Project Proposal

Server Operation cost reduction

Project Overview;

A majority of corporations which provide online server-based services to clients have their profits effected due to the large operation and maintenance costs of their always running server hardware. This project aims to assist such corporations in reducing that cost through the use of a real time tracking algorithm which would suggest time slots throughout the day at which a certain number of servers could be powered off in relation to the expected network traffic at the given time. This algorithm may then automate the performance of the required procedure to shut down the servers and then turn them back on when required.

Objectives;

- Track usage of servers: The algorithm tracks the number of servers handling client requests throughout a day and logs each change in the number against a time stamp. This includes simulating scenarios in accordance to possible real-life events for future use.
- Relate the server usage with time periods: After tracking server usage over a large period of time

(e.g. a month), the algorithm then understands the average number of servers used in a particular time period (calculated by making use of the logged time stamps) on a specific day of the week.

- Relate the server usage with real-world events: The algorithm gives the supervisor the functionality to manually activate a scenario for which the it was already trained (e.g. lock down for a period of time) which would cause the algorithm to manage servers differently as compared to the normal preset.
- Actively manage server operation: The algorithm will manage the number of servers online at a particular time and adjust the number according to its trained understanding (or in accordance to the preset selected by the supervisor). This would make changes more accurate and less time-consuming.

Scope of Work;

Phase 1: Research;

Since this algorithm will obtain traffic data from the server nodes, intensive research would be required into the hardware being commonly used by most corporations which offer such server-based services. This information will then be used to find available hardware that can provide the algorithm a way to manage the server's function state autonomously. This may include physical actuators, or integration within the server's software.

Phase 2: Algorithm Development;

- Learning Phase;
  Considering the project lead's rather disappointing skill set, he will have to understand and be able to make use of various new frameworks and python library functions at each stage of the algorithm development process to be able to build, troubleshoot and test the algorithm.
- Data collection;
  Use Flask (Python framework) to create endpoints that will receive data about server usage at specific timestamps during tracking.
- Data Storage;
  Setup an SQL database which would store the collected data about server usage, and utilize SQLAlchemy (Python library) to enable algorithm to access and store data in the database.
- Data Analysis;
  Give the algorithm the capability to analyse the stored data in the database to calculate average server usage for specific timestamps and then compare these values with real-time readings using pandas (Python library).
- Logic Implementation;
  Enable the algorithm to learn from the analysed traffic data to figure out patterns in server usage across a time span (e.g. a day), and then devise the

maximum number of servers required to be kept online in specific time slots.

Phase 3: Testing (1);

- Testing Platform;
  A small personal server with sufficient hardware would be setup to act as a testing environment for the algorithm.
- Test Data;
  The server will be accessed by a varying number of users throughout a span of 24 hours for a week at most. This would ensure if the algorithm responds to the changes and then devises appropriate recommendations accurately.
- Test Result Analysis;
  The algorithm's outputs would be monitored to understand how it is responding to the data it has been provided with. It is worth taking in consideration that the algorithm has not yet been given the access to the server's power state during the testing and training phases.

Phase 4: Algorithm training;

- Training Platform;
  The same hardware would be made use of as the testing platform.
- Scenario Training;

The algorithm will be put through a series of simulated scenarios where it would be given fabricated information about usage times in accordance to certain events (e.g. lock downs or national holidays). This would make the algorithm change its recommendations to suit the new data and the recommendations would then be stored for later used if such a scenario was to arise in the future.

Phase 5: Testing (2);

- Training Platform;
  Same hardware would be used as in the first testing phase.
- Test data;
  The server will be accessed by a varying number of users throughout a span of 24 hours for a week at most. The usage patterns of the server would be adjusted to match those of the pre-trained scenarios to ensure that the algorithm can actively change its recommendations based on supervisor input.

Phase 6: Integration with hardware;

- Results from prior research into the mostly commonly used server setups will be made use of to devise a method through which the algorithm would be able to control the power state of the physical server hardware. This may be through physical interactions through the use of servos, or through software communication between the server and the platform hosting the algorithm.

Phase 7: Final testing;

- With the algorithm properly connected with the server hardware it was designed to control, a final testing phase will be carried out to ensure that the algorithm is able to carry out it's decisions on time and is actively responding to supervisor inputs in real time.
- This final testing would ensure all hardware and software elements are working as intended and that the algorithm host and server are communicating with no interruptions.

Project Timeline;

Phase 1: Research;

- Expected duration: 5 days
- Expected completion: By the end of week 1

Phase 2: Algorithm Development;

- Expected duration: 4-5 weeks (includes learning phase)
- Expected completion: By the end of week 6

Phase 3: Testing (1);

- Expected duration: 2 weeks
- Expected completion: By the end of week 8

Phase 4: Algorithm Training and Deployment;

- Expected duration: 4 weeks
- Expected completion: By the end of week 12

Phase 5: Testing (2);

- Expected duration: 1 week
- Expected completion: By the end of week 13

Phase 6: Integration with hardware;

- Expected duration: 2 weeks
- Expected completion: By the end of week 15

Phase 7: Final Testing;

- Expected duration: 2 weeks
- Expected completion: By the end of week 18

Total predicted duration: 19 Weeks (4.5 months)

Conclusion;

This project aims to significantly reduce server operation costs for online service providers by implementing a real-time tracking algorithm that optimizes server usage based on actual demand. Through a phased approach encompassing research, algorithm development, and rigorous testing, the system will dynamically adjust server availability, minimizing energy consumption and maintenance expenses. By integrating advanced data analysis and machine learning techniques, the algorithm will learn from historical usage patterns and adapt to real-time conditions, ensuring efficient resource management. Ultimately, this innovative solution not only enhances operational efficiency but also supports sustainability goals, positioning corporations competitively in the market while promoting responsible resource management practices.