Mid Term Examination  
01219449 Principles of Software Architecture  
Department of Computer Engineering, Faculty of Engineering, Kasetsart University  
13.00 – 16.00, Tuesday 1st March 2022

Instructions: This is an open-book and open-Internet examination. Computers, text books, lecture notes, and Internet communications are ALLOWED in this examination. But any types of verbal or non-verbal human communications are PROHIBITED during the examination time.  
  
Using the designated communication channel (i.e. the Google Meet link), you must open the video streaming of yourselves (your face) with a proper dress during the entire examination time. You must set up your environment for this examination in a quiet room without any disruption or noise distraction.

Read carefully and answer all questions in these sheets.

There are 3 questions each with a score of 100. Therefore, the total score is 300 points.

1. You need to fill in the following tables and answer the below questions.

1.1 Analyze and compare advantages and disadvantages of three SPA frameworks, namely  
 "React, Vue & Svelte" by filling the following tables as good as you can.

React

|  |  |  |  |
| --- | --- | --- | --- |
| characteristics, quality attributes, or features | Your relative judgements (one of the five level -  Excellent, Great, Medium, Poor, Very Poor) | Evidence for pros  (describe/provide evidence why it is good) | Evidence for cons (describe/provide evidence why it is not so good) |
| State Management | Poor |  | <https://www.toptal.com/react/react-state-management-tools-enterprise>  Re-rendering everything when react detected change in component, problems may arise when the state should be shared among many components. |
| Routing | Poor |  | <https://www.geeksforgeeks.org/reactjs-router/>  Routing is not done in react itself but it's come from external library called "React Router" |
| How to set the project | Poor |  | The framework does very little support for you because goal of react is to be as lightweight and as minimalist as possible. And many things in react are forced to do from scratch and need to import from outside library. |
| Learning Curve for New Developers | Medium |  | There are not many suggestion for new developer. But it popular and have community, so if you found issues someone could help you. But React documentation is not friendly for new devs. |
| Size of Dev Community (Number of Developers) | Excellent | <https://reactjs.org/community/support.html> |  |
| Code Size (Maintainability) | Medium | <https://charlypoly.com/publications/build-maintainable-react-apps-series>  There are a lot of suggestion from community. |  |
| Performance | Medium | While developing, It able to save the running state. | But when changes are made, it re-rendering everything on the page. it has some jerky |
| Accessibility | Medium | <https://reactjs.org/docs/accessibility.html> |  |

Vue

|  |  |  |  |
| --- | --- | --- | --- |
| characteristics, quality attributes, or features | Your relative judgements (one of the five level -  Excellent, Great, Medium, Poor, Very Poor) | Evidence for pros  (describe/provide evidence why it is good) | Evidence for cons (describe/provide evidence why it is not so good) |
| State Management | Excellent | built in natively (this included animations and transitions), reactive |  |
| Routing | Excellent | built in natively |  |
| How to set the project |  |  | There are still limits your versatility. (forcing their own way of doing things) |
| Learning Curve for New Developers | Great | Giving us a lot of suggestions on how they want us to set up the project, also documentation far better than react. |  |
| Size of Dev Community (Number of Developers) | Medium | Community for Vue is smaller than react. | But you can still find solution for your issues |
| Code Size (Maintainability) | Great | Vue giving you the suggestions on how to minimize it. |  |
| Performance | Great | Better than React  <https://jelvix.com/blog/js-frameworks-is-vuejs-better-than-react> |  |
| Accessibility | Medium |  | <https://vuejs.org/guide/best-practices/accessibility.html> |

Svelte

|  |  |  |  |
| --- | --- | --- | --- |
| characteristics, quality attributes, or features | Your relative judgements (one of the five level -  Excellent, Great, Medium, Poor, Very Poor) | Evidence for pros  (describe/provide evidence why it is good) | Evidence for cons (describe/provide evidence why it is not so good) |
| State Management | Excellent | <https://www.youtube.com/watch?v=AdNJ3fydeao> |  |
| Routing | Excellent |  |  |
| How to set the project | Great | There also have offer sandbox to initial your project and some video tutorials. |  |
| Learning Curve for New Developers | Great | Documentation is very good. | They are new, so they have a epic update version more frequency. And community I so small. |
| Size of Dev Community (Number of Developers) | Very Poor | Small! |  |
| Code Size (Maintainability) | Excellent |  |  |
| Performance | Excellent |  |  |
| Accessibility | Excellent | When we run build its create build folder files that can be more effective to run inside of the user's browser |  |

1.2 Why React is not react enough?

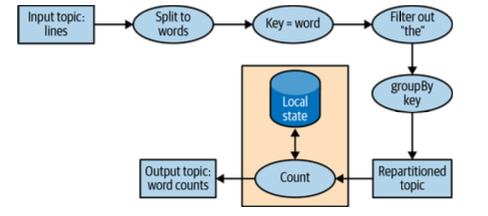
Because instead of only re-render the changed part, react re-rendering everything when detected change in a component. Even though it still save the state, it has some jerky.

1.3 React, Vue and Svelte - which is the best and why? You should mention also about the meta framework e.g. Next.js and SvelteKit.  
  
There is no the best framework, because it depends on what your project. For example. If your project build on enterprise level applications, you should use React. If you want something easy to learn and a lot more intuitive use Vue. And if performance is your major concern, you should use Svelte.

But according to the <https://www.youtube.com/watch?v=LSiV8A9Zqzw&t=4s> video, Phillip Rubin recommends you using meta framework (framework that build on top of vanilla framework) which easy to custom structure and API routes, do full stack dev, faster to develop than vanilla framework. Also, some could have static site generation, server-side rendering, and other convenient tools.

2. Every stream processing application implements and executes on topology. Topology, as it is called in Apache Kafka event-streaming framework (also called DAG, or directed acyclic graph, in other stream processing framework), is a set of operations and transitions that every event moves through from input to output.

The figure below is the topology for a simple word-count stream processing application with its source code shown on the GitHub here https://github.com/gwenshap/kafka-streams-wordcount.



2.1 Explain every line of code as appeared in https://github.com/gwenshap/kafka-streams-wordcount how it works, and also refer to the topology figure above.

Text

Description automatically generated

2.2 Customer Service (Batch Processing vs Stream Processing)

Suppose that we just reserved a room at a large hotel chain, and we expect an email confirmation and receipt. A few minutes after reserving, when the confirmation still hasn’t arrived, we call customer service to confirm our reservation. Suppose the customer service desk tells us, “I don’t see the order in our system, but the batch job that loads the data from the reservation system to the hotels and the customer service desk only runs once a day, so please call back tomorrow. You should see the email within 2–3 business days.”

This doesn’t sound like very good service, yet we’ve had this conversation more than once with a large hotel chain. What we really want is for every system in the hotel chain to get an update about a new reservation seconds or minutes after the reservation is made, including the customer service center, the hotel, the system that sends email confirmations, the website, etc. We also want the customer service center to be able to immediately pull up all the details about any of our past visits to any of the hotels in the chain, and the reception desk at the hotel to know that we are a loyal customer so they can give us an upgrade.

Building all those systems using stream processing applications allows them to receive and process updates in near real time, which makes for a better customer experience. With such a system, the customer would receive a confirmation email within minutes, their credit card would be charged on time, the receipt would be sent, and the service desk could immediately answer their questions regarding the reservation.

Instruction : design and describe a software architecture to connect the following systems

1) the hotel-chain reservation website

2) the software application of each individual hotel (e.g. used by the reception desk etc.)

3) the email system

4) the customer service center application (e.g. used by the customer service)

Using Apache Kafka event-streaming platform for enterprise integration to enable send events (you need a producer), receive and process updates (you need a consumer) in near real time (asynchronous communication through Apache Kafka) to solve the above problem.

Graphical user interface, text, application, Word

Description automatically generated

Flow:

1. Reservation had sent from reservation website to Apache Kafka (reservation topic)
2. Kafka logging the reservation to
   1. database as new item with key “confirm=false”
   2. email system to send confirmation to the customer
   3. notify each individual hotel
3. Customer App and individual hotel can read reservations
4. After the customer send confirmation email, Kafka will
   1. Change item states from database (“confirm=true”)
   2. email system to send notify to the customer
   3. notify each individual hotel

P.S. if Reservation detail change, use the almost same flow, change update key from “confirm” to something else –detail, date, customer, etc.

3. Referring to the George Fairbanks' lecture, a cloud provider company named Rackspace has built three generations of systems to handle the customer queries.

3.1 Draw the pictures of those three software architectures with key words in the pictures.

Chart

Description automatically generated with low confidence

3.2 Describe and discuss about the “data freshness”, “scalability” and “ad-hoc query ease” qualities of those three versions of architectures.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Version 1 | Version 2 | Version 3 |
| Data freshness | Queries run on current data (realtime) | Queries run on 10 minutes old data (a bit delay) | Queries run on 10-20 minutes old data (more delay) |
| Scalability | Since it stores and searching the logs on a live server, its noticeable email server slowdown (using dozens of servers) | Since it uses MySQL, the more data grew, the slower speed it use, and it also cause stability problems (using hundreds of servers) | Since it uses Map-Reduce that enables massive scalability across hundreds or thousands of servers, there is no scalability problems yet. |
| Ad-Hoc Query Ease | Using Regular expression (normal expression, no need of technical skill --no need of programmer) | Using SQL expression (better performance than Regular expression, and still no need of technical skill) | Using Map-Reduce program (better performance than SQL expression, but need programmer with technical skill to develop the query scripts) |

3.3 Why did Rackspace evolve into these three different architecture choices?

In this Rackspace project version 1-3 have the same goal to troubleshoot user problem, so the change of versions is to try out different problems they face during the huge growth in customers, mail servers, and problems.

In the first version, the problems are about the automate (Take too much of engineer scarce time) and the performance (Storing and searching the logs on a live server was negatively affecting the performance of the servers). So, they try using MySQL and MySQL database instead of using direct Email server and log.

In the secord version, the problems are about the performance (The LOADs would get progressively slower as the database grew), the reliablity (No backups, No way to recover the missing logs (after changing log format)), and the scalablity (Not have a good plan for scaling the log system beyond a single monolithic server). So, they change to use something more reliable and scalable on version 3.