

# Performance of Python and Haskell in Specifying and Executing List Comprehensions

Kumaravel. A, P. Kavitha, S. Thirunavukkarasu

**Abstract:** *The syntactical categories are numerous higher level programming language for facilitating the execution of instructions within optimal space and to save time. List comprehension is one such category created for parallel execution of predicates and expression in a simple specification. The purpose of List Comprehension is used to provide a executing expression easy method of simultaneously with a represented in a mathematical pattern. Each expression is evaluated and executed in a short span of time and effective with less error. It is always case change comparing the performance of such rich syntactic categories as most of the real time applications do depend on these key structures. As for as the 'Language constructs' concern, they are semantically similar in Haskell and in Python though the little differences in syntax. The aim of this work is to compare the specifications of List Comprehensions implemented in Haskell and Python environments. It is established Python reasonably comparable with Haskell in tackling List Comprehensions for certain class of simple applications.*

**Keywords:** *Haskell, Python, List Comprehension, Time complexity.*

## I. INTRODUCTION

List comprehension is a mathematical notation used for creating lists to manipulate data. It is applied as an alternative method for loop function, lambda function as well as the functions like map (), filter () and reduce (). The loop is executed for n number of iteration. The main idea of list comprehension is to optimize this loop execution[1-5]. Where in loop execution simple operations are executed in loop execution for multiple iterations that lead to a long execution time. The looping statement can use any programming pattern that has a side effect of getting different results for every execution. The performance of the program can be enhanced by optimizing the code. But it often comes at a cost. The alternate option to speed up the execution is to remove the looping structure altogether. Besides these advantages of list comprehensions, the performances are also often faster than equivalent use of map function.

**Revised Manuscript Received on July 22, 2019.**

**Dr.A.Kumaravel** Professor, Department of Information Technology, Bharath Institute of Higher Education and Research, Chennai, India. Email – drkumaravel@gmail.com

**P. Kavitha**, Assistant Professor, Department of Information Technology, Bharath Institute of Higher Education and Research, Chennai, India. Email – varshinikavitha@gmail.com

**S. Thirunavukkarasu**, Assistant Professor, Department of Information Technology, Bharath Institute of Higher Education and Research, Chennai, India. Email – ststarasu@gmail.com

The main advantages of using list comprehension are adhoc nature of its syntax, which helps in being consistent with imperative languages. Because[16-21], while constructing of codes in imperative languages it is practice to have plenty of keywords and syntax. For example, `++`, `++i`, `for(;;){}`, `while(){}`, `0x123`, `expr1 ? expr2 : expr3`, `sprint(...%s...,...)`, etc. The feature used by the functional programming is excellent when compared with any other programming languages. It also has the capability of getting executed simultaneously in multicore processor architecture. Whenever two computations are executed in parallel architecture, it doesn't get interacted with each other because of its parallel architecture design. Therefore the programming steps won't be collapsed at any order of execution. There are no certainty of allowing two instructions in the functional programming interact with each computation at any point of time similar to any other programming languages. In this experiment two functional programming languages are considered for the execution and the corresponding results are compared for the evaluation of result. The two Functional programming languages are Haskell and Python. These languages are considered for the experiment as it supports the Parallel Concepts. The features of functional programming are different from imperative and procedural programming in manipulating mathematical notation and formulas.

The high order function and the list comprehension are one-liner implementation which help in improving the computational speed. Therefore most of the experienced python programmers tend to prefer list comprehensions. It state s the intent clearly and always read from left to right also it doesn't make any function call, therefore the list comprehension solution is faster when compared with other looping functions[22-29].

## II. LITERATURE REVIEW

[30-41] Trinder et al(1995) proposes an list comprehension in a lazy functional language. The author considered the database with four algebraic functions with two implementations as enhancement strategies. The equivalent list comprehension has demonstrated in enhanced strategy, the database algorithm that uses these strategies can be emulated to improve comprehension queries. It is proved that these conversions can be employed to improve queries that need more power than the relational database.

[46-51] Wadler, Philip. (2019). proposes development

to list comprehensions that reduces the complexity of expressing queries that are written in SQL query are used for ORDER BY, GROUP BY, and LIMIT. The development includes significant power of comprehensions, and generalizes the SQL created. List comprehensions are a most familiar concept for design that provides the syntactic sugar with no new expressive power. The author offered to gave growth to Haskell list comprehensions which will parallel the execution of ORDER BY and cluster BY statements. this can be the projected technique is over general the strategies employed in the the otherlanguages, as a result of it performing arts with any operate in a very given sort, rather than being restricted to specific functions. The grouping technique is additionally uncommon because it regroups every variable in scope, from one priceto a group of values. This seems is extremely regarding sturdiness creating the behavior of cluster BY in SQL, however is maybe a lot of similarity. comprehension security are create use of wherever and HAVING clauses, and therefore the same fairlytechnique that supports every aggregation and nested lists. A accessible epitome of the rendering are enforced with the given methodology to determine its correctness and to implement it within the GHC compiler for Haskell thanks to clusterand order appear thus alike, it's traditional to reverence whether or not there may be an additional comman methodologyby that these 2 are thought of as a special case. The toughies are known as a generalization: finding C may be and even given C it {should} not be clear and difficult what the approved unfasten C should be. it's going to even be potential to generalize from lists to monads. This show relatively straightforward, modulo the issue of creating unzips. However, it's not clear whether or not or not it's use of the generalization would be sufficiently useful to justify the increase in complexity. In further, visible of the generalization of the new constructs, we tend to reverence whether or not they may alsoconstructively give analysis into the look of latest information programming language[52-58].

2] Latendresse, Mario (2007) List Comprehension may be a aphoristic well fashioned style to explain lists in practicallanguages. It uses nested generators (i.e., iterators) and filters (i.e., scientist expressions) of various stages. the previous manufacture lists, whereas the latter regulate the contents of these lists. List Comprehension translation is typicallysupported Wadler's rules that transmit code supported recursive functions. This code sometimes results have gotten either in stack overflow or in inefficient execution for many Common Lisp implementations. we've got an inclination to gift a extremely progressive technique to compile List Comprehension among the Loop Facility of Common Lisp, leading toeconomical code that skip from stack overflow. Our translation code is in addition really short, with Associate in Nursing gone code really close to the user-specified list comprehension. we've got an inclination to in addition shows a translation into further basic constructs of Common Lisp that perpetually finishes up in even additional economical code, thoughthis translation is further refined than practice the Loop Facility. It show that translating list comprehensions into the Common Lisp Loop

Facility is improbably sleek and creates further economical code as compared to Wadler's translation technique depends on recursive functions. the interpretation can also be simplycontinued to tuple assignment in generators and on vectors and strings. Compared to common recursivedefinitions, the Loop Facility keep from Associate in Nursing stack overflow since it's sometimes used as associate degree iteration mechanism that the stack house doesn't grow with the amount of iterations. The Loop Facility is economical, but we'veadditionally shown that the fundamental construct prog can provide even further economical code compare to differenttransactions, once rendering some list comprehensions. This comes with increase among the standard of the interpretation mechanism, however isn't over sophisticated practice recursive functions. The reader willalsohave completed that a group of the Loop Facility are often used as if it were a list Comprehension Facility — thoughthis is usually not as satisfy as a result of the standard syntax of List Comprehension.

### III. MATERIALS AND METHODS

#### A. List comprehensions across languages

List comprehension is a language produce that helps in specifying the contents of a list based on the set builder notation which used in mathematics to defined the members of a set.

The list can be represented as

{ Set member | generator (logical operator) filter }

Where set member desined each element of the set, the generator helps in generating values to be combined in the set. More than one generator can be used. In the case, all probable combinations are tested. And filter expressions generated values of filter, More than one filter expression can be used.

#### B. Haskell

Haskell is a functional programming that supports parallel execution. It also substantially increases programmer productivity as it has Shorter, clearer, and more maintainable code. As it obtains fewer errors it is high reliable language. The semantic gap between the language and the programmer are small it is a wide-spectrum language which is proper for a variety of applications. It is particularly applicable for programs which need to be highly modifiable and maintainable.

#### C. Python

Python is a most popular language in developing industry and also supports functional programming. It contains a lot of library to support math environment. It has build in support for Lists, Tuples, and sets to all notations to represent similar type of mathematical notation. It also has the

features in the form of package itertools package that are used for performing maths functions as Python iterator and generator, where python iterator object are used for execution of iteration in the list of collection.

#### IV. EXPERIMENTAL PROCEDURES

Experiments For the experiment python and Haskell programming languages are used for the implementation. The hardware and software setups utilized for these experiments are processors having Intel® Core™ i3, Disk space of 3 GB and windows version 7 operating system. Later the Python versions 3.6 with its compatibility and development are installed.

##### A. Implementation

List comprehensions for computing even numbers using python can be written as  
`even = [x for x in integers if x % 2 == 0]`

Though list comprehension avoids function call the execution speed will be higher while comparing to loop functions. Let us consider an example that,  
`S = [2*n for n in range(0,9) if (n % 2) == 0]`  
`print S`

The list comprehension are executed and evaluated within one line. When the same example is coded using 'for loop' it will need several lines of code and therefore the execution speed gets lowered comparing list comprehension. The number passed to the range () function is actually the integer value, which generate, result value starting from zero, of course. That is the range(10) will return [0,1,2,3,4,5,6,7,8,9] and obtain the final output of this expression syntax as prints [0, 4, 8, 12, 16]. It is a special syntax that is applied with a value and unnecessary values are filtered and the overall computations are done in one expression.

Initially it creates a list from 0 to 8 as the specified in function range (0,9), and then manipulate the value using if condition to eliminate odd numbers by  $(n \% 2) == 0$ , and in the final step it is multiplied by 2 therefore the  $2*n$ , returns the resultant list of value.

##### B. Coding

###### HASKELL PYTHON

```
pyth :: Int -> [(Int, Int, Int)]
pyth n =
  concatMap
    (\x ->
      concatMap
        (\y ->
          concatMap
            (\z ->
              if x ^ 2 + y ^ 2 == z ^ 2
              then [(x, y, z)]
              else [])
            [y .. n])
          [x .. n])
```

```
[1 .. n]
main :: IO ()
main = print $ pyth 25
  from functools import (reduce)
  from operator import (add)
  def pts(n):
    m = 1 + n
    return [(x, y, z) for x in range(1, m)
              for y in range(x, m)
              for z in range(y, m) if x**2 + y**2 == z**2]
  def concatMap(f):
    return lambda xs: (
      reduce(add, map(f, xs), []) )
  def bindList(xs):
    return lambda f: (
      reduce(add, map(f, xs), []) )
  def main():
    for f in [pts, pts2, pts3]:
      print (f(20))
  main()
```

#### V. EVALUATION MEASURES AND RESULT ANALYSIS

Here in this paper the execution speed of two functional programming implemented using list comprehension is compared as an evaluation measure. The time complexity of python language is lesser when compared to Haskell language. As shown in the figure 1 the time taken by the python program to compute even numbers using the list comprehension is 0.009 where for the Haskell language it takes 0.05 seconds.

#### VI. SUMMARY

In this paper, the advantage of using list comprehension is demonstrated by implementing it using two functional languages namely Haskell and Python. Later the performance of the functional language is evaluated using computational speed therefore to establish the better functional language for computation. while the loop Facility is economical, however here it's well-tried that the elemental style program will offer even a lot of economical code once translating loop perform to some list comprehensions. This comes with a rise within the complexness of the interpretation mechanism, nevertheless isn't a lot of complicated than mistreatment algorithmic functions. Even though, Haskell and python are functional languages that support lots of built-in library. In this experiment the mathematical notation are applied as a list comprehensions and the corresponding parameters are passed to it. As the result the execution time are calculated therefore to analyze the time complexity for the two functional languages. As the result it shown that time complexity of the Python Functional programming is lesser than the Haskell Functional Programming. It's additionally ended that a set of the loop facility are often used as if it were a listing comprehension facility.



## REFERENCES

- [1]Kumaravel A., Rangarajan K.,Algorithm for automaton specification for exploring dynamic labyrinths,Indian Journal of Science and Technology,V-6,I-SUPPL5,PP-4554-4559,Y-2013
- [2]P. Kavitha, S. Prabakaran “A Novel Hybrid Segmentation Method with Particle Swarm Optimization and Fuzzy C-Mean Based On Partitioning the Image for Detecting Lung Cancer” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-8 Issue-5, June 2019
- [3]Kumaravel A., Meetei O.N.,An application of non-uniform cellular automata for efficient cryptography,2013 IEEE Conference on Information and Communication Technologies, ICT 2013,V-,I-,PP-1200-1205,Y-2013
- [4]Kumarave A., Rangarajan K.,Routing algorithn over semi-regular tessellations,2013 IEEE Conference on Information and Communication Technologies, ICT 2013,V-,I-,PP-1180-1184,Y-2013
- [5]P. Kavitha, S. Prabakaran “Designing a Feature Vector for Statistical Texture Analysis of Brain Tumor” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-8 Issue-5, June 2019
- [6]Dutta P., Kumaravel A.,A novel approach to trust based identification of leaders in social networks,Indian Journal of Science and Technology,V-9,I-10,PP--,Y-2016
- [7]Kumaravel A., Dutta P.,Application of Pca for context selection for collaborative filtering,Middle - East Journal of Scientific Research,V-20,I-1,PP-88-93,Y-2014
- [8]Kumaravel A., Rangarajan K.,Constructing an automaton for exploring dynamic labyrinths,2012 International Conference on Radar, Communication and Computing, ICRCC 2012,V-,I-,PP-161-165,Y-2012
- [9]P. Kavitha, S. Prabakaran “Adaptive Bilateral Filter for Multi-Resolution in Brain Tumor Recognition” International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-8 June, 2019
- [10] Kumaravel A.,Comparison of two multi-classification approaches for detecting network attacks,World Applied Sciences Journal,V-27,I-11,PP-1461-1465,Y-2013
- [11] Tariq J., Kumaravel A.,Construction of cellular automata over hexagonal and triangular tessellations for path planning of multi-robots,2016 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2016,V-,I-,PP--,Y-2017
- [12] Sudha M., Kumaravel A.,Analysis and measurement of wave guides using poisson method,Indonesian Journal of Electrical Engineering and Computer Science,V-8,I-2,PP-546-548,Y-2017
- [13] Ayyappan G., Nalini C., Kumaravel A.,Various approaches of knowledge transfer in academic social network,International Journal of Engineering and Technology,V-,I-,PP-2791-2794,Y-2017
- [14] Kaliyamurthie, K.P., Sivaraman, K., Ramesh, S. Imposing patient data privacy in wireless medical sensor networks through homomorphic cryptosystems 2016, Journal of Chemical and Pharmaceutical Sciences92.
- [15] Kaliyamurthie, K.P., Balasubramanian, P.C. An approach to multi secure to historical malformed documents using integer ripple transfiguration 2016 Journal of Chemical and Pharmaceutical Sciences92.
- [16] A.Sangeetha,C.Nalini,”Semantic Ranking based on keywords extractions in the web”, International Journal of Engineering & Technology, 7 (2.6) (2018) 290-292
- [17] S.V.GayathiriDevi,C.Nalini,N.Kumar,”An efficient software verification using multi-layered software verification tool “International Journal of Engineering & Technology, 7(2.21)2018 454-457
- [18] C.Nalini,ShwtambariKharabe,”A Comparative Study On Different Techniques Used For Finger – Vein Authentication”, International Journal Of Pure And Applied Mathematics, Volume 116 No. 8 2017, 327-333, Issn: 1314-3395
- [19] M.S. Vivekanandan and Dr. C. Rajabhushanam, “Enabling Privacy Protection and Content Assurance in Geo-Social Networks”, International Journal of Innovative Research in Management, Engineering and Technology, Vol 3, Issue 4, pp. 49-55, April 2018.
- [20] Dr. C. Rajabhushanam, V. Karthik, and G. Vivek, “Elasticity in Cloud Computing”, International Journal of Innovative Research in Management, Engineering and Technology, Vol 3, Issue 4, pp. 104-111, April 2018.
- [21] K. Rangaswamy and Dr. C. Rajabhushanmc, “CCN-Based Congestion Control Mechanism In Dynamic Networks”, International Journal of Innovative Research in Management, Engineering and Technology, Vol 3, Issue 4, pp. 117-119, April 2018.
- [22] Kavitha, R., Nedunchelian, R., “Domain-specific Search engine optimization using healthcare ontology and a neural network backpropagation approach”, 2017, Research Journal of Biotechnology, Special Issue 2:157-166
- [23] Kavitha, G., Kavitha, R., “An analysis to improve throughput of high-power hubs in mobile ad hoc network” , 2016, Journal of Chemical and Pharmaceutical Sciences, Vol-9, Issue-2: 361-363
- [24] Kavitha, G., Kavitha, R., “Dipping interference to supplement throughput in MANET” , 2016, Journal of Chemical and Pharmaceutical Sciences, Vol-9, Issue-2: 357-360
- [25] Michael, G., Chandrasekar, A.,”Leader election based malicious detection and response system in MANET using mechanism design approach”, Journal of Chemical and Pharmaceutical Sciences(JCPS) Volume 9 Issue 2, April - June 2016 .
- [26] Michael, G., Chandrasekar, A.,”Modeling of detection of camouflaging worm using epidemic dynamic model and power spectral density”, Journal of Chemical and Pharmaceutical Sciences(JCPS) Volume 9 Issue 2, April - June 2016 .
- [27] Pothumani, S., Sriram, M., Sridhar, J., Arul Selvan, G., Secure mobile agents communication on intranet,Journal of Chemical and Pharmaceutical Sciences, volume 9, Issue 3, Pg No S32-S35, 2016
- [28] Pothumani, S., Sriram, M., Sridhar , Various schemes for database encryption-a survey, Journal of Chemical and Pharmaceutical Sciences, volume 9, Issue 3, Pg NoS103-S106, 2016
- [29] Pothumani, S., Sriram, M., Sridhar, A novel economic framework for cloud and grid computing, Journal of Chemical and Pharmaceutical Sciences, volume 9, Issue 3, Pg No S29-S31, 2016
- [30] Priya, N., Sridhar, J., Sriram, M. “Ecommerce Transaction Security Challenges and Prevention Methods- New Approach” 2016 ,Journal of Chemical and Pharmaceutical Sciences, JCPS Volume 9 Issue 3.page no:S66-S68 .
- [31] Priya, N.,Sridhar,J.,Sriram, M. “Vehicular cloud computing security issues and solutions” Journal of Chemical and Pharmaceutical Sciences(JCPS) Volume 9 Issue 2, April - June 2016
- [32] Priya, N., Sridhar, J., Sriram, M. “Mobile large data storage security in cloud computing environment-a new approach” JCPS Volume 9 Issue 2. April - June 2016
- [33] Anuradha.C, Khanna.V, “Improving network performance and security in WSN using decentralized hypothesis testing “Journal of Chemical and Pharmaceutical Sciences(JCPS) Volume 9 Issue 2, April - June 2016 .
- [34] Anuradha.C, Khanna.V, “A novel gsm based control for e-devices“ Journal of Chemical and Pharmaceutical Sciences(JCPS) Volume 9 Issue 2, April - June 2016 .
- [35] Anuradha.C, Khanna.V, “Secured privacy preserving sharing and data integration in mobile web environments“ Journal of Chemical and Pharmaceutical Sciences(JCPS) Volume 9 Issue 2, April - June 2016 .
- [36] Sundarraj, B., Kaliyamurthie, K.P. Social network analysis for decisive the ultimate classification from the ensemble to boost accuracy rates 2016 International Journal of Pharmacy and Technology
- [37] Sundarraj, B., Kaliyamurthie, K.P. A content-based spam filtering approach victimisation artificial neural networks 2016 International Journal of Pharmacy and Technology83.
- [38] Sundarraj, B., Kaliyamurthie, K.P. Remote sensing imaging for satellite image segmentation 2016 International Journal of Pharmacy and Technology83.
- [39] Sivaraman, K., Senthil, M. Intuitive driver proxy control using artificial intelligence 2016 International Journal of Pharmacy and Technology84.
- [40] Sivaraman, K., Kaliyamurthie, K.P. Cloud computing in mobile technology 2016 Journal of Chemical and Pharmaceutical Sciences92.
- [41] Sivaraman, K., Khanna, V. Implementation of an extension for browser to detect vulnerable elements on web pages and avoid click jacking 2016 Journal of Chemical and Pharmaceutical Sciences92.

## AUTHORS PROFILE



**Dr.A.Kumaravel** Professor, Department of Computer Science & Engineering, Bharath Institute of Higher Education and Research, Chennai, India



**P. Kavitha** Assistant Professor, Department of Computer Science & Engineering, Bharath Institute of Higher Education and Research, Chennai, India



**S. Thirunavukkarasu** Assistant Professor, Department of Computer Science & Engineering, Bharath Institute of Higher Education and Research, Chennai, India