# March 10, 2015

|  |  |
| --- | --- |
| Jay Mehta | jay87.mehta@gmail.com |
| Python Developer | +91 9987 67 5857 |

Career Summary

I am a *Python developer* with rich experience in designing and development of applications, Web Apps, Web framework and databases. Possess strong expertise on Linux system and shell scripting. Have good knowledge of client server model, strong OOPs and C/C++ skills.

Skill Summary

|  |  |
| --- | --- |
| Total Experience | 6 Years 6 months |
| Current Designation | Technical Lead |
| Design | Design Patterns, UML, Object Oriented Analysis and Design |
| Computer Languages | Primary: Python, Django  Secondary: C/C++, Linux Shell Scripting, Java, HTML, CSS, SQL, PL-SQL |
| Software Development Methodology | Agile Scrum, RUP |
| Tools | StarUML GDB, Multi, GCOV, LCOV, Valgrind, Wireshark |
| Source versioning & revision control tools | Clearcase, SVN |
| Bug Tracking and reporting tool | Clearquest, Bugzilla |
| Operating Systems Knowledge | Linux |
| Databases Worked On | Oracle 9i & 10g, PostgreSQL |
| Educational Qualifications (Month and Year of Passing) | 1. Bachelor of Engineering in Information Technology (May 2008) Agg Per: 65.02% 2. Higher Secondary Certificate (February 2004) 3. Secondary School Certificate (March 2002) |

Work Experience

IGATE Global Solutions Ltd – 35 months (27th Feb, 2012 – till date)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Project Name | Platform API(PAPI) | | |
|  | Name of the Client | EMC Corporation | | |
| Description of the project | | | |
| OneFS is a FreeBSD base, scale-out NAS file system from Isilon, now a part of EMC Corporation. PAPI is implemented as Fast CGI daemon in C++, using libfcgi. PAPI provides set of RESTful web interfaces, or handlers, used to query, configure, and administer OneFS. The OneFS CLI and WebUI are clients for PAPI, and do their actual configuration through it.  This project involves implementing network related CLI commands in **Python** and PAPI handlers in C++. | | | |
| Role in the project | | : | Programmer ( Development & Unit Testing) |
| Tools /Frameworks / Methodologies used in the project | | : | 1. Python, C++  2. GCC & Make  3. SVN |
| Responsibility | | : | 1. Implementing new CLI commands to existing framework, in **Python** 2. Implementing PAPI handlers (Rest interfaces) to accept CLI command requests, serve and respond 3. CLI command testing using **Unittest** module of Python 4. Writing unit testing for handlers in Python **nosetest** |
| Duration of the Project | | : | From : Aug 2014 To : till date |
| Hardware | | : | OneFS VM cluster |
| Software | | : | SVN |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | Project Name | IPv6 support | | |
|  | Name of the Client | EMC Corporation | | |
| Description of the project | | | |
| OneFS is a FreeBSD base, scale-out NAS file system from Isilon, now a part of EMC. This project involves working on all modules and features of the system, modifying code in order to make it IPv6 compliant. | | | |
| Role in the project | | : | Programmer ( Defect fixing & Testing) |
| Tools /Frameworks / Methodologies used in the project | | : | 1. C, C++, Python  2. GCC & Make  3. SVN  4. Bugzilla |
| Responsibility | | : | 1. Understanding scale-out NAS *OneFS* file system and IPv6 2. Analyze bugs and provide code fixes 3. Modify code in C, C++ to enhancement product to support IPv6. Update CLI in **Python** language to support modifications 4. Convert List files to **Python** based testing framework. Perform Unit Testing, Smoke testing, File System Regression testing 5. Configure NAT64 software solutions Tayga along with BIND for DNS, allowing communication between IPv4 and IPv6 infrastructures 6. Customized open source IP tunneling software Miredo so IPv4 infrastructure can access IPv6 servers |
| Duration of the Project | | : | From : March 2014 To : Aug 2014 |
| Hardware | | : | Network attached system in IPv6 environment |
| Software | | : | SVN, Bugzilla, FreeBSD OS, OneFS virtual machine, VMWare workstation |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Project Name | iScale, a Scale-out NAS solution | | |
|  | Name of the Client | IGATE | | |
| Description of the project | | | |
| This C based storage project is a distributed file system that will run on the Network Attached Storage (NAS) boxes providing a unified namespace. The communication between this Storage boxes is implemented using SUN RPC protocol. The product provides an admin purpose CLI & web management interface to configure Storage boxes, check their status, disk usage, dynamically add and remove Storage boxes, rebalance the disk content for equal distribution. | | | |
| Role in the project | | : | Designer & Programmer (Requirement Analysis, Code design and implementation, Unit Testing & defect fixing) |
| Tools /Frameworks / Methodologies used in the project | | : | 1. **C, Python & Django, Ctypes**  2. GCC & Make  3. SUN RPC & RPCGEN  4. FUSE (File system in User Space framework)  5. Libconhash (consistent hashing library)  6. GDB, Valgrind, GCOV |
| Responsibility | | : | 1. Understanding requirement, scope and designing scale out NAS solution(distributed file system, scale out features) 2. Implementation of scale out NAS, running on Linux OS Storage boxes written in C 3. Distributed communication using SUN RPC with help of RPCGEN compiler 4. Command Line Interface development in C. 5. Design and development of web based admin utility in **Python** using **Django framework**. Used Python **CTYPES** to communicate with the backend. 6. Unit Testing and defect fixing 7. Performance benchmarking using IOzone 8. Glusterfs installation for performance comparison |
| Duration of the Project | | : | From : Mar 2012 To : Feb 2014 |
| Hardware | | : | Network attached system |
| Software | | : | Eclipse, SSH-Client, GDB, SVN, EMACS, GCC & MAKE, SUN RPC, Django, Linux OS, IOzone |

Wipro Technologies - 26 Months (30th Nov, 09 – 20th Feb, 12)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Project Name | ALU-FTTU-OLTSW | | |
|  | Name of the Client | Alcatel-Lucent, USA | | |
| Description of the project | | | |
| Fiber to the User is the technology to bring the triple play services to the end user over the Optical Cable Fiber. The project involves client’s hardware called OLT (Optical Line Termination) and ONT (Optical Network Termination). To enable/disable features at the end user side equipment ONT, several commands are executed at the server side OLT. These commands are implemented in TL1 language/CLI and C++. | | | |
| Role in the project | | : | Programmer |
| Tools /Frameworks / Methodologies used in the project | | : | 1. IBM Rational ClearCase tool for SCM 2. C++ and TL1 language for implementation of TL1 based commands 3. MULTI code debugger 4. Wireshark tool to send and capture network packets 5. Cygwin |
| Responsibility | | : | 1. Implementation and porting of TL1 commands in TL1 language and C++ on Solaris 2. Defect fixing. Debugging using MULTI debugger and Wireshark tool to capture network packets traces 3. Test case preparation, reviewing and execution on different OLT and ONT cards 4. Maintenance activities like branch merging using IBM Cleartool SCM, defect reproduction and bug fixing |
| Duration of the Project | | : | From : Nov 2009 To : Feb 2012 |
| Hardware | | : | GLT2-A, GLT2-B, GLT4-A Alcatel GPON line termination cards, Fiber cables, Broadcom ONT, Data only card ONT, |
| Software | | : | MULTI, GCC, Cygwin, Wireshark, Cleartool |

Homi Bhabha Centre for Science Education, a Tata Institute of Fundamental Research - 17 Months (10th July 08 - 22nd Nov 09)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Project Name | SELF-Platform | | |
|  | Name of the Client | Gnowledge Lab at Homi Bhabha Centre for Science Education | | |
| Description of the project | | | |
| SELF Platform is a web-based, collaborative and distributed authoring system for learning materials open for all to share knowledge. The platform is based on **Python, Zope**, and GNOWSYS. http://www.selfplatform.eu/ | | | |
| Role in the project | | : | Programmer (Detailed Design and Development) |
| Tools /Frameworks / Methodologies used in the project | | : | 1. Server implementation languages: Python, Zope Page Template and DTML of Zope Server 2. Front end development languages: XHTML, Javascript, XML, CSS, DHTML, AJAX 3. PostgreSQL database server |
| Responsibility | | : | 1. XMLfile parsing for data extraction to load in SELF-Platform application. 2. Client side implementation: 3. Dynamic form creation for add, edit and delete operation for Content Manager Module using tiny-MCE editor and HTML DOM. 4. Content viewer for Content Manager module using Ajax and JS 5. Content organizer in Organizer module using JavaScript 6. Server side implementation: 7. Configuring User Authentication plug-in SQLPass Plug-in with ZOPE server 8. HTML Page generation using GNOWSYS store, Python, Zope Page Template 9. Storing and retrieving content versions for Version Manager Module using GNOWSYS APIs, python |
| Duration of the Project | | : | From : July 2009 To : Nov 2009 |
| Hardware | | : | N/A |
| Software | | : | Zope Server, Plone CMS, PostgreSQL database server, GNOWSYS, Ubuntu OS |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | Project Name | GNOWSYS | | |
|  | Name of the Client | Gnowledge Lab at Homi Bhabha Centre for Science Education | | |
| Description of the project | | | |
| GNOWSYS, Gnowledge Networking and Organization System, is a distributed knowledge network management system. It is a free software web application where generic classes viz. metatype, relation type, object type, object, relation, attribute, etc., are provided for developing and maintaining semantic web. It works as an installed product in Zope. | | | |
| Role in the project | | : | Programmer (Detailed Design and Development) |
| Tools /Frameworks / Methodologies used in the project | | : | 1. Server implementation languages: Python, Zope Page Template and DTML of Zope Server 2. Front end development languages: XHTML, Javascript, XML, CSS, DHTML, AJAX 3. PostgreSQL database server |
| Responsibility | | : | 1. Concept learning of GNOWSYS 2. Database Schema designing 3. Implementation in python programming language with PostgreSQL back end 4. Unit Test Cases using unittest module of Python 5. Writing bash shell script to automate GNOWSYS installation with its dependencies (Plone and Zope) 6. Writing GNOWSYS APIs for applications that will use GNOWSYS as backend. |
| Duration of the Project | | : | From : Nov 2008 To : July 2009 |
| Hardware | | : | N/A |
| Software | | : | Zope Server, Plone CMS, PostgreSQL database server, Ubuntu OS |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Project Name | WET | | |
|  | Name of the Client | Gnowledge Lab at Homi Bhabha Centre for Science Education | | |
| Description of the project | | | |
| Word Encoding Technique is an application written in python programming language with PostgreSQL database handler that performs compression of data by assigning least integer number to most frequently used words. | | | |
| Role in the project | | : | Programmer (Detailed Design and Development) |
| Tools /Frameworks / Methodologies used in the project | | : | 1. Python Language for implementation 2. PostgreSQL database server |
| Responsibility | | : | 1. Implementation of an encoding framework WET 2. Writing scripts for downloading Wikipedia pages and performing words extraction for collecting word frequency 3. Database schema designing and implementation in PostgreSQL |
| Duration of the Project | | : | From : July 2008 To : Nov 2008 |
| Hardware | | : | N/A |
| Software | | : | EMACS, Python, PostgreSQL database server, Ubuntu OS |

Personal Details

|  |  |
| --- | --- |
| Date of Birth | 25/03/1987 |
| Marital Status | Unmarried |
| Nationality | Indian |

Technical Participation

* Qualified Google Code Jam 2009, 2010, 2011, 2012, 2014

Hobbies

* Sports
* Photography