацний відступ і починатися або з малої літери українського алфавіту з дужкою, або з дефісу (перший рівень деталізації).

**Фрагмент відредагованого тексту**

SWARM: Adaptive Load Balancing in Distributed Streaming Systems for Big Spatial Data

1 Introduction

The recent growth in spatial data has been phenomenal due to the proliferation of GPS-enabled devices, e.g., smartphones, smart watches, health monitors, and connected vehicles. Also, social networks generate huge deluge of spatial data, e.g., 500 million tweets are created daily, and they can be geotagged [[5].](#page30) This growth leads to the development of location-based services, e.g., Internet search engines that return results based on user location, self-driving cars, video games (e.g., Pokemon GO), and ride-sharing services. Five billion Google search queries are generated every day [[5].](#page30) Supporting these services place a huge demand on developing real-time, e cient, and scalable systems for processing location-based queries. Therefore, there is a growing demand to develop new systems that are optimized to process big spatial data instead of using general-purpose systems that are not tunable for the needs of spatial data [[25].](#page32)

Distributed data streaming systems have the potential to provide real-time scalable solutions. There is an increasing number of spatial applications that are being implemented using these systems. Examples include Storm [[38],](#page33) Twitter Heron [[21],](#page32) and SparkStreaming [[44].](#page33)

|  |  |  |
| --- | --- | --- |
| **6:00 am (New York)** |  | **6:00 pm (New York)** |
|  |  |  |

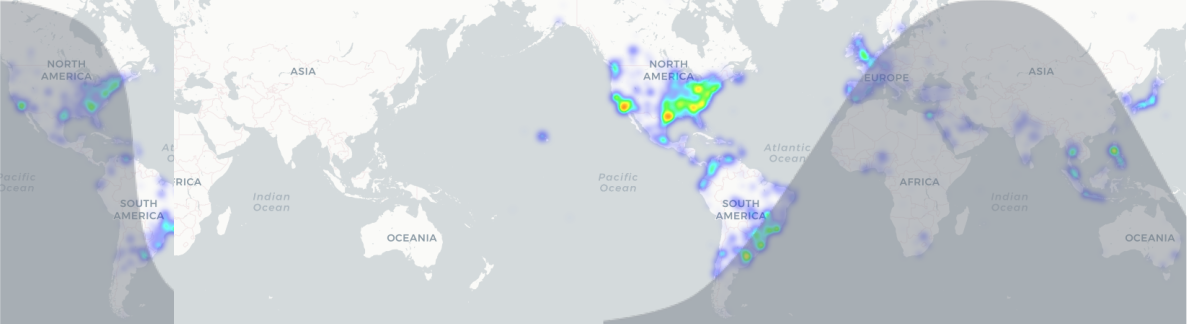


Figure 1: Heatmap of tweets during di erent times

Our Goal. We propose SWARM, a Spatial Workload-aware Adaptive Routing Manager. SWARM is a layer that can be integrated into any distributed data streaming system that processes spatial data. SWARM adaptively load balances the workload among the available machines. SWARM is generic and does not depend on a speci c spatial application. It can be used directly with minimal changes to the code of the spatial application. SWARM achieves high machine utilization. This leads to high performance, low response time, and the handling of larger volumes of spatial data and queries.

Challenges and Contributions. SWARM addresses the following challenges:

1. Let Ln denote the number of complete τ-exceptional sequences in mod Λ2 n. Then Ln is given by the closed formula.
2. The first set contains the indecomposable projective modules Pj for 1 ≤ j ≤ n.
3. Let Gn denote the number of complete τ-exceptional sequences in mod Γ2 n . The exponential generating function of Gn is as follow.

Variability of optimal partitioning plan due to changes in workload distribution: The rapid changes in the distribution of spatial data and queries directly a ect the optimality in the way the data is partitioned. SWARM adaptively and incrementally alters the data partitions according to the workload changes to achieve higher throughput. SWARM uses a cost model to adaptively change the data partitions to achieve an optimized partitioning.

SWARM distributes the system workload using a cost model that depends on the amount of data points in each partition. The cost model gives higher weight to partitions having a high number of queries. Moreover, the cost model predicts the future workload of each partition based on the workload history. This prediction serves as a scale factor for the overall cost and workload of each partition. Assume that we have a distributed streaming system, say S, that has a set M of executor machines. Each machine m 2 M holds some partitions Pm, where jPmj = nm, nm is the number of partitions in Machine m. Each partition p 2 Pm, locally maintains some statistics. The ost estimate C(p) of a partition p is as follows:

(2) 

is the number of points in Partition p, Q(p) is the number of queries that overlap p, and P rob(p) is the probability that new data and queries land in p. P rob(p) depends on the arrival of data and queries during the last round of repartitioning. Note that the workload history is captured via N and Q while P rob is a weighting factor to the cost of this history. The e ect of old data can fade with time as discussed in Section [4.2.2.](#page11) P rob(p) is estimated as follow:

(2)  where R(p) and R(S) are the number of data points and queries received by p and all of S, respectively, during the last round of repartitioning. R(S) is computed as follows.

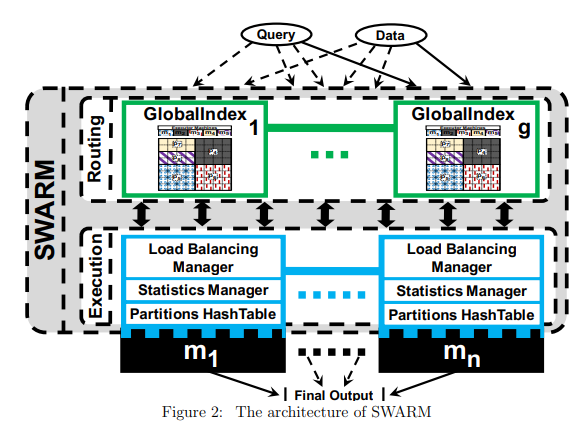
4 SWARM Architecture

SWARM can work with any distributed streaming system that processes spatial data. It does not require changing the original code of the system’s executor machines, e.g., their indexes, their way of handling data or processing queries. Refer to Figure [2.](#page6) SWARM is composed of two layers, the routing layer and the execution load-balancing layer. SWARM replaces the partitioning layer of spatial streaming applications (that have a partitioning layer). SWARM is placed on top of the original executor machines m1; ; mn and directly receives the incoming

* We will also take all objects to be basic where possible and they will be considered up to isomorphism.
* The main result of this note is to establish the type I property for algebraic groups. Let K be an algebraically closed field.
* Theorem 4 generalizes several results from [vD19] and provides a positive solution to a much stronger version of Conjecture 14.1 stated there.
* Every projective unitary representation of G can be lifted to an ordinary unitary representation of a central extension of G
* The following result is part of the so-called Mackey machine; it is a basic tool which reduces the determination of the factor (or irreducible) representations of G to the determination of the factor (or irreducible) representations of subgroups of G.
* SWARM’s second layer contains load-balancing units above the system’s original executor machines m1; ; mn. Each unit has a load-balancing manager, a statistics manager, and a HashTable to index the partitions. Each unit communicates with all other load-balancing and GlobalIndex machines. The unit receives data points and queries only for the partitions that are under its control. The unit’s statistic manager of updates its statistics given the new object, as in Section [4.2.](#page9) Moreover, the unit uses a HashTable to identify the partition(s), say p, that overlap the received object, as in Section [4.1.](#page7) Then, the original application code in the executor machine processes this object on p. The load-balancing manager computes the cost of its executor machine periodically using the cost model in Section [3](#page4) using local statistics. It shares this cost with one GlobalIndex machine. Executor machines with the highest and lowest workloads, say mH and mL, respectively, are identi ed. mH moves part of its workload to mL, as explained in Section [4.3.](#page16)

|  |  |  |
| --- | --- | --- |
| SWARM’s first layer | SWARM’s second layer | SWARM’s third layer |
| GlobalIndex | LocalIndex | MaxIndex |
| Load Balancing Manager | Statistics Manager | Partitions Hash Table |
| Load Lazing Manager | Dynamycs manager | 256x Hash Table |
| mH | mL | mL |

* 4.3 explains that futher SWARM



4.1 Indexing and Initialization

SWARM does not require prior knowledge about the distribution of the incoming data or queries. Initially, SWARM divides the whole spatial area evenly among all executor machines. This section introduces the global and local indexes used by SWARM.

4.1.1 The Global Index

SWARM uses a 2D spatial grid index in each GlobalIndex machine to divide the space into grid cells of a prede ned size C1 C2 (refer to Figure [3)](#page7). This global index replaces the partitioning index of spatial applications that have a partitioning layer. As in Figure 3a, each cell points to a partition that covers this cell. A partition has a unique ID, partition borders, and the ID of the executor machine that handles the partition. Thus, it takes O(1) operations to route an object. Figure 3b gives an example for initial con guration of the index in the GlobalIndex machines of a system with 5 executor machines m1; ; m5 and 5 partitions p1; ; p5. The patterns (colours) of the partitions link them to the executor machines that control them.

1.4.2.3 Correctness of the Statistics

In this section, we prove the correctness of the statistics that SWARM collects and maintains about data points and queries. To show that, we need to prove that the maintained statistics always represent the true number of data points and queries without any over- or under-counting.

First, we prove the correctness of the statistics for data points. Assume that we have a partition that has k rows and only one column. This results in k cells in total as in Figure [7.](#page13)

Finally, let qo(i) be the number of queries whose upper and lower boundaries do not intersect celli but their ranges overlap celli. Therefore, the true number, q(i)(8) of queries that intersect celli is the sum of these four variables, i.e.,



**Висновки i контрольні запитання**

Висновновок: Я ознайомився з роботою в програмі Microsoft Word і зі стандартом СТП 15-96. Отримав нові знання по роботі зі списками, таблицями, заголовками та стилями.

1.5.2 Як задати масштаб перегляду документу?

1. У рядку стану програми Office клацніть повзунок **масштабування** .
2. Протягніть повзунок до потрібного відсотка масштабування. Натисніть **кнопку** або **+** , щоб збільшити поступове збільшення.

1.5.3 Який пункт меню призначений для встановлення інтервалу між рядками тексту?

**Головна -> Iнтервал**

1.5.4 Як встановити відстань між абзацами 18 пт?

**Дизайн-> Iнтервал мiж абзацами-> Призначені для користувача інтервали для абзацу -> Розмiр 18пт**

1.5.7 Як підкреслити текст пунктирною лінією зеленого кольо-

ру?

Натиснуты на трикутник бiля підкреслиного накреслення та обрати пунктирну лінію. Далi натиснути на колiр підкреслинння та обрати зеленый.

1.5.11 Яким чином можна змінити регістр тексту?

1. Виділіть текст, у якого потрібно змінити регістр.
2. Перейдіть на вкладку **основне** Кнопка "змінити буквиці" для O365 і Mac **змінити регістр** #a0.
3. Виконайте одну з таких дій:
   * Щоб починати речення з великої букви та залишити решту букв у нижньому регістрі, клацніть **Як у реченнях**.
   * Щоб великі букви в тексті змінилися на малі, клацніть **усі малі**.
   * Щоб усі букви в тексті набули верхнього регістра, клацніть **УСІ ВЕЛИКІ**.
   * Щоб починати кожне слово з великої букви та лишати решту букв малими, клацніть **Кожне Слово З Великої**.
   * Щоб переключитися між регістрами клацніть **зМІНИТИ рЕГІСТР**.