Project Plan

Job Allocation Recommender System Using Machine Learning

Danial Zarei

February 2021

Contents

Project assignment

Context

NXP is a large semiconductor firm which among many other solutions and products, they provide license environments and computational resources such as memory, CPU's and storage with the help of cloud technology to many end users around the globe. For this purpose, NXP uses LSF – Load Sharing Facility[1] - Sofware to allocate their resources to their clients. More about this will be explained later in this report.
In a large-scale semiconductor firm like NXP, compute and license environment provide the infrastructure and resources to enable the full cycle of the chip design process. These resources include hardware infrastructure (CPU, storage, memory), software licenses, and etc. The amount of compute resources needed by an end-to-end chip design process is substantially dependent on the design complexity through the entire design cycle and specifically prior to the final stage of it where the design is shipped for manufacturing, also known as “Tapeout”.

This project is being developed by a small team of 3, two interns and one data analyst as a mentor. Furthermore, this project is held under the supervision of HPC (high performance computing) department, and our team is the data science sector as a smaller branch to HPC department.

The project 's goal

The errors in designing ICs (integrated circuits) process can lead to malfunctional products which are significantly costly and time consuming. Therefore, the engineers try to catch these errors as soon as possible in their projects with the help of software tools for EDA (electronic design automation). NXP-semiconductor is trying to effectively utilize their resources and optimally allocate these resources to different jobs. This can significantly reduce the costs for NXP's end users and reduce the job pending times.

During the ICs design process, the end-users (engineers) will request HPC resources to run their jobs on LSF. This can result in both over-requesting and under-requesting resources. In case of over-requesting resources, many jobs can be put into a pending state since the resources are finite and have a shared pool. This will prevent NXP to fully benefit from their resources. Pending state can be extremely costly for the clients and slow down their development process in the competitive market. On the other hand, under-requesting can be even more problematic. It may lead to forceful exit of the jobs due to lack of required resources.
Therefore, NXP is looking for a software solution to effectively and optimally allocate resources to ensure that all jobs are receiving enough resources required for proper execution. Our team is trying to use the collected data in past and in real time to build a machine learning model which can predict an optimal resource allocation amount for the client as a suggestion base on their project phase, experience and many other factors. This will have a significant impact on setting the cost of the end-user which helps improving company’s position in today’s competitive market.

---

[1] The IBM Spectrum LSF ("LSF", short for load sharing facility) software is industry-leading enterprise-class software which balances load and allocates resources, and provides access to those resources.

Expected results

I expect to not only get acquainted with several data sources and software tools to do big data and machine learning, but also contribute to the optimization of HPC (High Processing Computing) and EDA (electronic design automation) resource utilization in NXP. The insights gained during the various phases of this assignment will be used to configure and automatically optimize the LSF job scheduling. Furthermore, I expect our team to contribute to the development of the software, which is needed to close the loop, e.g.: from big data analysis to LSF job scheduling through machine learning back to LSF. Working together with the R&D IT teams HPC and DA will help me to contribute to a working HPC big data system and enterprise scale.

Our product will be a solution using machine learning models to help the engineers to reserve their needed resources and minimize problems such as over-under estimations. We are expecting to reduce the queue time for each job by optimizing the recourses in our pool.
Since at the moment, there is no solution to these problems, NXP would appreciate any automated solution which can help them and their clients to take the most benefits as possible by optimization.

At the moment, there is no hard requirement for our models (accuracy for instance), but we are aiming for a solution which can help this problem in any degree. Looking at competitors, such as Qualcome, there have built similar solutions with accuracy of 90% although this is not at their production level.

Strategy

Our team is working with scrum framework which is a lightweight framework that helps people, teams and organizations to generate value through adaptive solutions for complex problems. Our scrum team consists of a Scrum Master, a Product Owner, and three Developers. Our team is a relatively small team which makes communication easy and increases the overall productivity.
We are going to work in Sprints which are fixed length events of one month or less to create consistency. These sprints consist of, sprints planning, daily and weekly scrums, sprint reviews and sprint retrospective. In our planning phase, these questions will be considered:

1. **Why** this sprint is valuable?
2. **What** can be Done this Sprint?
3. **How** will the chosen work get done?

Research questions

The main research question to this project would be, how can we optimize our resources allocations and reduce over/under resources estimation from our clients? How can we tackle the queuing issue? (When there are not enough resources in the pool, jobs will be in queue phase although there are a lot of resources reserved and unused due to over estimation by other clients).

Here there few sub questions which can help us to break the problem into smaller pieces.

1. What are the existing solutions to our problem in the market? - **Available product analysis/ Competitive analysis**
2. Where can we find general information, guidance and best practices to tackle our problem? - **Literature study**
3. What are the Strengths, Weaknesses, Opportunities and Threats related to our project? **-SWOT analysis**
4. How are the users using our solutions and what are their requirements? - **Explore user requirements**
5. How can we study and analyse our data? What are the best tools for this purpose? **-Data analytics**
6. How can we detect problems and bugs in our solution before going live? **-Usability testing**
7. Which similar products can we compare our product to? **-Benchmark test**
8. Does our product conflict with certain norms and values? **-Ethical check**
9. How can we review our product before releasing to our clients and users? **-Product review**

Activities and time plan

Phases of the project

This project consists of 4 main phases (motioned below) which are the scope of the project. Each phase lasts from 4 to 10 weeks (total of 26 weeks). Here you can see an overview of each week's assignments and milestones.

Exploratory phase:

The first phase of the assignment is an exploratory phase to get familiar with the data, the IT systems and software tools (Splunk, AWS athena and AWS Sagemaker) and the problem domain. After the initial introductions, our will work together with experts from DE and HPC to have a closer look into the available data sources (RTM, PA, FNM). The DA specialists will assist our team in this stage to understand the structure of the data and the software tools (Splunk, AWS athena). Together with the domain experts from DE and HPC we will select an EDA tool to focus on in the next phase, namely: profiling. During this phase, we expect to answer some questions from our research questions sections.

These are questions number 1,2,4 and 5.

Profiling phase:

The profiling phase is aiming to profile specific EDA tools in combination with HW designers (or another relevant factor in consultation with the DE / HPC domain experts) such that it is clear from the analysis how much HPC resources the selected EDA tool requires. Once this analysis is complete and confirmed with the domain experts, the acquired insights will be used in the next phase, namely: prediction.

Prediction phase:

The prediction phase is used to further use the findings of the previous analysis by actively predicting CPU, memory and EDA license usage for the selected EDA tools. After this phase is complete, we will help the HPC team to implement the next phase, which is optimization and learning.

Optimization phase:

During the optimization phase the insights from big data will be applied in HPC jobs, i.e.: every job that starts with the selected EDA tool will automatically request the amount of CPU and memory based on big data. The assumption is that the duration of HPC LSF jobs will not be adversely affected, while the utilization of CPU and memory is optimized by reducing the amount of waste (idle CPU and unused memory claims). The learning phase involved active monitoring of the LSF job duration to ensure that our assumption is indeed correct.

During this phase, we will try to answer some of our research questions such as number 6,7,8 and 9 as listed in the previous section, 'Research questions'.

To verify our solution, in the testing phase, we'll try to inject existing data from past and let the model try to predict the required resources for each job. Since in the older time frame data, we know how each job ended and how much resources were required, we can compare our model's prediction to produce an accuracy. If the accuracy is acceptable (at the moment, we are not having a clear desired accuracy, anything above 50% would be a good start) then we can try to use the model for future submitted jobs and monitor its behaviour. Hopefully, the models will be improved over iterations until we hit a plateau.

Possible extensions: **(outside of the project's scope)**

The second half of the assignment (e.g., starting from week 14 onwards) will be dedicated to an extension based upon big data insights from the exploratory phase and the consultation of the domain experts (DE, HPC, and DA). This extension will be done - if time permits - in parallel to the optimization and learning phase. Possible extensions can either focus on areas to further optimize CPU (e.g. address idle slot consumption) or at more efficient job scheduling based on predicted EDA license utilization.

Time plan and milestones

Phasing		Start date	Finish date
1 Exploratory phase		Week 1	Week 4
2 Profiling phase		Week 5	Week 10
3 Prediction phase		Week 11	Week 16
4 Optimization phase		Week 17	Week 26

Project Plan

During the project the phases of the data science methodology as created by IBM analytics (IBM, 2015) will be used. This will be done in three iterations. For each iteration the required resources, deliverables and dependencies will be listed. The IBM methodology is an effective way of working which includes iterations process that follows as a sequence. This can help us to have an effective structure during the project. Our project is a continuous cycle. Which means the models get trained, evaluated, tested and reviewed by the project manager. Then we use the feedback received and eventual new data sets to proceed further and update the models for a better performance. During our 4 phases as mentioned before, we will proceed with this structure as shown in the figure 1 below.

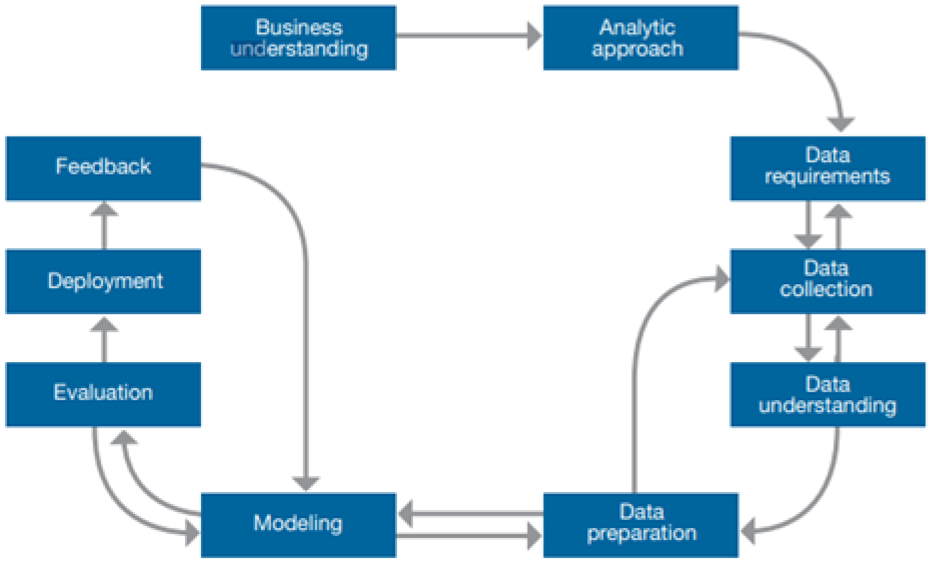
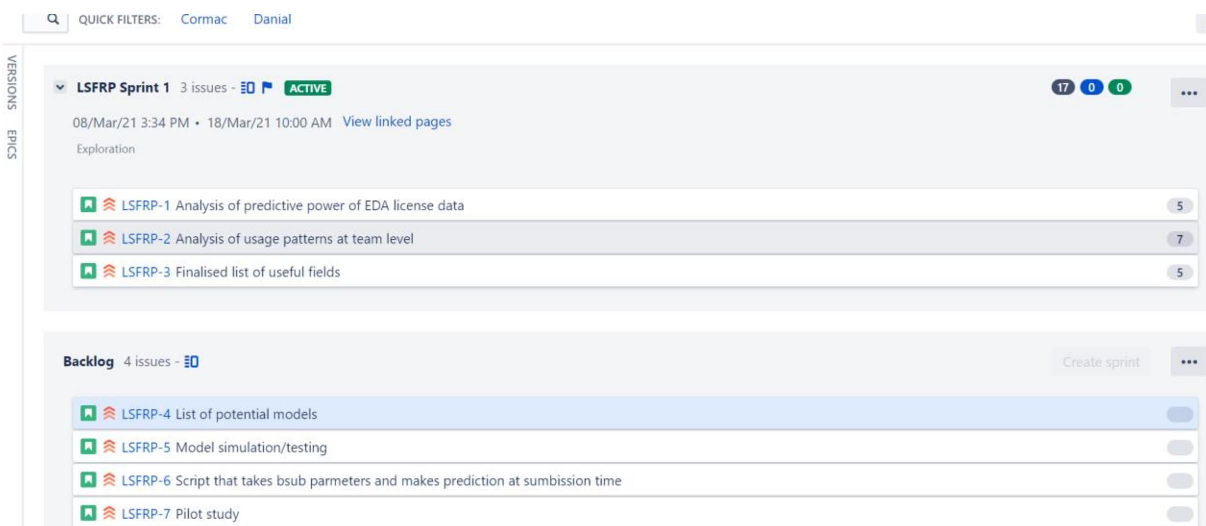


Figure 1: IBM Data Science Methodology

Planning

Our team is working with sprints of 2-3 weeks. Here you can see the sprints created so far and keep in mind that these sprints can be added, removed or modified as we proceed in the project. Each backlog contains of multiple smaller tasks, assigned to each member. This board will be shown weekly to the HPC team in the morning stand-ups where we can share and give feedbacks to other projects.



Risk and mitigation

To show the probability of the risks and how much of a setback it would be if these risks would occur we have given the risks a score. The Calculating Risk Score from http://intaver.com/risk-scores/ has been used.

Label	Score	Probability	Impact
Very low	1	1 in 100	1 day
Low	2	1 in 10	< 1 week
Medium	3	1 in 5	2 weeks
High	4	1 in 2	1 month
Very high	5	>1 in 2	> 1 month

The following formula is used to calculate the risk score:
The risk score = probability score * impact score

1. **Lack of model variability**

This risk is also known as over-fitting can occur in machine learning projects. This can happen when a model is built using a particular set of data. The inputs are tweaked to give the absolute best output without regards to variability of data (e.g., new data is never introduced). When model is applied to new, real world data, it doesn't perform anywhere near as well as it did on the old tested data.
This over-optimization can be managed with various performance measures and using a method called walk-forward optimization to try to get as much data in as many different timeframes as possible into the model. Our team needs to make sure the data we are feeding our machine learning models are varied across both data types, timeframes, demo-graphical data-sets and as many other forms of variability that we can find.
The probability of this happening is medium also the impact of this happening would be medium, therefore the risk score is: 3(Medium) * 3 (Medium) = 9

2. **Data**

There are plenty of issues that can be introduced via data. With data, we can have many different risks including Data quality (e.g. bad data), not enough data, Homogeneous data (not necessarily lack of the amount of data but the lack of variability of the data) and etc.

To prevent this issue, our team needs to find a couple of different data sets with many different types of demo-graphical data points and then spend time doing some feature engineering to find the best model inputs for accurate outputs.

The probability of this happening is medium also the impact of this happening would be medium, therefore the risk score is: 3(Medium) * 3 (Medium) = 9

3. Low correlation

There is an assumption that there is correlation between different columns in our data set and the amount of resources (CPU's memories and etc.) being used by our clients. In case this assumption is false, or the correlation is simply too small to make conclusions, this can be problematic in choosing our inputs during training the models.

To prevent this, in our exploratory phase, we need to use techniques such as correlation matrix using heatmaps or packages from matplotlib and seaborn.

The probability of this happening is low but the impact of this would be high,

Therefore the risk score is 2 (Low) * 4 (High) = 8.

4. Poor productivity

It is important to remember that humans are not machines, therefore it's not realistic to expect them to be productive every hour they spend on their work. To avoid a project group falling behind the planned timeframes, we need to constantly examine the productivity of our development team.

But how do we measure productivity?

To determine the productivity level of the team, we can use tools such as burn-down charts, Trello and keep up with the iteration planning.

Another solution would be to set achievable timeframes and a sustainable pace in our project's expectations to avoid burn-out of members.

The probability of this happening is low to medium. Due to pandemic and lockdowns, our team is working from home and this might have impact on our productivity. This impact could also vary from low to high depending on how serious we track our progress.

Using our calculation method the risk score is: 2(Low) * 3 (medium) = 6.