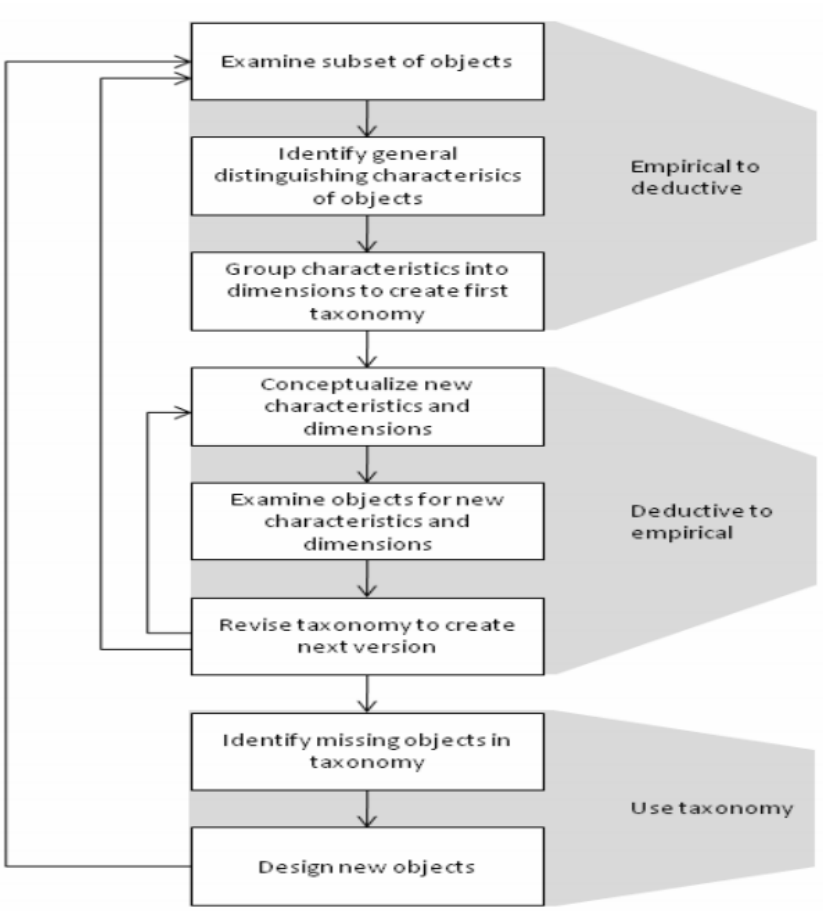
Surrogates

-Researched taxonomy development:
In order to classify surrogates, I investigated some taxonomy development methodologies [1-3] and decided to use Bailey's three-level indictor model and combine both the empirical to deductive approach and the deductive to empirical approach [2]. The figure below shows a summary of this method. Taxonomy development is an iterative active process. This iteration covers the empirical to deductive approach.



I examined a collection of surrogate papers [4-13] and identifies the following characteristics:

- *Machine learning model:* most work to date in building surrogates uses random forests[4], Gaussian Processes[5], Neural Networks, or other machine learning models [6,7]

- *Dimentionality of the problem:* as the dimensionality grows, the complexity (or computational cost) grows exponentially.

- *Data set generation and labeling:* in supervised (sequential) learning, the labeled data set is given beforehand and learning is performed afterwards[4]. In concurrent machine learning, the data set is generated and labeled as model training proceeds[8]. In active learning, we are given the unlabeled data, and we decide which ones should be labeled during the training process[9].

- *Variability of the architecture:* Most surrogates used in scientific workflow use a fixed architecture [4-7]. However, a recent paper [14] successfully used NAS

(Neural Architecture Search) in order to infer the architecture of the neural surrogate.

- *Data selection method:* Most surrogates models require human intervention for the selection and adjustment of training data [4-7]. However, an active approach to surrogates learning that enables automatic training data selection has emerged in GP surrogates[15].

- *Algorithmic area:* The scientific problem the surrogate is investigating. For example: partial differential equations, particle dynamics.

- *Link between ML and HPC:* HPCforML uses HPC to execute and enhance ML performance, or uses HPC simulations to train ML algorithms, which are then used to understand experimental data or simulations. MLforHPC uses ML to enhance HPC applications and systems [10].

The next step is to group the noted characteristics into dimensions to create the first iteration of the taxonomy:

- *Algorithmic dimension:* concerned with the machine learning model used and the resulting architecture and its properties.

- *Data dimension:* is a sub-dimension of the algorithmic dimension. Concerned with the generation, labeling and selection of the data used to train the ML model.

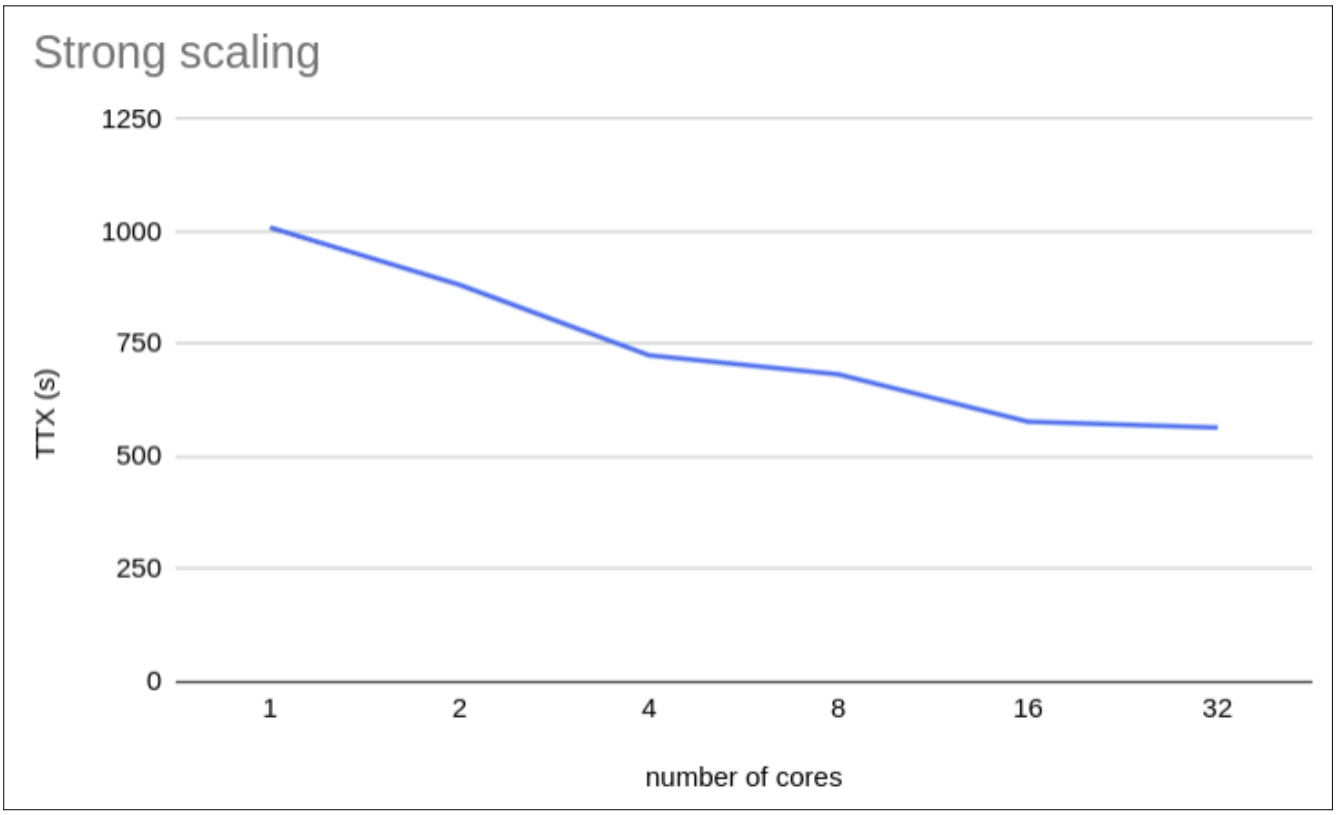
- *High Performance Computing dimension:* concerned with the link between ML and HPC.

- *Problem Dimension:* concerned with the algorithmic area and the dimentionality of the problems.

RCT

- Performed an experiment to understand strong and weak scaling properties of EnTK

A- Strong Scaling: To investigate strong scaling, I run 6 applications, each consists of 1 pipeline containing 1 stage. Each stage contain 32 tasks that sleeps for 100s. The number of stages and tasks is constant while the number of cores increases for every application (1,2,4,8,16,32). The figure below shows the result



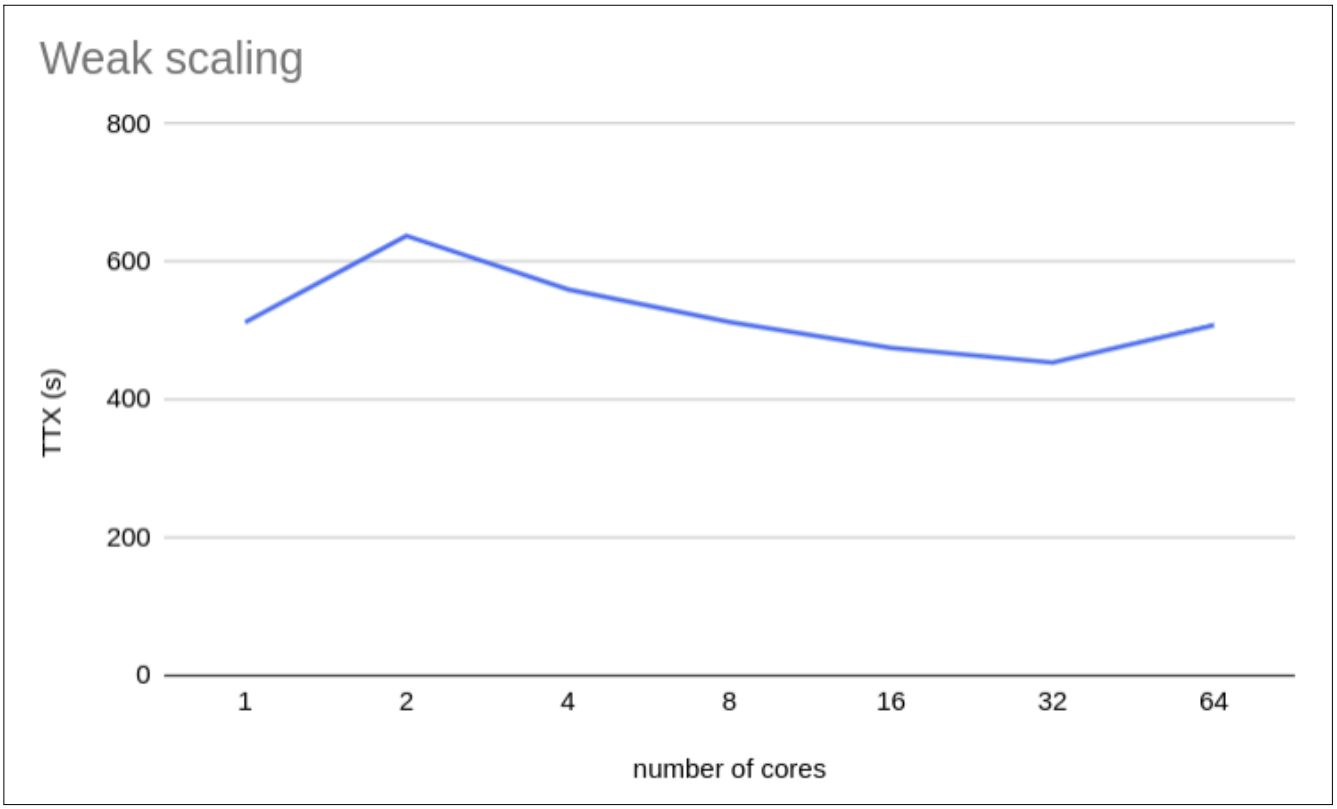
As expected, the time needed to execute the application decreases with the increase in the number of cores. This is because there are more available resources for the fixed number of tasks. I expected the decrease to be linear which is not the case. This could be due to variations in the overhead from one application to another. At 16-32 cores, the decrease in TTX is minimal. This is because only 24 cores are available.

```
(ve-rct) esraa@js-17-185:~$ lscpu
Architecture:            x86_64
CPU op-mode(s):          32-bit, 64-bit
Byte Order:              Little Endian
CPU(s):                  24
On-line CPU(s) list:     0-23
Thread(s) per core:      1
Core(s) per socket:      1
Socket(s):               24
```

B- Weak Scaling: To investigate weak scaling, my experiment consists of 7 applications, each consists of 1 pipeline containing containing 1 stage. Each task sleeps for 100s The table below shows the number of tasks and cores used for every application.

Number of cores	Number of tasks
1	4
2	8
4	16
8	32
16	64
32	128
64	256

The figure below shows the results of this experiment



For the weak scaling experiment, I expected TTX to increase non-linearly and gradually because EnTK does not have ideal weak scaling due to delays in the Executor module of the RTS Agent. However, the TTX initially increases, then decreases, then increases again. To understand this, I consider the number of "batches" (of tasks) running in series for every application. For every application, there are 4 batches running in series, each containing a varying number of tasks. Therefore, the TTX of each application should be almost the same with a slight increase when needing to engage more cores. Perhaps the overhead from EnTK is high enough that its fluctuation masked the expected increase.

Resources:

[1] D. E. Taylor. Survey and Taxonomy of Packet Classification Techniques. Technical Report WUCSE-2004-24, Washington Univ., St. Louis, May 2004.
[2] T. M. Mengistu and D. Che, "Survey and taxonomy of volunteer computing", ACM Comput. Surveys, vol. 52, no. 3, pp. 1-35, Jul. 2019..
[3] Nickerson, R., Muntermann, J., Varshney, U., & Isaac, H. (2009). Taxonomy Development in Information Systems: Developing a Taxonomy of Mobile Applications.
[4] JL Peterson, KD Humbird, JE Field, ST Brandon, SH Langer, RC Nora, BK Spears, and PT Springer. Zonal flow generation in inertial confinement fusion implosions. Physics of Plasmas, 24(3):032702, 2017.
[5] Juliana Kwan, Katrin Heitmann, Salman Habib, Nikhil Padmanabhan, Earl Lawrence, Hal Finkel, Nicholas Frontiere, and Adrian Pope. Cosmic emulation: fast predictions for the galaxy power spectrum. The Astrophysical Journal, 810(1):35, 2015.
[6] Felix Brockherde, Leslie Vogt, Li Li, Mark E Tuckerman, Kieron Burke, and Klaus-Robert M¨uller. Bypassing the kohn-sham equations with machine learning. Nature communications, 8(1):872, 2017.
[7] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert M¨uller, and O Anatole Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. Physical review letters, 108(5):058301, 2012
[8] Linfeng Zhang, Han Wang, and Weinan E. Reinforced dynamics for enhanced sampling in large atomic and molecular systems. The Journal of Chemical Physics, 148(12):124113, 2018.
[9] Linfeng Zhang, De-Ye Lin, Han Wang, Roberto Car, and Weinan E. Active learning of uniformly accurate interatomic potentials for materials simulation. Physical Review Materials, 3(2):023804, 2019.
[10] Geoffrey Fox, James A. Glazier, JCS Kadupitiya, Vikram Jadhao, Minje Kim, Judy Qiu, James P. Sluka, Endre Somogyi, Madhav Marathe, Abhijin Adiga, Jiangzhuo Chen, Oliver Beckstein, and Shantenu Jha, "Learning Everywhere: Pervasive Machine Learning for Effective HighPerformance Computation," presented at the HPDC Workshop at IPDPS, Rio de Janeiro [Online]. Available: https://arxiv.org/abs/1902.10810, http://dsc.soic.indiana.edu/publications/Learning_Everywhere_Summar y.pdf
[11] Geoffrey Fox, James A. Glazier, JCS Kadupitiya, Vikram Jadhao, Minje Kim, Judy Qiu, James P. Sluka, Endre Somogyi, Madhav Marathe, Abhijin Adiga, Jiangzhuo Chen, Oliver Beckstein, and Shantenu Jha, "Learning Everywhere: Pervasive Machine Learning for Effective HighPerformance Computation:

Application Background,” Feb. 2019 [Online]. Available: http://dsc.soic.indiana.edu/publications/Learning_Everywhere.pdf
[12] Geoffrey Fox, Shantenu Jha, “Understanding ML driven HPC: Applications and Infrastructure,” in IEEE eScience 2019 Conference, San Diego, California [Online]. Available: https://escience2019.sdsc.edu/
[13] Integrating Machine Learning with Physics-Based Modeling
[14] MF Kasim, D Watson-Parris, L Deaconu, S Oliver, P Hatfield, DH Froula, G Gregori, M Jarvis, S Khatiwala, J Korenaga, et al. Up to two billion times acceleration of scientific simulations with deep neural architecture search. arXiv preprint arXiv:2001.08055, 2020
[15] Francisco Sahli Costabalab, Paris Perdikaris, Ellen Kuhle, and Daniel E. Hurtado. Multi-fidelity classification using gaussian processes: accelerating the prediction of large-scale computational models. Computer Methods in Applied Mechanics and Engineering, 357:112602, 2019