Baseline

Wei Fan

Brief review

Project

FECBench is an open source framework, which guides providers in building performance interference prediction models for their services without incurring undue costs and efforts.
FECBench comprises an offline stage with a set of steps to create a knowledge base followed by an online stage. Developers can use the same offline stage process to further refine this knowledge base [1].

Dataset

The dataset includes 110 benchmarks' execution status and system resource pressure, used for the construction of design space for FECBench, provides data for data view and clustering visualization. I did data preprocessing such as removing empty attributes, dealing with missing values, and made a distinct benchmark execution status subset. There are 18 effective system metrics, which are regarded as attributes in the dataset, including resource pressure, like clockcycle per instruction, memory bandwidth of L2 cache, disk i/o etc.

Project task

For this baseline, my task is to do visualizations for benchmark warehouse and system clusters using this dataset. The benchmark warehouse is important for constructing the design space of system resource stress, so doing visualizations on it is necessary for profiling the offline steps in the FECBench model.

Research paper

In order to do visualization on resource stressor, which uses multiple dimensions (>3) to represent, I need to use PCA to reduce dimension. Therefore, I want to apply iPCA to visually analyze system metric data and perform interactive actions. [Paper: iPCA: An Interactive System for PCA-based Visual Analytics]
As we all know, Principle Component Analysis is a method that projects a dataset to a new coordinate system by determining the eigenvectors and eigenvalues of a matrix and can find the factors which explain the most variation among data points. The baseline paper, iPCA, visualizes the results of principle component analysis using multiple coordinated views and a rich set of user interactions, to help user understand the data space and relationships among data items.
There are four views provided in iPCA: Data View , Eigenvector View , Projection View and Correlation View.
Project view, uses Principle Component Analysis to get two principal components and uses them to project data points onto a two-dimensional coordinate system. iPCA also provides interface for user to assign components to make projections.
The Data View is located below the Projection View, and shows a parallel coordinates visualization of all data points in the original data dimensions.
In the Eigenvector View, data points are shown in the eigenspace. The calculated eigenvectors and their eigenvalues are displayed in a vertically projected parallel coordinates visualization, with eigenvectors ranked from top to bottom by dominance.

The correlation view gives Pearson-correlation coefficients and relationships between variables are represented as a matrix of scatter plots and values.

iPCA provides bunch of interactive tools and is proved to be with effectiveness by performing a comparative user study [2].

My baseline

Objective

Like what iPCA did, I want to do visualizations on projection view, correlation view, and dataview to give a clear insight on benchmark warehouse.

Encodings

First, I did principle components analysis on the dataset and get the top two components: host_context_switch and host_disk_weighted_io_time. Then I use the tramsformed result to do the 2 dimensional projection. I also did k-means clustering and find 13 is the best cluster number.

The projection view contains 3 subplots: one scatter plot and two histogram plots. The scatter plot is the result of PCA. The color represents which cluster it belongs to because I did clustering and get 13 clusters.

I also did lasso for brushing. The different opacity which is the result of selection. When you double click, it will recover. So the opacity in scatter plot represents whether this data item is selected.

The two histogram plots indicates the distribution of brushing results. For example, if you select several data points using lasso, the left histogram will calculate the minimum and maximum points' value in xaxis, here is the host context switch attribute you select and calculate the data point number at this xaxis value in your selection. And the right histogram will do the y part at the same time.

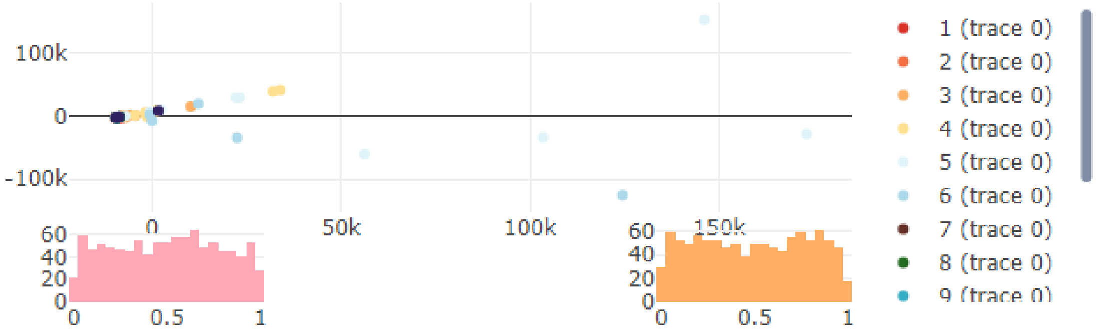


Fig. Projection view

I use parallel coordinates to give a data view. Each dimension represents the dimensions in the original data, and each line represents each data item.

Color of each line also represents cluster number and opacity represent if the data item is selected by user.

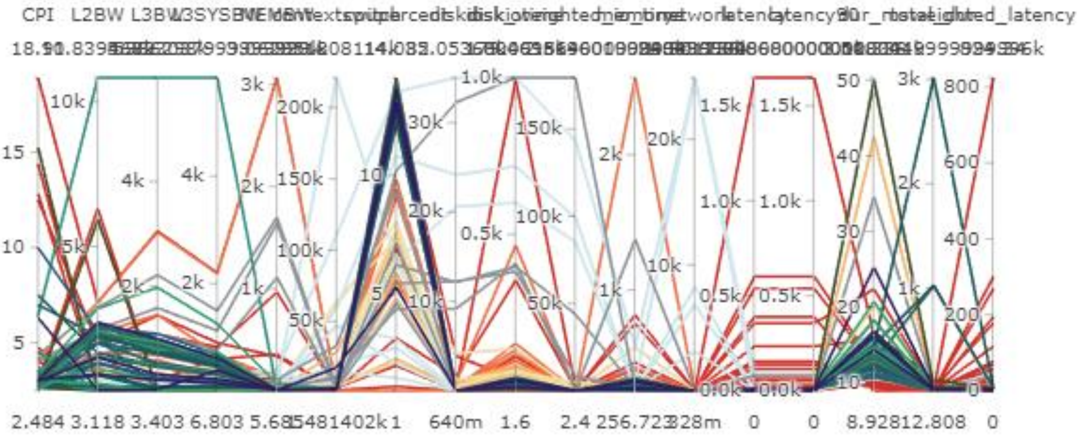


Fig. Data view

I use min-max function to plot the normalization view. The encoding is the same as data view.

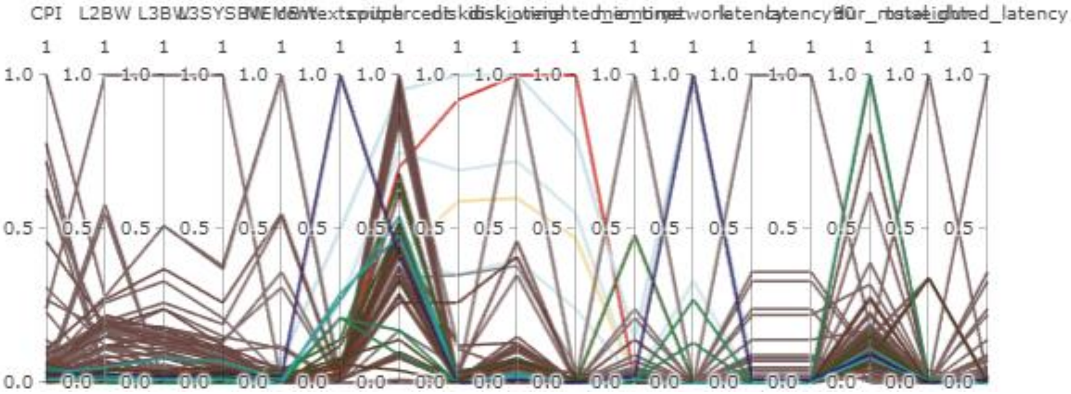
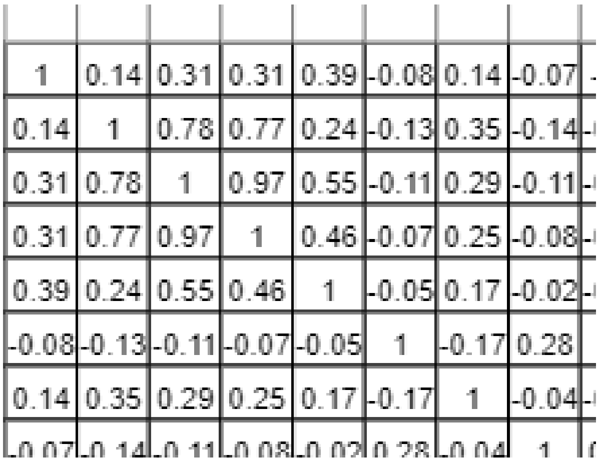


Fig. Normalization view

I did Pearson-correlation coefficients and make user understand the relation of dimensions.



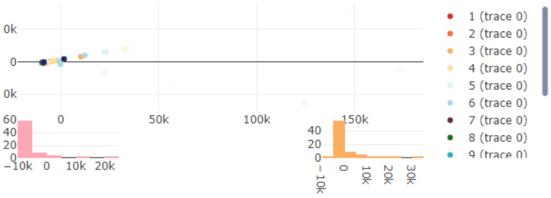
1	0.14	0.31	0.31	0.39	-0.08	0.14	-0.07
0.14	1	0.78	0.77	0.24	-0.13	0.35	-0.14
0.31	0.78	1	0.97	0.55	-0.11	0.29	-0.11
0.31	0.77	0.97	1	0.46	-0.07	0.25	-0.08
0.39	0.24	0.55	0.46	1	-0.05	0.17	-0.02
-0.08	-0.13	-0.11	-0.07	-0.05	1	-0.17	0.28
0.14	0.35	0.29	0.25	0.17	-0.17	1	-0.04
-0.07	-0.14	-0.11	-0.08	-0.02	0.28	-0.04	1

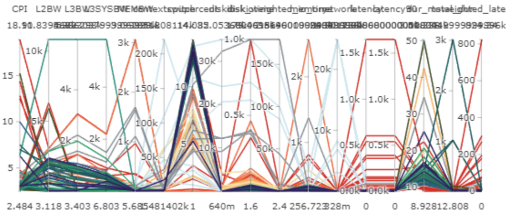
Fig. Correlation view

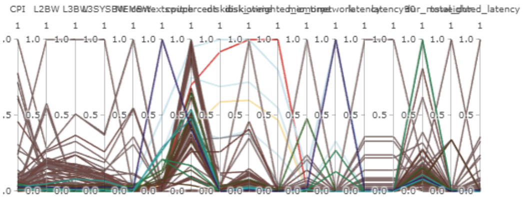
Interaction

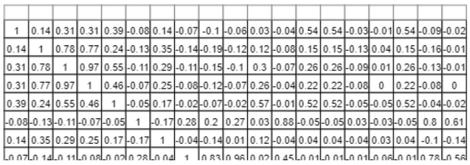
Using brushing, linking views, zooming which can be found in ‘Encodings’ section.

Overall view









1	0.14	0.31	0.31	0.39	-0.08	0.14	-0.07	-0.1	-0.06	0.03	-0.04	0.54	0.54	-0.03	-0.01	0.54	-0.09	-0.02
0.14	1	0.78	0.77	0.24	-0.13	0.35	-0.14	-0.19	-0.12	0.12	-0.08	0.15	0.15	-0.13	0.04	0.15	-0.16	-0.01
0.31	0.78	1	0.97	0.55	-0.11	0.29	-0.11	-0.15	-0.1	0.3	-0.07	0.26	0.26	-0.09	0.01	0.26	-0.13	-0.01
0.31	0.77	0.97	1	0.46	-0.07	0.25	-0.08	-0.12	-0.07	0.26	-0.04	0.22	0.22	-0.08	0	0.22	-0.08	0
0.39	0.24	0.55	0.46	1	-0.05	0.17	-0.02	-0.07	-0.02	0.57	-0.01	0.52	0.52	-0.05	-0.05	0.52	-0.04	-0.02
-0.08	-0.13	-0.11	-0.07	-0.05	1	-0.17	0.28	0.2	0.27	0.03	0.88	-0.05	-0.05	0.03	-0.03	-0.05	0.8	0.61
0.14	0.35	0.29	0.25	0.17	-0.17	1	-0.04	-0.14	0.01	0.12	-0.04	0.04	0.04	-0.04	0.03	0.04	-0.1	-0.14
-0.07	-0.14	-0.11	-0.08	-0.02	0.28	-0.04	1	0.83	0.96	0.02	0.45	-0.01	-0.01	-0.01	-0.06	-0.01	0.78	-0.56

References

[1] FECBench: A Holistic Interference-aware Approach for Application Performance Modeling

[2] iPCA: An Interactive System for PCA-based Visual Analytics, Dong Hyun Jeong, Caroline Ziemkiewicz