Using Machine Learning to Guide Architecture Simulation

Paper Discussion by Bhaskar Gautam

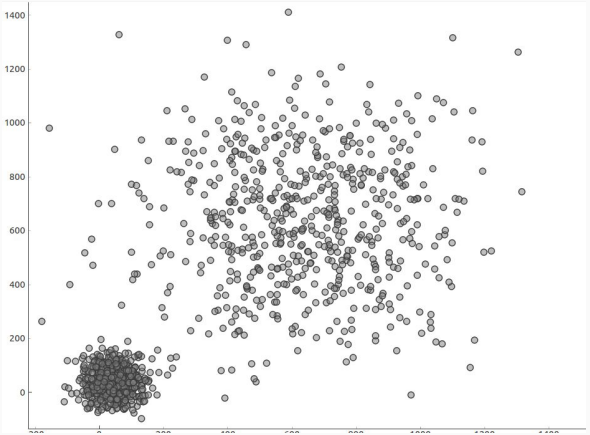
Problem Addressed in Paper

- On fastest Simulators, if we want to simulate full execution of single benchmark to determine cycle level behaviour of a processor then it can take week or months to complete the entire simulation.
- This creates a serious problem to reduce this machine months without introducing an unacceptable error or excessive simulator complexity

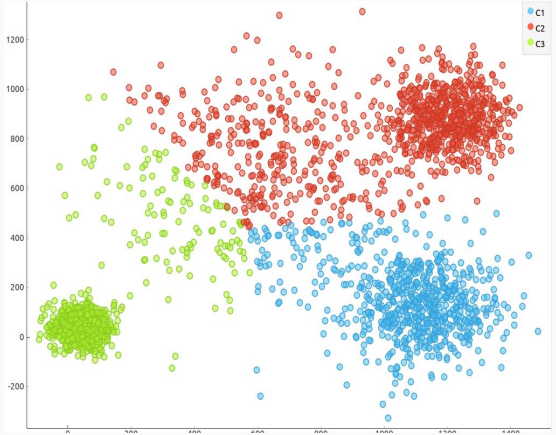
SimPoint

- SimPoint chooses a very small set of samples from an executed program termed as “Simulation Points”.
- When simulated and executed appropriately, it provides an accurate picture of complete execution of program.
- Simulation in details only these chosen “Simulation Points” can save hours of simulation time.

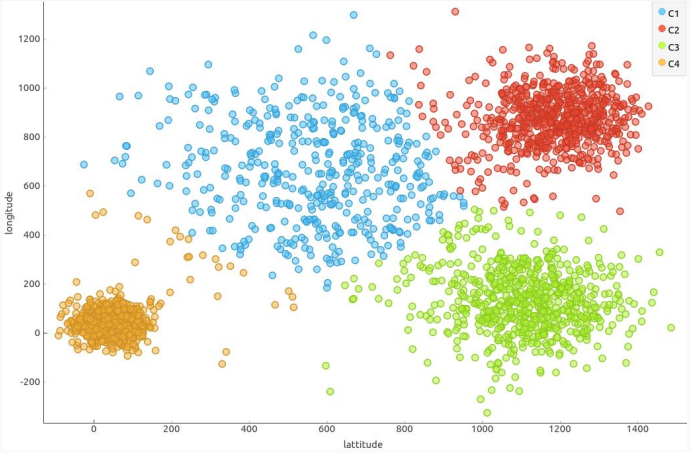
Overview of k-means Clustering Algorithm



Simple Plot



K is 3



K is 4

Related Terms

- Interval
  - Break a program's execution into non-overlapping intervals (In paper 100M instruction)
- Similarity Metric
  - It measures the similarity in behaviour between two intervals of a program execution
- Phase(Cluster)
  - A set of distinct intervals within a program execution that all have similar behaviour, regardless of temporal adjacency.
  - A well formed phase should have intervals with similar behavior across various a architecture metrics (eg: CPI, Cache misses, Branch Prediction).
- Phase Classification
  - Using k-means algo to group intervals into phases with similar behaviour.

Frequency Vector

- A basic block is a single-entry, single-exit section of code with no internal control flow.
- Basic Block Vector (BBV) are one of the type, which represent basic block etc.
- Use Case
  - As a signature for each interval of execution such that each vector tells us what portions of code are executed, and how frequently those portions of code are executed.
  - As a comparison between BBV's of two intervals.
- If two intervals have similar BBVs, then the two intervals spend about the same amount of time, and hence the performance of those two intervals to be similar.

Basic Block Vector Algorithm

- For Each interval, i {
  BBV[] = {0} // Each element of an array points to each Basic Block of program
  for each basic block, j {
    - Count <- Frequency(i) // Calculates no. of times each block in program entered
    - Count <- Count * Number_Instruction_in_Block(i)
    - BBV[i] <- Count

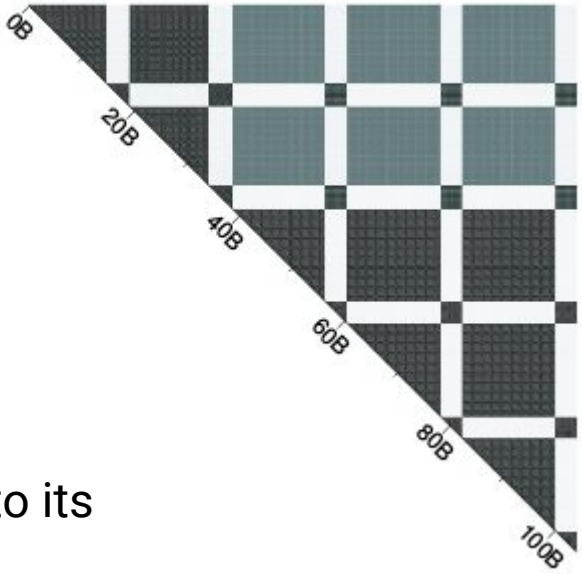
  }

  Divide the each element of BBV by the sum of all the elements in the BBV.

  }
- We measure the similarity of two BBV using Euclidean distance

Basic Block Similarity Matrix

- It is an Upper Triangular nxn matrix, to relate all intervals
- An entry at (x,y) in the matrix represents the Manhattan distance (d) two intervals x & y
- The Diagonal of matrix represents the program exec. from start to completion
- The Darker the point more similar the intervals (d ~ 0) & Lighter means more different (d ~ 2)
- An interval (represented by triangle) depicts it is similar to its neighbor



Automatically Finding Phase Behaviour

- SimPoint automatically extract phase information from programs.
- It breaks the complete execution of program into phases that have similar **Frequency Vectors** using “ Unsupervised **k-means** Data Clustering **Algorithm** “

SimPoint Phase Clustering Algorithm

- Profile the program by dividing the execution into fixed length contiguous intervals.
- For each interval, i

  {

  FV = Frequency(i) // Collects frequency vector for interval i

  FV = Normalized(FV) //Sum of all Normalized value equal to 1

  FV = Random_Linear_Projection(FV) // Reduce the dimensions

  }
- k_mean(FV's) // Run the k-means algo upto max phase detected
- BIC(Clusters, k) // Compare different cluster formed using k
- Choose the clustering with a small k s.t its BIC Score(~ Best Observed)

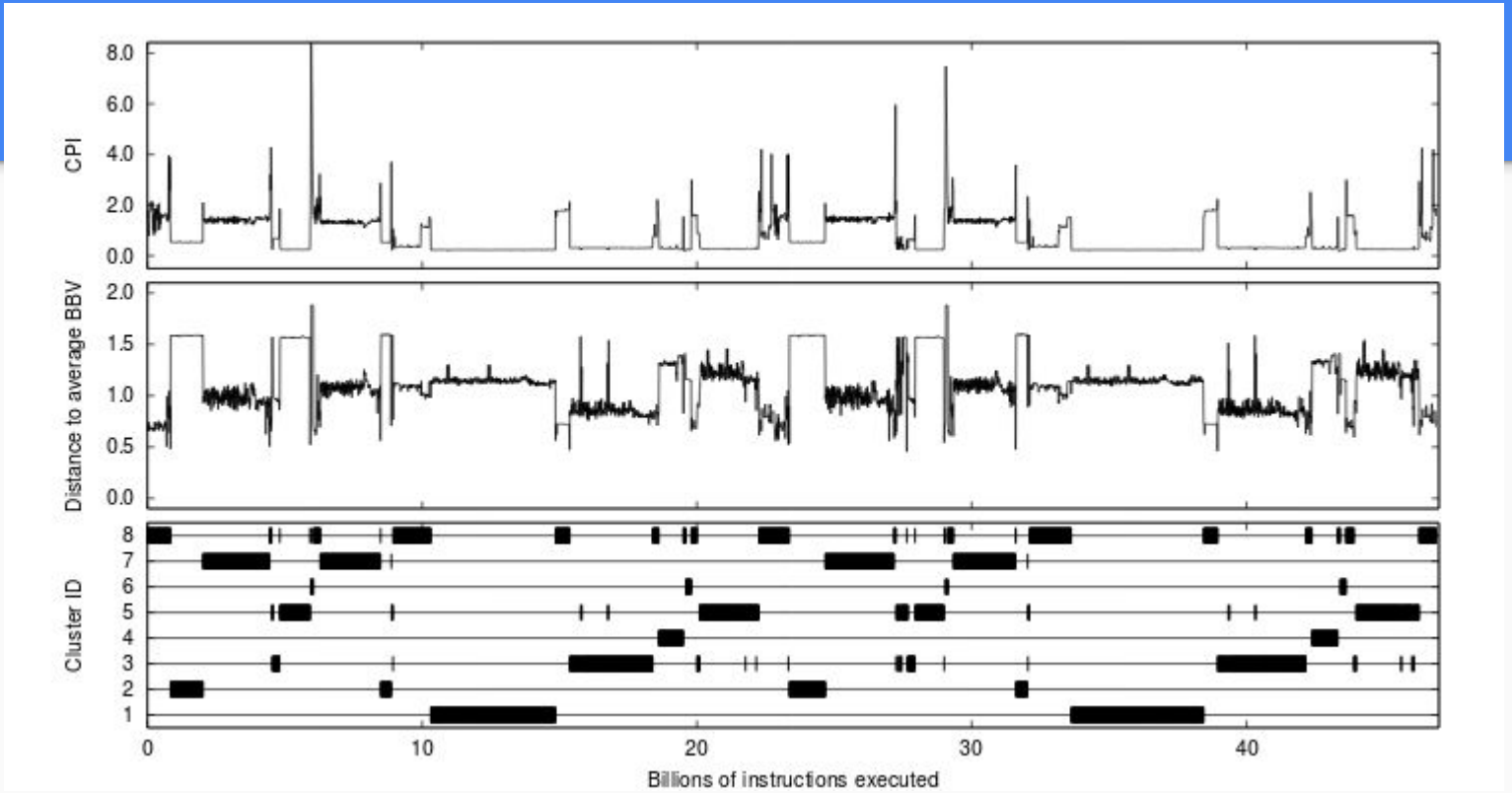
Bayesian Information Criterion

- Gives score of the how well a clustering represents the data it clustered.
- **BIC** directly proportional to **k** inversely proportional to **Accuracy**
- **SimPoint** default threshold **90%**

Tuning of Parameters

1. **Reducing Projected Dimensions**
    a. Selection *// Removes Unusual Dimensions*
    b. Reduction *// Create new lower_dimension space and projecting each into new space*
2. **Bayesian Information Criterion**
3. **Interval Length**
    a. When very small interval length(say < 1M) creates million of intervals to cluster
4. **Number of Cluster**

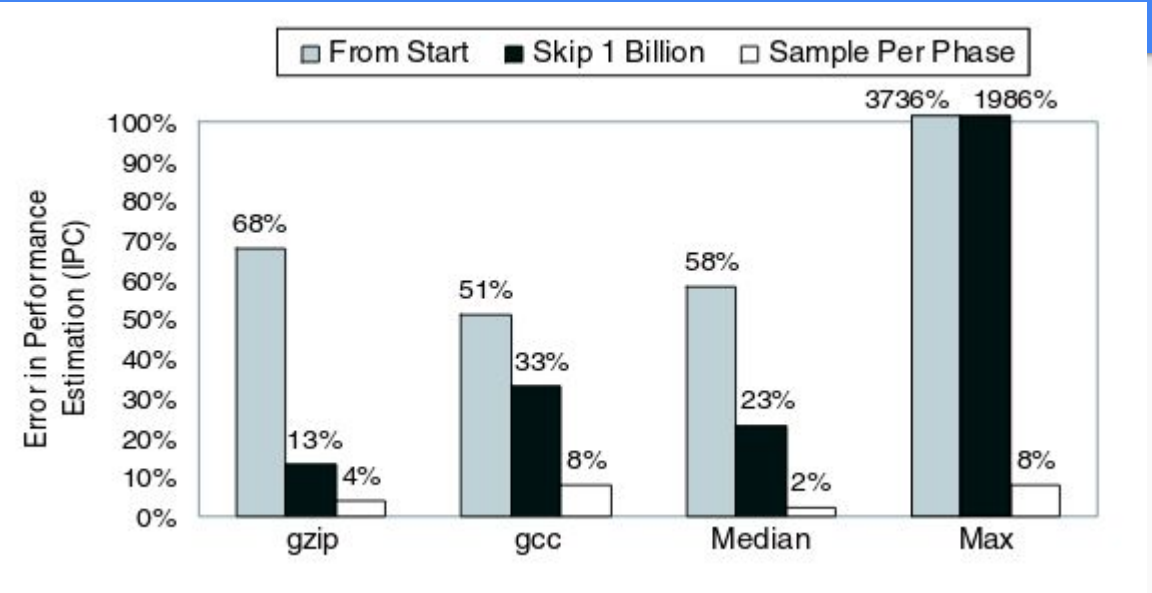
Phase Clustering of gcc



Selection of SimPoints

- The centroid is the average of all the intervals in the cluster.
- From each Cluster, SimPoint picks the interval that is closest to the centroid of each cluster.
- Detailed simulation is then performed on the set of simulation points.
- SimPoint also gives weight for each simulation point
    - (Number instruction represented by the intervals in the cluster) / (Total Number of instruction in the program)

Accuracy of SimPoint



Result

- SimPoint has less than a 6% error rate
- 1500 times faster on average than performing simulation for the complete program