K.Srinivasan

(also known as: Srinivasan Kannan, Ka.Shrinivaasan, Shrinivas Kannan)

Permanent Address:

Srinivasan Kannan,

S/O. P.R.ES.Kannan,

172, Gandhi Adigal Salai,

Kumbakonam-612001, TamilNadu, India.

e-mail : ka.shrinivaasan@gmail.com

shrinivas.kannan@gmail.com

kashrinivaasan@live.com

Mobile : 9791499106

Name spellings in consultancy/employer/academic records : K.Srinivasan (academics, BaaN,Sun Microsystems) Srinivasan Kannan (Verizon, Clockwork Interviews, CloudEnablers), Shrinivas Kannan(webMethods/SoftwareAG and CMI), Ka.Shrinivaasan (Global Analytics)

Personal website(research) :- https://sites.google.com/site/kuja27/

Krishna iResearch Open Source: https://sourceforge.net/users/ka_shrinivaasan, https://github.com/shrinivaasanka

OpenHub profile: https://www.openhub.net/accounts/ka_shrinivaasan

Date of Birth as per records: 19 May 1977

Passport: M9583737

PAN: AOVPS2844F

Aadhaar: 772717942542

COMPLETE ACADEMIC AND WORK HISTORY

ACADEMICS

1. Elementary Schooling, RC Morning Star, Kumbakonam (1982-1987)
2. Higher Secondary, Town Higher Secondary School,Kumbakonam (1987-1994) (SSLC -470/500 and Plus Two 1115/1200)
3. B.A - Hindi - Praveen Uttarardh – Dakshin Bharat Hindi Prachar Sabha,Chennai (privately tutored from Kumbakonam) – (1988-1992)
4. Bachelor of Engineering (Computer Science & Engg) (1995-99),
PSG College of Technology, Coimbatore-641004 INDIA (CGPA/Percentage: 8.8/87.75)
5. M.Sc (Computer Science), Chennai Mathematical Institute, Chennai (2008 – 10) CGPA 8.08 (coursework - excluding MSc thesis), 8.06(including MSc thesis)
6. Ph.D (Research Scholar in Computer Science), Chennai Mathematical Institute, Chennai (coursework, TAC 2010 - expansion of MSc thesis above - and Complement Function miniproject - August 2010 – August 2011) – resigned due to personal reasons and went back to Industry. Presently pursuing independent research.

COURSES DONE IN M.Sc and Ph.D and Self-study

M.Sc courses [CMI and IIT-Chennai] and self-study (2008-2010):

Haskell, Operating systems, Distributed systems, Theory of computation, Databases,Logic-1, Complexity-1, Principles of Programming Languages(Java and lambda calculus), Algorithms, Graph theory, Cryptography, Datamining-1, Information Retrieval, AutomataConcurrencyTimedSystems

Ph.D courses[CMI and IMSc] and self study (August 2010 - October 2011):

Complexity-2, Topics in data mining(Recommender Systems,Streaming Algorithms), Randomized algorithms(including PTAS), Logspace computation, Program Verification, Program Analysis, Program Slicing, Computational number theory and algebra, Computational geometry, Expander graphs, Combinatorics(Generating functions), Probabilistic method, Communications Complexity, Linear Programming And Combinatorial Optimization and Computational Biology (BioInformatics) algorithms for sequence alignment.

PUBLICATIONS

(Google Scholar URL - http://scholar.google.co.in/citations?user=eLZY7CIAAAAJ&hl=en)

1.(During PhD) **Decidability of Existence and Construction of a Complement of a function with Prof.Meena Mahajan, IMSc, 2011**

(http://arxiv.org/abs/1106.4102)

- Some unreviewed draft additions to the above for Complement Function Circuit and its connection to Riemann Zeta Function, Ramanujan graphs and Ihara Zeta Functions are at: **http://sourceforge.net/p/asfer/code/HEAD/tree/AstroInferDesign.txt with some illustrations:**

https://sites.google.com/site/kuja27/RamanujanGraphsRiemannZetaFunctionAndIharaZetaFunction.pdf?attredirects=0&d=1 and

https://sites.google.com/site/kuja27/RZFAndIZF_25October2014.pdf?attredirects=0&d=1

2.(Master's Thesis) **Few Algorithms for Ascertaining Merit of a document (Citation graph Maxflow, Recursive Gloss Overlap Algorithm and Interview algorithm applying either of the previous two with applications of them) with Profs. Ravindran(IIT Chennai) and Madhavan Mukund (CMI) in 2009-2010** – http://arxiv.org/abs/1006.4458.

3. (During PhD) **Evaluated NIST TAC 2010 dataset (Summarization track) for Update Summarization by applying Interview Algorithm - appeared in proceedings of TAC 2010** at:
http://www.nist.gov/tac/publications/2010/participant.papers/CMI_IIT.proceedings.pdf

RESEARCH STATEMENTS:

4. **Research Statement 1** - https://sites.google.com/site/kuja27/ResearchStatement2.pdf?attredirects=0

5. **Research Statement 2 - with some proof sketches (includes a new timeout manager algorithm implemented in Global Analytics Decision Platform 3.0)** - https://sites.google.com/site/kuja27/PhDThesisProposal.pdf?attredirects=0

6. **Research Statement 3 – with some proof sketches -** https://sites.google.com/site/kuja27/Research_Writeup.pdf?attredirects=0&d=1

PUBLICATION DRAFTS (not final versions) - in progress:

(written and typeset – complexity and machine learning – substantially expanded from publications above – unreviewed)

7.(draft1) **IntegerPartitions and Hash Functions** (https://sites.google.com/site/kuja27/IntegerPartitionAndHashFunctions.pdf?attredirects=0)

8.(draft2) **IntegerPartitions and Hash Functions**

(https://sites.google.com/site/kuja27/IntegerPartitionAndHashFunctions_2014.pdf?attredirects=0&d=1)

9.(draft1) **Interview Algorithm is in IP=PSPACE** (https://sites.google.com/site/kuja27/InterviewAlgorithmInPSPACE.pdf?attredirects=0)

10.(draft1) **Few Non-trivial questions and Shell Turing Machines** (https://sites.google.com/site/kuja27/UndecidabilityOfFewNonTrivialQuestions.pdf?attredirects=0)

11. (draft1) Lower Bounds for Majority Voting and Pseudorandom choice - https://sites.google.com/site/kuja27/LowerBoundsForMajorityVotingPseudorandomChoice.pdf

12. (draft1) Circuits for computation of error probability in majority voting - https://sites.google.com/site/kuja27/CircuitForComputingErrorProbabilityOfMajorityVoting.pdf

13. **(draft2) Circuits for computation of error probability in majority voting –** https://sites.google.com/site/kuja27/CircuitForComputingErrorProbabilityOfMajorityVoting_2014.pdf

14. **(draft1) Indepth analysis of a variant of Majority Voting with relation to ZFC** - https://sites.google.com/site/kuja27/IndepthAnalysisOfVariantOfMajorityVotingwithZFAOC.pdf?attredirects=0

15. **(draft2) Indepth analysis of a variant of Majority Voting with relation to ZFC** - https://sites.google.com/site/kuja27/IndepthAnalysisOfVariantOfMajorityVotingwithZFAOC_2014.pdf?attredirects=0

16.(draft1) A Chaos theoretic Parallel Pseudorandom generator in RNC For Majority Voting and Pseudorandom Choice (https://sites.google.com/site/kuja27/ChaoticPRG.pdf?attredirects=0)

17. (draft1) Analysis of a Randomized Space Filling Algorithm and its Linear Program Formulation (https://sites.google.com/site/kuja27/Analysis%20of%20a%20Randomized%20Space%20Filling%20Algorithm%20and%20its%20Linear%20Program%20Formulation.pdf?attredirects=0)

18. (draft2) Analysis of a Randomized Space Filling Algorithm and its Linear Program Formulation (Parallel PRG and Cellular Automaton algorithm) - **http://sourceforge.net/p/asfer/code/HEAD/tree/AstroInferDesign.txt**

19. **(draft1) Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization (version 1)** - https://sites.google.com/site/kuja27/DiscreteHyperbolicPolylogarithmicSieveForIntegerFactorization.pdf?attredirects=0&d=1

20. (draft2) Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization – with Interpolation Search (version 2) - https://sites.google.com/site/kuja27/DiscreteHyperbolicPolylogarithmicSieveForIntegerFactorization_updated_interpolation_search.pdf?attredirects=0&d=1

21. (draft3) Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization – with Interpolation Search (version 3) - https://sites.google.com/site/kuja27/DiscreteHyperbolicPolylogarithmicSieveForIntegerFactorization_updated_interpolation_search_30June2013.pdf?attredirects=0&d=1

22. (draft4) Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization – with Interpolation Search (version 4 and version 5 with handwritten illustrations and calculations) - http://sourceforge.net/projects/acadpdrafts/files/DiscreteHyperbolicPolylogarithmicSieveForIntegerFactorization_updated_interpolation_search.pdf/download

23. (draft5) Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization - using Rectangular Binary (or) Interpolation Search http://sourceforge.net/projects/acadpdrafts/files/DiscreteHyperbolicPolylogarithmicSieveForIntegerFactorization_updated_rectangular_interpolation_search.pdf/download

24. (draft6) Informal Notes on Derivation of Upperbound for Discrete Hyperbolic Factorization with Stirling Formula - using Rectangular Binary or Interpolation Search (http://sourceforge.net/projects/acadpdrafts/files/DiscreteHyperbolicFactorization_UpperboundDerivedWithStirlingFormula_2013-09-10.pdf/download)

25. (draft7) Discrete Hyperbolic Polylogarithmic Sieve For Integer Factorization – using Rectangular Binary (or) Interpolation Search and Upperbound derived with Stirling Formula http://sourceforge.net/projects/acadpdrafts/files/DiscreteHyperbolicPolylogarithmicSieveForIntegerFactorization_updated_rectangular_interpolation_search_and_StirlingFormula_Upperbound.pdf/download
(a C++ implementation of this algorithm is at: http://sourceforge.net/p/asfer/code/HEAD/tree/cpp-src/miscellaneous/DiscreteHyperbolicFactorizationUpperbound.cpp and factorization result logs at: http://sourceforge.net/p/asfer/code/HEAD/tree/cpp-src/miscellaneous/)

26. (draft8) An NC algorithm and some Sequential Search Algorithms for Discrete Hyperbolic Polylogarithmic Sieve For Factorization using Binary or Interpolation Search with

Stirling Formula and Logarithmic Sorted Tile Merge in PRAM model - http://sourceforge.net/projects/acadpdrafts/files/DiscreteHyperbolicPolylogarithmicSieveForIntegerFactorization_PRAM_TileMergeAndSearch_And_Stirling_Upperbound.pdf/download

27. (draft 9) Parallel RAM algorithm for Discrete Hyperbolic Factorization - AsFer PRAM implementation design notes with tile id(s)
(http://sourceforge.net/p/asfer/code/HEAD/tree/asfer-docs/ImplementationDesignNotesForDiscreteHyperbolicFactorizationInPRAM.jpg)

28. (draft10) An NC algorithm and some Sequential Search Algorithms for Discrete Hyperbolic Polylogarithmic Sieve For Factorization using Binary or Interpolation Search with Stirling Formula and Logarithmic Sorted Tile Merge in PRAM model - updated draft with PRAM to NC reduction and input size details and references - http://sourceforge.net/projects/acadpdrafts/files/DiscreteHyperbolicPolylogarithmicSieveForIntegerFactorization_PRAM_TileMergeAndSearch_And_Stirling_Upperbound_updateddraft.pdf/download
(Most recent of all the previous drafts on Discrete Hyperbolic Factorization)

29. Individual Invention Disclosure for Survival Index Based Transaction Timeout Manager (Sun Microsystems) - 2002-2003
https://sites.google.com/site/kuja27/SurvivalIndexBasedTxnTimeoutManager.pdf?attredirects=0

30. AstroInfer (AsFer) Open Source Product Design with theoretical interludes which substantially expand on the publications above - http://sourceforge.net/p/asfer/code/HEAD/tree/asfer-docs/AstroInferDesign.txt

31. VIRGO Open Source Product Design with theoretical interludes - http://sourceforge.net/p/virgo-linux/code-0/HEAD/tree/trunk/virgo-docs/VirgoDesign.txt

32. King Cobra - Byzantine distributed request servicing software design with queues and arbiters – theoretical interludes - https://sourceforge.net/p/kcobra/code-svn/KingCobraDesignNotes.txt

33. (draft1) Informal notes – 1 : on Implication Graphs, Error probability of Majority Voting and P Versus NP Question http://sourceforge.net/projects/acadpdrafts/files/ImplicationGraphsPGoodEquationAndPNotEqualToNPQuestion_excerpts.pdf/download

34. (draft1) Informal notes - 2 : on Minimum Convex Hulls of Implication Graphs and Hidden Markov Model on class nodes of Concept Hypergraph

35. (draft1) Informal notes - 3 : on Minimum Convex Hulls of Implication Random Growth Networks and Perfect Voter Decidability

36. (draft1) Informal handwritten notes on Philosophical Analysis of Democracy Circuit and Pseudorandom Choice - https://sites.google.com/site/kuja27/PhilosophicalAnalysisOfDemocracyCircuitAndPRGChoice_2014-03-26.pdf

37. (draft1) Informal Handwritten Notes - Distinct Partitions, Hash collision chains and Schur Theorem - https://sites.google.com/site/kuja27/SchurTheoremMCPAndDistinctPartitions_2014-04-17.pdf

38. (draft1) Document Summarization from WordNet Subgraph obtained by Recursive Gloss Overlap - https://sites.google.com/site/kuja27/DocumentSummarization_using_SpectralGraphTheory_RGOGraph_2014.pdf?attredirects=0&d=1

39. (draft1) Arrow's Theorem, Circuit For Democracy and Pseudorandom Choice and P Versus NP - https://sites.google.com/site/kuja27/CircuitsForDemocracyAndPseudorandomChoice_and_PVsNP.pdf?attredirects=0&d=1

40. (draft1) Krishna iResearch Open Source Products (AsFer, USBmd, VIRGO, KingCobra, Acadpdrafts) - High Level Handdrawn Architecture Diagram - http://sourceforge.net/p/acadpdrafts/code/ci/master/tree/Krishna_iResearch_opensourceproducts_archdiagram.pdf

---

41. (draft1) Miscellaneous notes on Krishna iResearch Open Source products design, Democracy Circuit, Complement Function circuit and Parallel RAM to NC reduction for ANSV algorithm in Discrete Hyperbolic Factorization

---

TEAM PATENTS

---

1. Simultaneous global transaction and local transaction management ...

US Pat. 7610305 - Filed 24 Apr 2003 - Issued 27 Oct 2009 - Sun Microsystems, Inc.

2.Read/write lock transaction manager freezing

US Pat. 7739252 - Filed 14 Jul 2003 - Issued 15 Jun 2010 - Oracle America, Inc.

3.Utility for configuring and verifying data sources

US Pat. 7134008 - Filed 4 Sep 2003 - Issued 7 Nov 2006 - Sun Microsystems, Inc.

4.Transaction optimization of read-only data sources

US Pat. 7165061 - Filed 31 Jan 2003 - Issued 16 Jan 2007 - Sun Microsystems, Inc.

5.Common transaction manager interface for local and global transactions

US Pat. 7743083 - Filed 24 Apr 2003 - Issued 22 Jun 2010 - Oracle America, Inc.

6.Specifying transaction manager type at various application levels

US Pat. 7082432 - Filed 24 Apr 2003 - Issued 25 Jul 2006 - Sun Microsystems, Inc.

7.Transaction manager freezing

US Pat. 7640545 - Filed 14 Jul 2003 - Issued 29 Dec 2009 - Sun Microsytems, Inc.

8.Identity for data sources

US Pat. App 10655346 - Filed 4 Sep 2003 – Sun Microsystems, Inc.

9.Common transaction manager interface

US Pat. App 10422453 - Filed 24 Apr 2003 – Sun Microsystems, Inc.

DETAILS OF WORK (1999 – Present)

C/C++/Java/Python on Windows and various Unix flavours

PRODUCTS

Netscape Application Server 4.0, iPlanet/SunONE Application Server 6.x, SunONE Web/Proxy Server 3.6/4.0, Apache web server 1.4.x/2.0.x, webMethods Broker Messaging Server 5.x/6.x/7.x, ASFER Classification and Inference Software for Large Data Sets (10.0,12.0,14.9.9,15.1.8,15.6.15), USBmd (1.0,3.0,14.9.9,15.1.8,15.6.15), VIRGO Linux Kernel Extensions for Cloud (5.0,6.0,7.0,8.0,10.0,12.0,14.9.9,15.1.8,15.6.15), Global Decision Platform(GDP) 2.3.0/ 2.3.1/ 2.5.0/ 2.5.1/ 2.7 /2.7.1/3.0, KingCobra (1.0,14.9.9,15.1.8,15.6.15), Automated Modelling Platform(AMP), Python 2.4.3 and 2.7 source code, Linux kernel [3.2.0, 3.7.8, 3.9.x, 3.15.5, 4.0.5, 4.1.5, 4.10.3], Maitreya's Dreams 7.0.3, STL, BOOST, wxWidgets library, Swiss Ephemeris API (based on NASA JPL

DE406 Ephemeris).

LANGUAGES/PLATFORMS/TOOLS

C/C++, Java 1.5-1.8, J2EE, Python, R and rpy2, Haskell, Solaris 6-10, Windows 2000, HP-UX 11.23 (PA-RISC and IA64),AIX, Linux, Sun Studio, CORBA (Visibroker), VAX FORTRAN, VAX VMS, GNU collection, MS Visual Studio .NET C++, WinDbg, DDD, Visual Basic and ASP, OptimizIt, Eclipse, EclipseCDT, Code::Blocks, ActiveState Python, Cython, IDLE, Haskell FP, Insight, GreatCircle, Valgrind, Callgrind, Python Dumbo MapReduce on Hadoop, Python NLTK, Java MapReduce on Hadoop, Apache HBase, Apache Pig, Apache Hive, Apache Cassandra, Apache Mahout, Apache Spark, Pygraphs, GraphViz, DOT, PiCloud, ActiveMQ, RabbitMQ, Pyutilib workflow, PyF workflow, Linux kernel – [3.2.0, 3.7.8, 3.9.x,3.15.5,4.0.5,4.1.5], Maitreya's Dreams 7.0.3, wxWidgets library, Swiss Ephemeris API (based on NASA JPL DE406 Ephemeris), BioInformatics Sequence Alignment Tools (BioPython, ClustalOmega), ROOT(CERN), Geolocation Python API, Beautiful Soup, CRF tools, GATE, SentiWordNet, Stanford CoreNLP, ENT, MongoDB, WEKA, Jinja2,Flask,Cloud related - Corestack, Openstack, Mistral, Jclouds, Libcloud, Docker etc.,Tornado,ZeroMQ,Kafka

---

May 2003 – present – Krishna iResearch (parallel, self-started, not-for-profit, non-funded, open-source long-term research initiative) - https://sourceforge.net/users/ka_shrinivaasan and replicated at https://github.com/shrinivaasanka

---

Working on Research, Design and Development of open source projects – a machine-learning, cloud and queue augmented Linux kernel - NeuronRain - started by self, copyleft licensed under GPL v3 . This is not a commercial startup yet. Premium technical support is

available for above opensource codebases. The GitHub repositories implement NeuronRain Enterprise version and SourceForge repositories implement NeuronRain Research version. Latest version 2017.01.23:

1. Krishna_iResearch_DoxygenDocs - Documentation for AsFer, VIRGO, KingCobra, Acadpdrafts and USBmd open source product codebases which are subsystems of Krishna iResearch Intelligent Cloud OS - NeuronRain (https://github.com/shrinivaasanka/Krishna_iResearch_DoxygenDocs , http://neuronrain-documentation.readthedocs.io/en/latest/)

2. AsFer (subsystem of NeuronRain) - AstroInfer astronomy and astrology machine learning inference open source software which uses algorithms viz.,

2.1 Bayesian Classifier,

2.2 Support Vector Machines,

2.3 Decision Tree Classifier,

2.4 String mining - Pairwise String Sequence Alignment (Needleman-Wunsch) ,

2.5 String mining - Multiple String Sequence Alignment(BioPython,ClustalOmega),

2.6 String mining - Powerset construction (C++) and String matching implementation in python(Jellyfish distance measures like Jaro-winkler, Edit-distance, Phonetic Matching rate etc.,) and

2.7 Parsers and Encoding features for USGS and NOAA HURDAT2 data

2.8 Automated Generation of Training and Test set for Classifiers that segregate the datasets into Event Classes and Automated generation of encoded astronomical object files specific to each Event Class.

2.9 Sequential implementation of Discrete Hyperbolic Factorization and PRAM design for publication drafts in https://sites.google.com/site/kuja27/

A Spark-Python-C++ implementation of Parallel NC Factorization has been included which internally executes NC Bitonic tiles Parallel Merge Sort on Spark Cloud.

2.10 A Concept Hypergraph and HMM(Hidden Markov Model) based Experimental Inference Model Design inspired by Cognitive Psychology – a minimal sentiwordnet evocatives implementation of this sans HMM named ThoughtNet is part of AsFer

2.11 Design for extracting patterns from numerical data

2.12 Time Series Analysis - A Chaos Attractor implementation that generates a sequence of numbers using logistic equation and uses Python and R to compute a correlation coefficient between a non-Chaotic and Chaotic sequence of numbers.

2.13 Discrete Fourier Transform computation for a parsed numerical input sequence using Python and R

2.14 Time Series Analysis - Spline interpolation, LOESS regression, ARMA, Approximate Linear Polynomial interpolation and graph plot for a parsed numerical input sequence using Python and R (at present implemented for Dow Jones Industrial Average historical dataset and Riemann Zeta Function Zeros dataset)

2.15 An Experimental Text Compression algorithm that uses Hidden Markov Model Maximum Likelihood Estimation based on a String Complexity measure that compresses vowels in text

2.16 An Experimental Minimum Description Length implementation that uses the previous string complexity measure, computes MDL using Kraft inequality and Shannon entropy.

2.17 Probabilistic Approximate Correct – PAC - Learning Implementation related to http://arxiv.org/abs/1106.4102

2.18 Wagner-Fischer edit distance implementation for pairwise similarity of encoded strings

2.19 An example implementation of Interview Algorithm and Intrinsic Merit Ranking for : http://arxiv.org/abs/1006.4458 and http://www.nist.gov/tac/publications/2010/participant.papers/CMI_IIT.proceedings.pdf

2.20 An experimental Expirable Objects implementation

2.21 Linear and Logistic Regression

2.22 K-Means clustering implementation

2.23 K-Nearest Neighbours supervised classifier implementation

2.24 Linear Perceptron with Gradient Descent Learning

2.25 Encoding of Maitreya Dreams to strings and some bugfixes for SWISS Ephemeris degree computation errors

2.26 An experimental logical clock for cloud – EventNet – using Python pygraphs and C++ boost::graph with GraphViz rendering (from dot files)

2.27 Longest Common Substring implementation for extracting common patterns in already classified and clustered encoded strings dataset

2.28 Python script for translating the Pairwise Longest Common Substrings obtained from a cluster of Kmeans or KNN clustered astronomical dataset into rules that correlate location of astronomical entities to a class of events.

2.29 Utility Knuth-Morris-Pratt String Match algorithm implementation

2.30 Multipurpose bigdata analytics subsystem (storage abstraction on Cassandra/Hbase/Hive/File) for computing metrics from datasets before and after processing by above ML algorithms.

2.31 Streaming Algorithms – Flajolet-Martin LogLog and HyperLogLog Counter implementations for computing cardinality (distinct elements) in streamed multiset data

2.32 Streaming Algorithms – CountMinSketch implementation for computing frequencies (heavy hitters) of elements in streamed multiset data

2.33 Streaming Algorithms - Bloom Filter implementation for membership query

in streamed multiset data stored in Apache Cassandra/Hbase/Hive/File NoSQL tables or disk and stream-simulated through python Generator/Iterable abstraction

2.34 Pig script clients for Hbase/Hive and Python clients for Hive/Cassandra/Hbase

2.35 Apache Spark Python RDD Transformation MapReduce script for mining Linux kernel logs and exporting the mined data as config to VIRGO kernel_analytics module. This integrates the AsFer or other Machine Learning algorithms into VIRGO Linux Kernel (with USBmd, KingCobra modules).

2.36 Sequence Mining of ordered encoded strings (Apriori GSP Algorithm) for inferring Class Association Rules for USGS Earthquake datasets

2.37 Named Entity Recognition using Conditional Random Fields and Hidden Markov Models (by finding Viterbi path through Viterbi Dynamic Programming Algorithm)

2.38 Graph Search – Mine for Graph Relations (and render them in graphics) in Text Data by constructing the WordNet subgraph using the recursive gloss overlap algorithm in http://arxiv.org/abs/1006.4458 and http://www.nist.gov/tac/publications/2010/participant.papers/CMI_IIT.proceedings.pdf

2.39 New Unsupervised text classification algorithm implementation that computes nodes with prominent core numbers and PageRanks (nodes with high core-numbers and PageRanks are names of the classes) in the definition graphs obtained by Recursive Gloss Overlap algorithm above.

2.40 Lambda Expression Compositionality from the depth first search closure of Recursive Gloss Overlap graph obtained from a text projected to WordNet.

2.41 Social Network Analysis – Graph creation from social networking sites data (LinkedIn profiles, Twitter Followers etc.,) and their Psycho-Social-Sentiment Analysis (Bonacich Power Centrality, Eigen Vector Centrality, Recursive Gloss Overlap Graph Sentiment Belief Propagation and SentiWordNet based sentiment scoring etc.,)

2.42 Cloud Object Move (or) Cloud Perfect Forwarding implementation – This is a standalone client-server implementation that extends C++ std::move() and rvalue references

to moving objects across cloud without copies. Functionality similar to this is required in KingCobra currency object transactional move without duplication.

2.43 Deep Learning – Convolution Neural Networks implementation (multifeature kernel maps) for a primitive pattern recognition

2.44 Deep Learning – Backpropagation in Multilayered Neural Networks implementation – applied to a Software Analytics usecase of CPU and Memory usage

2.45 Sequence Mining of ordered encoded strings (Apriori GSP Algorithm) for inferring Class Association Rules for HURDAT2 NOAA Hurricane datasets

2.46 Dependency Injection based BigData Storage Backend Abstraction subsystem for SQL and NoSQL (MySQL, MongoDB etc.,)

2.47 C++-Python Embedding - Python CAPI Embedding which executes python machine learning scripts from C++

2.48 Config File Support for selectively enabling/disabling AsFer algorithms.

2.49 Web Spider implementation for crawl-scrape of URLs (e.g streamed news updates) with Scrapy Framework

2.50 Sentiment Analyzer implementation (with RGO algorithm etc.,) for Spidered Texts (e.g streamed news updates)

2.51 Streaming Frequency Estimation (Heavy Hitters) with CountMeanMinSketch implementation

2.52 An expanded implementation of Interview Algorithm and Intrinsic Merit Ranking with added complexity-machine learning theoretical analysis for :

http://arxiv.org/abs/1006.4458 and

http://www.nist.gov/tac/publications/2010/participant.papers/CMI_IIT.proceedings.pdf

that ranks a crawled-spidered webpage, writes a DOT file of the graph, visualizes the RGO NetworkX graph in matplotlib, constructs a junction tree and computes standard qualitative graph complexity measures (viz., connectivity, mincut, treewidth of junction tree etc.,) where as intrinsic merit is quantitative (depends on number of edges, vertices, extent of overlap

and depth of recursion)

2.53 A rule search script that sifts through a range of astronomical data (with Maitreya-Swiss Ephemeris) and matches sequence mined rules from past training data to predict future weather phenomena (experimental).

2.54 Python Tornado based WebService that exposes REST API via a browser GUI in URL http://host:33333/neuronrain_auth At present a login and simple template that accepts AsFer scripts and arguments as inputs and executes, has been implemented (and an experimental redundant AngularJS client interface) With this NeuronRain is a SaaS and PaaS product that can be deployed on Cloud OS and Containers like Docker.

2.55 Apache Spark MapReduce implementation of Interview Algorithm and Intrinsic Merit Ranking (Global MemCache and Local dict() cache enabled) that parallelizes Recursive Gloss Overlap graph construction algorithm in:

http://arxiv.org/abs/1006.4458 and
http://www.nist.gov/tac/publications/2010/participant.papers/CMI_IIT.proceedings.pdf

and is thus deployable on large scale clouds. Presently benchmarked on single node Spark cluster with local threads SparkContext and various spark-defaults.conf settings.

2.56 Singular Value Decomposition and Principal Component Analysis - R+rpy2 wrapper implementation.

2.57 Approximation with a Probability Distribution – Kullback-Leibler Divergence

2.58 Basic statistics – L1,L2 norms, median etc.,

2.59 A Generic Theory and Implementation of Recursive Lambda Function Growth Learning Algorithm based on Cognitive and Psycholinguistics of which Recursive Gloss Overlap implementation is a special case.

2.60 Python-C-VIRGO Kernel – Cpython extensions – invoking VIRGO system calls (both VIRGO32 and VIRGO64 kernels) from Python userspace for dynamic setting of kernel analytics config key-value pairs

2.61 Python-C++-VIRGO Kernel – Boost::Python extensions – invoking VIRGO

system calls(both VIRGO32 and VIRGO64 kernels) from Python userspace for dynamic setting of kernel analytics config key-value pairs

2.62 Cython Python-C compiler Optimization of previous PySpark Interview Algorithm implementation that has been benchmarked on single node cluster to have substantial performance gains.

2.63 Software Analytics – SATURN Program Analysis Software integrated with VIRGO – SATURN error logs are analyzable by AsFer.

2.64 Perl WordNet::Similarity support and WordNet hypernym-hyponym path subroutine for 2 text words.

2.65 Bitonic Sort implementation – Serial and Parallel on Spark – This can be used as a bigdata sorting tool for numeric data which is in NC when run on Cloud and is O(nlogn) in serial. In the absence of PRAM implementation, NC Bitonic Sort has been invoked as a suitable alternative in parallel tile merge sort step of Parallel Discrete Hyperbolic Factorization Implementation.

2.66 Long Term Short Term Memory Recurrent Neural Networks implementation

2.67 Reinforcement Learning – Monte Carlo Policy Search implementation and ThoughtNet evocative actions based implementation

2.68 File and Neo4j Graph Database ThoughtNet hypergraph storage backend (which can be generalized to store any graph relations including WordNet subgraphs constructed by Recursive Gloss Overlap and can be queried in NoSQL)

2.69 Automatic ThoughtNet Hypergraph Creation from Recursive Gloss Overlap classified edges input from environment.

2.70 Software Analytics – Spark computed Cyclomatic Complexity of codebases from VIRGO linux SATURN driver generated .dot graph files

2.71 Audio Pattern Mining – Machine learning of patterns in music samples by computing Jensen-Shannon Divergence Distance between Fast Fourier Transform Frequency Probability Distributions of music data

2.72 Patterns within Strings – Longest Repeated Substring Implementation by Suffix

Arrays – as a special case pattern in binary encoded time series data is retrieved.

2.73 Locality Sensitive Hashing Implementation – for clustering similar documents and nearest neighbour search.

2.74 ZeroMQ CLI client and server frontend for concurrent requests to NeuronRain

2.75 Kafka Message Queue backend for AsFer datasource abstraction which includes KingCobra request-reply disk persisted queue.

2.76 Boyer-Moore Algorithm implementation for finding Majority element in Streaming Sequences.

2.77 Graph Mining – GSpan algorithm implementation for graph substructure mining

2.78 ThoughtNet Hypergraph Evocation based Recommender Systems Implementation

2.79 Software Analytics – by Deep Learning (Convolution, RecurrentLSTM, RecurrentGRU, BackPropagation) from system performance logs

2.80 Polynomial representation of text strings and distance measures based on it (L2 norms and Jensen-Shannon divergence) – alternative to edit distance

2.81 Polynomial Reconstruction – Error correcting codes – BerlekampWelch decoder implementation

2.82 Polynomial Time Approximation Scheme CNF SAT Solver – solves CNF SAT with high probability depending on error of least squares algorithm

2.83 Java Spark Streaming and ETL for analyzing generic arbitrary URL data – extracts text from any URL with Jsoup and does Spark Dstream and RDD transformations and actions on it.

2.84 Deep Learning – Convolution Networks + Backpropagation Multilayered Neural Networks implementation for image pattern recognition(PILlow imaging library support).

2.85 Deep Learning – Recurrent Neural Networks – Gated Recurrent Unit implementation

2.86 Unified Streaming Abstract Generator ETL pipeline (Jsoup + Java Spark Streaming Parquet/Kafka/Cassandra/Hive/Logfiles + Java/Python Spark transformations

and Loading)

2.87 Cloud implementations of recursive gloss overlap graph intrinsic merit, bitonic sort have been Benchmarked with Spark 2.1.0 and HiveServer2 Metastore is supported

2.88 Recursive Lambda Function Growth - Random Walk based intrinsic merit computation by growing recursive lambda function composition tree for each random walk in Recursive Gloss Overlap Graph of a text

2.89 Korner Graph Entropy – an alternative Quantitative Intrinsic Merit measure for Recursive Gloss Overlap graph of a text

2.90 Usecases for deploying NeuronRain

2.91 Graph Theoretic alternative instrinsic measures for Recursive Gloss Overlap Graph based on Kleinberg small world lattice, Cheeger Constant, Expander Graphs and Bose-Einstein Condensate Fitness model.

2.92 Index implementation for crawled documents with LocalitySensitiveHashing-Redis key-value store and ThoughtNet Hypergraph as backends

2.93 Small World Indexless Hyperball web crawler – based on Stanley Milgram, Kleinberg lattice Small world experiments and Hyperball algorithm.

2.94 Recursive Lambda Function Growth – maps text to a neural network - Graph Tensor Neuron Network which is amalgamation of Graph Neural Network and Tensor Neuron Network for evaluation of lambda function composition tree for each random walk in Recursive Gloss Overlap Graph of a text

2.95 ThoughtNet Index and LSH Index Query from GUI and Rank text by Graph Tensor Neuron Intrinsic Merit and Korner Entropy

2.96 Unguided Text Summarization – based on Unsupervised Recursive Gloss Overlap graph core number classifier: traversal of k-core subgraph and sentences related to prominent core number class vertices

2.97 ConceptNet 5.4 Python RESTful Client implementation (a potential alternative to WordNet)

2.98 Unsupervised text classifier by ranking based on Betweenness Centrality,

Closeness Centrality, Degree Centrality of the word vertices in Recursive Gloss Overlap graph of texts.

AstroInfer is an inference system for Big Data to mine patterns in them. Presently implemented for mining patterns in astronomical datasets(degrees of astronomical objects viz planets, constellations etc.,) and prediction based on rules and execution of those rules, but works for any dataset. It is also used in USBmd and VIRGO codebases below for traffic analytics and kernel analytics. Design started in May 2003.

(Latest NeuronRain AsFer research version Code at sourceforge repository: http://asfer.sourceforge.net and periodically updated Design Details at: http://sourceforge.net/p/asfer/code/HEAD/tree/asfer-docs/AstroInferDesign.txt)
(Latest NeuronRain AsFer enterprise version Code at github repository: https://github.com/shrinivaasanka/asfer-github-code and periodically updated Design Details at:https://github.com/shrinivaasanka/asfer-github-code/blob/master/asfer-docs/AstroInferDesign.txt)

3. VIRGO (subsystem of NeuronRain) – VIRtual Generic Os Linux Kernel (compatible with 4.x.x mainline kernel) – Linux Kernel Extensions for cloud - kernel modules, system calls for cloud rather than at application level high up the stack with :

3.1 Config file support

3.2 Psuedorandom Generator Based Loadbalancer

3.3 Kernel space remote execution

3.4 User space remote execution with kernel upcall and pthread creation of userspace library function or executable

3.5 Example unit test cases

3.6 Usermode output redirected logging feature for Kernel upcall to Userspace

3.7 Intermodule Function Invocation in Kernel Space - through which any machine on cloud can be completely remote-controlled deep upto board and hardware cards through function names or commands sent through virgo_clone() calls.

3.8 CPU Pooling Driver – Multi-kernel-threaded VIRGO cloudexec Kernel Driver Module for unrestricted service of virgo_clone or other client requests.

3.9 Memory Pooling Driver (and a key-value store) - and system calls and userspace clients for it on Cloud nodes – virgo_malloc, virgo_get, virgo_set, virgo_free.

3.10 Queueing Driver - that implements a wrapper over linux workqueue and also a native local queue - used for KingCobra requests queuing with handler invocation. Also implemented is a standalone kernel queueing service that listens on requests rather than being forwarded by CPU and Memory pooling drivers above.

3.11 VIRGO Cloud File Systems Driver - that implements distributed cloud file system calls and telnet userspace clients for – virgo_open, virgo_close, virgo_read and virgo_write of a file on remote cloud node

3.12 All the drivers above for CPU, Memory and File System on cloud have 3 paths each for telnet connection to remote driver kernel server socket and system call connection to remote driver kernel server socket – 1) parameter is executable in userspace 2) parameter is a function name which uses a kernel upcall plugin to execute in userspace 3) parameter is a function name executable in kernel space – configured by a boolean flag with in the driver binaries. Based on this flag either kernel upcall to userspace or kernel intermodule invocation is done.

3.13 A config driver - for exporting config symbols

3.14 Experimental Bakery algorithm kernel module implementation – for

synchronization in cloud

3.15 Utilities driver kernel module – a universal kernel module with exported utility function symbols that can be invoked across VIRGO Linux subsystems and Linux kernel including EvenNet logging kernel socket client and skbuff kernel socket debugger.

3.16 Experimental EventNet driver kernel service module – This listens on incoming EventNet log messages (Vertex and Edge) and writes to EventNet Vertex and Edge text files by VFS write. These files can then be massively processed by the boost::graph or pygraph GraphViz code in AsFer to create DOT files and graphics. Event vertices and edges can be logged by virgo_eventnet_log() utility function from any kernel module on any VIRGO cloud node.

3.17 From the above EventNet graph, a logical time ordering can be obtained on the cloud events and partakers which is for example useful in establishing money trail in KingCobra MAC eletronic money.

3.18 Kernel Analytics Kernel Module – reads the config key-value pairs set by AsFer or any other machine learning software and exports them which can be looked-up in any other kernel module. The key-value config are analytics variables learnt by mining kernel or other logs and objects. With this an adaptive dynamic kernel which changes over a period of time depending on a machine learnt config is obtained. At present an Apache Spark usecase which mines kern.log and exports a config variable through kernel_analytics has been implemented.

3.19 Thus VIRGO has all the requisite minimum functionalities for a linux variant cloud, artificially intelligent, operating system

3.20 SATURN Program Analysis Module – Intercepts linux build and extracts program analysis data in trees. SATURN logs are analyzable by AsFer (e.g null pointers, aliasing, memory etc.,).

3.21 64 bit version of VIRGO Linux kernel based on mainline 4.10.3 kernel (in a separate repository) is in development and has been found to be better performing and

stabler than VIRGO 32bit. It has lot of random panics related resolutions related mostly to i915 drivers. Though functionally similar to 32 bit VIRGO linux, 64 bit kernel is suited for large RAM installations, with CPU level security features and faster than 32 bit binaries.

VIRGO enables viewing entire cloud as a single "logical machine" upto hardware level with machine learning support by implementing lowlevel primitives that can be invoked by highlevel cloud OS components viz., OpenStack Neutron. This differentiates from other cloud libraries which are in userspace mostly. AstroInfer together with KernelAnalytics-VIRGO-USBmd-KingCobra make it a Cloud OS kernel with Machine Learning and Analytics abilities. Support for mobile operating systems in cloud are, among other things, long term goals for VIRGO.
(Latest NeuronRain VIRGO research version Code at sourceforge repository: https://sourceforge.net/projects/virgo-linux/ , https://sourceforge.net/projects/virgo64-linux/ and periodically updated Design details at: http://sourceforge.net/p/virgo-linux/code-0/HEAD/tree/trunk/virgo-docs/VirgoDesign.txt)
(Latest NeuronRain VIRGO enterprise version Code at github repository: VIRGO – https://github.com/shrinivaasanka/virgo-linux-github-code , https://github.com/shrinivaasanka/virgo64-linux-github-code and periodically updated Design details at: https://github.com/shrinivaasanka/virgo-linux-github-code/blob/master/virgo-docs/VirgoDesign.txt)
Design started in 2008.

4. USBmd (subsystem of NeuronRain) - Linux kernel USB device driver module for debugging USB issues using machine learning algorithms for fault detection :
Contains a modified version of Linux kernel mainline USB and USBWWAN broadband modem drivers with debug statements and functions that capture and dump data transfer buffers in kern.logs. These logs are parsed and processed with Apache Spark and other AsFer machine learning algorithms (e.g Streaming Algorithm implementations) to create a debug and traffic

analytics for in and out data – a typical software analytics usecase with widespread necessity in anti-cybercrime, anti-theft and anti-virus products..

(Latest NeuronRain USBmd research version Code at sourceforge repository: https://sourceforge.net/p/usb-md/ and periodically updated Design Details at: http://sourceforge.net/p/usb-md/code-0/HEAD/tree/USBmd_notes.txt).
(Latest NeuronRain USBmd enterprise version Code at github repository: https://github.com/shrinivaasanka/usb-md-linux-github-code and periodically updated Design Details at: https://github.com/shrinivaasanka/virgo-linux-github-code/blob/master/USBmd_notes.txt).

5. King Cobra (subsystem of NeuronRain) – Byzantine Distributed Request Servicing and Computational Economics Software with Queueing and Arbiters (that includes a new experimental Electronic Money Protocol – MAC (MessageAsCurrency) Design – which depends on Google Protocol Buffer based Cloud Object Move functionality in AsFer) on either Krishna iResearch intelligent Cloud platform or Hadoop Cluster
(NeuronRain KingCobra research version sourceforge Repository - https://sourceforge.net/p/kcobra/ and periodically updated design notes at https://sourceforge.net/p/kcobra/code-svn/KingCobraDesignNotes.txt)
(NeuronRain KingCobra enterprise version sourceforge Repository - https://github.com/shrinivaasanka/kingcobra-github-code and periodically updated design notes at https://github.com/shrinivaasanka/kingcobra-github-code/blob/master/KingCobraDesignNotes.txt)

Implements a minimal kernelspace and userspace messaging framework using VIRGO cpupooling (virgo_clone) and memory pooling drivers that in turn queues the requests into a kernel workqueue. There is also a standalone VIRGO queue kernel service that listens on the requests without dependencies on VIRGO cpupooling and memorypooling drivers to forward

the requests. The VIRGO workqueue handler pops the request from workqueue and invokes KingCobra driver's servicerequest exported function and replies to the publisher and optionally disk persists the incoming requests to filesystem through VFS. Differentiator is KingCobra implements a cloud messaging with a decentralized queue with disk persistence and workflow in kernel level so that hardware is easily integrated into cloud.

6. acadpdrafts - All publications and drafts with code (along with important documents and Photo ID proofs) are uploaded at http://sourceforge.net/projects/acadpdrafts/

The opensource codebases above are nonprofit academic research efforts. Premium Technical Support for above opensource codebases available. Premium Dual licensed Closedsource products (E.g GRAM – a graph theoretic ranking algorithm that enhances Recursive Gloss Overlap) derived from codebases in: **https://github.com/shrinivaasanka** and https://sourceforge.net/u/userid-769929/profile/ are in development since 2010.

---

7. GRAFIT Open Learning : https://github.com/shrinivaasanka/Grafit/tree/master/course_material – May 2010 – Present – Free online courses on miscellaneous topics and based on NeuronRain codebases - An introductory presentation on NeuronRain based Analytics is at:

https://github.com/shrinivaasanka/Grafit/blob/master/EnterpriseAnalytics_with_NeuronRain.pdf

---

April 2015 – September 2015 - CloudEnablers - CusDelight - Corestack, Chennai – Architect

1. Design and PoC implementation in python for translating Topology Orchestration Specification for Cloud Applications (TOSCA) XML BPMN Plan Models to Stackforge/Mistral Workflow YAML (Mistral-translator - similar to Heat-translator project for Simple TOSCA YAML to Heat Orchestration Templates)
2. Design and PoC implementation in python for Jinja2 rendering template feature addition to TOSCA Plan XML and Mistral YAML based on JSON context for conditionals and control structures support
3. Creating a Capability matrix of different Cloud Orchestration and template softwares: AWS,Heat,Mistral,Puppet,Chef,Ansible,Saltstack,Twisted,Nunjucks
4. Study of OpenTOSCA implementation of TOSCA (University of Stuttgart) and feasibility of its integration into Corestack (Cloudenablers)
5. Feasibility study of new features to Mistral Workflow Engine Parser (if-else,nested for-each,with-items,concurrency)
6. Feasibility of Apache Brooklyn CAMP blueprints based cloud application deployment
7. Design and PoC Development of Service Dynamic Data Discovery, MetaData and Actions REST API, Cloud Federation and Abstraction Layer (for abstracting providers, federated clouds and multiclouds like ComputeNext, Libcloud, Jclouds, Dasein, AWS etc.,) for provider abstraction and Directory Service REST API
8. Corestack code review
9. Study of feasibility of Jclouds integration into corestack
10. Design and PoC for Process Checkpoint and Restoration with CRIU in baremetal, VMs and Docker (with overlay and aufs union mount filesystems). Docker public repository - https://hub.docker.com/r/srinivasankannan/cloud-migration/
11. Design and PoC sync scripts implementation for Automated VM image synchronization with bup, rsync, lsyncd-lua and inotifywait (inotify-tools).
12. Design and PoC implementation for Service Discovery through above REST API and migrating them with Jinja2 Heat and Mistral YAML templates extraction and Performance tunables for cloud.

June 2014 – July 2014 – Clockwork Interviews Pvt Ltd, Chennai (Hiring product - http://piqube.com) – Consultancy on Design and a Java based minimal Implementation for computing the various statistical measures from a job experience sequence: 1) Stability or predictability of job experience sequence using Shannon Entropy and Inversion number, 2) Estimating various types of average experience metrics, 3) Mobility scoring, 4) Initial work on Skills based clustering of linkedin profiles using String similarity and 5) Initial Hidden Markov Model design for job change motivation inference

---

February 2014 – Clockwork Interviews Pvt Ltd, Chennai (Hiring product - http://piqube.com) – In-office non-profit Consultancy to PiQube for initial design of a Multivariate Linear Regression model for Stability and Gap scoring in Resume that includes entropy independent variables using ENT sequence disorder measure, study of various tools (including Stanford CoreNLP - Classifiers and Recursive Tensor Neural Network Sentiment Analysis Tools) for Sentiment Analysis of a Resume and Holistic Resume scoring model using social network analysis from multiple datasources

---

October 2013 – January 2014 – continued work on my opensource products (Krishna iResearch - https://sourceforge.net/users/ka_shrinivaasan) and non-profit machine learning remote consultancy to Clockwork Interviews Pvt Ltd, Chennai (Hiring product - http://piqube.com) for initial design of a Resume scoring, Named Entity Recognition in Resume using GATE, CRF++, IIT-Bombay CRF, study of Social Network Analysis Tools (SocVnet,SNAP etc.,) and informal mail interactions on my open source products among other things

---

September 2012 - February 2013 – Parttime Consultancy for Global Analytics(Chennai) and Work on my opensource products (Krishna iResearch)

1. Did the **Python AMP workflow implementation** as per the workflow specification for AMP using PyUtilib, RabbitMQ and Config files support. Wrote some python scripts for AMP workflow config files support and parsing .
2. Wrote Design spec analyzing workflow alternatives (like PyF, cascade) .
3. Wrote **AMP workflow-over-cloud** (AMP on Hadoop) specification in addition to AMP workflow spec.
4. Guided on GDP 2.7.x and 3.0 bug fixes and issues.
5. Helped in debugging GDP 3.0 timeout manager implementation in Feb 2013 which was found to be working.

Apart from the above worked on design and development of my nonprofit open source products- https://sourceforge.net/users/ka_shrinivaasan

January 2011 – March 2011 and July 2011 – August 2012 – Global Analytics(Chennai) – Senior Lead Software Architect (C/C++/Python)

1. Mentored and Managed a team of 3 people .

2. Released version 2.3.1 of Global Decision Platform(GDP) with fixes for crashes in production environment.

3. Did refactoring, many bug fixes and enhancements to Global Decision Platform(GDP)

version 2.5.0 for **master demo lead passthroughs** to work.

4. Implemented **acceptor-worker thread model** request handling in GDP 2.5.0 with single acceptor thread and configurable number of worker threads for Service Execution Manager(SEM) and Service components of GDP. This resulted in significant improvement in response time.

5. Designed and implemented **session based request-response** in GDP 2.5.0

6. Implemented modified timeout error logging with placeholders for errorcodes and request id and did fixes for frequent crashes in production environment for GDP 2.5.0.

7. Released GDP version 2.5.1 for which 2 minor binaries were developed to clear the queue and test the creation of queues (for System V message queues)

8. Released GDP version 2.7 which has important bug fixes for crashes in python container(needed for AMP runtime described later), few features for **graceful shutdown** of the SEM, option to reconnect to MySQL during exception in Python container etc.,

9. Redesigned GDP 2.5.0 **SEM loadbalancer for GDP 3.0** to route requests to service processes by periodically monitoring the service process load and removing a bottleneck due to wait/notify code. This resulted in 10x speed up of the request-response throughput time.

10. For GDP version 3.0, designed and Implemented a **new Session Timeout Manager** algorithm to timeout the request session based on user configured timeout value(simplified version of Survival Index Based Transaction Timeout Manager mentioned later).

11. For GDP version 3.0, worked on embedding **multiple python interpreters** in Service process with Python C API replacing boost::python calls. This involves facility to have a global dictionary across multiple interpreters (or) local dictionary per request (or) per-thread dictionary for each service worker thread (using thread local storage) and also creation of pool of interpreters from which interpreters are allocated per request and returned to the pool after the request is serviced. This obviated the need for boost::python since boost did not support multiple interpreters with multithreading and interpreter pooling. This involved fixing one of the most time consuming bug related to restricted mode python in 2.4.3. Since it was deprecated in python 2.7, fixed Python 2.4.3 source to circumvent restricted mode and also did another fix for the same problem with PyImport_ImportModuleEx()

12. Mentored the team to work on Dynamic Risk Tables (DRT) library of GDP to extend the database compatibility to MySQL and Oracle

13. Was involved in the Automated Modelling Platform(AMP) development to automate the predictive modelling from the production data. Fixed many crashes in python runtime during testing of AMP prototype realtime design. Suggested an alternative design for AMP realtime based on external user address-space message queue, which populates the in-memory queue from on-disk request log, periodically replenishes it from disk, reads requests from message queue and posts them in parallel to Realtime-subset and Realtime-superset python services, and gives a virtual view of the single physical queue to each queue client(queue clients are located in SEM) – meant to replace an existing queueing module with msgsnd/msgrcv which involves user and kernel address-space copying to remedy huge queue backlog seen in AMP realtime prototype. This tightly couples GDP Python runtime with the AMP runtime

14. For GDP 3.0, implemented a GDP pluggable queue named **GDAMPQ** designed for AMP above – Global Decisioning and Automated Modeling Platform Queue which is a simple in-

memory queue with optional disk persistence to replace an existing queueing module with msgsnd/msgrcv which involves user and kernel address-space copying. This in-memory queue in best testing circumstances is 30% better than the System V unix message queues.

15. Worked on Python 2.4.3 source code to analyze crashes in PyMalloc. Modified Python 2.4.3 source to add a flag DISABLEPYMALLOC to disable calls to PyMalloc and re-route them to unix malloc/free systemcalls. Also analyzed Python 2.4.3 source code for **restricted mode related** errors while designing multi-interpreters for GDP 3.0 and fixed them.

16. Worked on writing python configuration scripts for **automating the build configuration and installation** of GDP components like SEM, Service and Master .

17. Implemented a minor **encrypt utility** for master component of the GDP

18. Wrote Functional specification for **GDP 3.0 Automated Installer design**

19. Wrote Functional specification for **AMP Workflow**

20. Wrote Functional specification for **AMP Cloud deployment**

21. Studied python workflow packages for AMP workflow viz., Pyutilib.workflow, PyF and implemented a workflow prototype for AMP workflow in Pyutilib.workflow.

22. Implemented a **small prototype stub using Dumbo Mapper-reducer** for Cloud Deployment of AMP

23. Did 2 one-week training sessions for GDP versions

24. Did one-week training session for Python.

25. Implemented a **mini-GDP client and server** for debugging socket related errors in production.

26. Developed a **documentation knowledgebase** intranet website for GDP documentation, design documents, installation troubleshooting documents and debugging documents.

27. Implemented a **Build Version feature** for GDP 2.7.1 and GDP 3.0 which dynamically generates a buildversion.h headerfile which contains #defines for build machine and architecture information,SVN source tree version information for the build and is built at runtime by SEM, Service and Master binaries which include this buildversion.h header file **(In August 2012 resigned for personal reasons.)**

January 2011 – March 2011 – Consultant for Global Analytics(Chennai) and Work on my opensource products (Krishna iResearch) – (C++/Python)

Worked on my Open Source Products Design & Development (Krishna iResearch) and Did consulting and development for Global Analytics,Chennai (optimization, a smart-pointer reference-counted memory manager backed-up by an Object Pool, and refactoring) in Global Decision Platform (GDP) 2.3 for GPD 2.3.1 and 2.5 which is written in C++ with boost::python embedding.

August 2008 – June 2011 – Work on my opensource products (Krishna iResearch)

Started working on open source project ASFER (http://asfer.sourceforge.net) as part of my non-profit open source initiative Krishna iResearch – ASFER is a rule miner and executer-presently uses Vector space retrieval and Support Vector Machines with added Support for Naive Bayesian Multinomial Classifier and Decision Tree Classifier

March 2006 – July 2008 - webMethods and webMethods (now Software AG) (Bangalore) (C/C++/Java)

Worked on WebMethods Broker Server 5.x/6.x/7.x– a heavily multithreaded messaging productwritten in C/C++/Java based on publisher/subscriber model – Clients can publish and/or subscribe to certain predefined types of messages called "Events". My responsibilities included development of new functionalities and fixing customer escalation issues in core areas of Broker (Connection Layer, Territories/Gateways,Broker Admin Tool etc.,) and writing knowledgebase documents.

November 2005 – February 2006 – Krishna iResearch (self-started, not-for-profit, open-source research initiative)

Initial design work for Krishna iResearch open source products focussing on algorithms for web and BigData

August 2005 – November 2005 – Verizon Data Services India, Chennai

In Verizon India for brief period and then self-study in Mathematics and Computer Science.

May 2003 – Krishna iResearch (self-started, not-for-profit, open-source research initiative)

Registered on SourceForge.net and Initial design work started for open source products focussing on algorithms for web and BigData

February 2000 – July 2005 – iPlanet (Sun Microsystems-Netscape Alliance) and Sun Microsystems (now Oracle) (Bangalore)

Ported SunONE Application Server to Red Hat Enterprise Linux Advanced Server 2.1 (2005)

Certified JWSDP 1.3 webservices pack on Sun One Application Server 7.1 (2005)

Studied feasibility of supporting SunOne Web/Proxy Server on Solaris 10 zones (2005)

On Academic Sabbatical for 3 months (February 2005-May 2005)

Implementation of a threaded ICP server and porting SOCKS server for Sun Java system web proxy server 4.0 (C/C++/Solaris/Windows/RHELinux) (2004-2005):
Implemented a threaded ICP(Internet Cache protocol) server functionality using NSPR threads.
Also ported a legacy SOCKS server to proxy server 4.0.

Load Balancer Module for Apache Web Server 1.3.27 / 2.0.47 (C/C++) (2003-2004) :

Designed and implemented a Load balancer module for Apache 1.3.27 which will route the HTTP/HTTPS requests onto SunONE Application Server instances by Round Robin Algorithm.

This module is part of Sun ONE Application Server 7, Enterprise Edition.Also ported the module to Apache 2.0.47.

Optimizations and Features for iPlanet/Sun ONE Application Server 6.x (Java) (open sourced at http://glassfish.java.net) (2002) :

a) Fixed bugs related to performance, admin-tool and security in iPlanet Application Server 6.x. Also fixed few CTS bugs in SunONE Application Server 7 Standard Edition.

b) Involved in optimization of SUN J2EE Reference Implementation (RI) to make the JTS transaction manager enterprise-class. This included introducing a new Component Level Transaction Attribute setting feature in iPlanet Application Server 6.5. With this, user can set the type of transaction (XA or Local) at J2EE application component level. [Patents granted – details in the end]

c) Added a Trace functionality for the JTS transaction manager in SunONE Appserver 7.0 Enterprise Edition (http://glassfish.sourcearchive.com/documentation/1:2ur2-b04-1/dir_512e7fccb5ab465c594742cd72317a0e.html).

Debugging and Monitoring framework (MagicDraw) (2002) :

Designed a Debugging and Monitoring framework for Sun ONE Application Server 7.0 Enterprise Edition.

Store Adapter prototype for High Availability(Java) (2002) :
Implemented a store adapter prototype for Clustra High
Availability Database. This was to test the feasibility of using
Clustra HADB as backend store for session data in SunONE Application Server 7.0
Enterprise Edition.

Minor Feature Addition to SunOne Application Server 6.5 SP1 (2002)
Added support for enabling/disabling TCP_NODELAY in SunOne Application Server 6.5
SP1

Survival Index based Transaction Timeout Manager for iPlanet Application Server (Java) (2002):
Designed and implemented a new Transaction Timeout Manager for iPlanet Application
Server.This resulted in an overall speed-up of 10% for the application server. (Invention
Disclosure done in 2002-2003)

Miscellaneous Bugfixes for customer escalations in the BillerXpert product which was dependent on iPlanet Application Server. This was one of the most involved bugfixes that required many days of debugging with JVM Profiling tools. (2001)

**An automated Test suite for nightly build setup of
iPlanet Application Server 6.0 (2001):**
Designed and implemented a test suite which ran a set of regression tests on nightly builds
of iPlanet Application Server - written on Korn shell.

JVM related bugfix for IBM's Java Virtual Machine team related to native thread and green thread implementations of JVM (2000)

Created a Resume Querying System using JDBC with Oracle Backend for Sun Microsystems Human Resources Department to track candidate profiles – named ResumeXpert (2000)

Official Training Courses (SunU)

1. Solaris internals
2. Multithreaded Application development in Solaris
3. SOLARIS DEVICE DRIVER INTERNALS

July 1999 - January 2000 - BaaN Infosystems (now SSA Global) (Hyderabad)

Developed a storefront for an e-commerce application called
E-Enterprise using Microsoft Internet Information Server/Site Server, Visual Basic, Active Server Pages and Visual Interdev

January 1999 – May 1999 – BE Thesis and Project(team) – COBRA – Distributed Computing Framework based on CORBA(Visibroker) and JAVA.

April 1998 – June 1998 – BE Internship at Steel Authority of India Limited (Raurkela and Durgapur Steel Plants) – Enhancements to Automated Furnace Data Acquisition software through a GUI developed on VAX Fortran over VAX

VMS

MISCELLANEOUS

1. Gold medal for proficiency from PSG Tech for ranking first in B.E.(CSE) – Received in February 2000
2. GRE (2002) – score 2040
3. Joint Entrance Screening Test(JEST) 2006 in Theoretical Computer Science - Rank 21 (Reg No: 22123).
4. Participated in Winter School 2010 on Machine Learning and Computer Vision, Microsoft Research & CIFAR, IISc
5. UGC Net July 2016 – 66 (I), 54 (II), 50.67 (III) (in percentage) (Roll No. 69004241)

INTERESTS

Writing, reading non-fiction mostly on science,mathematics, religion and philosophy.