

ILLYOUNG CHOI

Mobile: (+1) 520-461-9666 | Email: iychoi@cs.arizona.edu | GitHub: <https://github.com/iychoi>

PROFILE

A Ph.D. candidate in Computer Science with 6 years of experience in IT industry as a software engineer. Built experiences on software design and development in various platforms. Pursuing a career in a cloud computing sector to solve real world problems.

SUMMARY

- 6 years of experience in IT industry with software design and development experience
- Development experience in various platforms and languages
- Featured Projects:
 - Stargate: A file-system for remote data access between Hadoop clusters
 - Syndicate: A virtual cloud storage
 - iRODS CSI Driver: A Kubernetes persistent storage driver for iRODS
 - iRODS-FUSE: A FUSE implementation of iRODS file-system client
 - Libra: A Hadoop-based metagenomic sequence analysis tool
 - Nabi: A web widget runtime & IDE for smart phones and set-top boxes
 - Moon Framework: A framework for feature phone application
 - Moon Rabbit: A light-weight web-browser for feature phones

PROFESSIONAL EXPERIENCE

- Worked as a **Summer Intern (Software Engineer)** at **Open Networking Foundation**, 2019
- Worked as a **Software Engineer** at **Minigate**, 2010-2012
- Worked as a **Software Engineer** at **Entaz**, 2007-2009
- Worked as a **Software Engineer** at **ENSI Information Technology**, 2005-2007
- Worked as a **Software Engineer** at **AT Solution**, 2004-2005

EDUCATION

- A Ph.D. candidate in **Computer Science, University of Arizona (US)**, 2014-Present (Expected graduation date: Fall 2020)
- Received Master of Science in **Computer Science, University of Arizona (US)**, 2014
- Received Bachelor of Engineering in **Computer Science and Engineering, Hankuk University of Foreign Studies (South Korea)**, 2009

ACADEMIC EXPERIENCE

- Work as a Research Assistant for a project "iRODS CSI Driver: A Kubernetes persistent storage driver for iRODS", Edwin Skidmore and John Hartman, 2020-Present
- Worked as a Research Assistant for a project "Stargate: Remote Data Access between Hadoop Clusters", John Hartman, 2017-2020
- Worked as a Research Assistant for a project "Libra: A Comparative Metagenomic Sequence Analytic Tool using Hadoop", Bonnie Hurwitz and John Hartman, 2014-2018
- Worked as a Research Assistant for a project "Tools and data-driven discovery platform for the microbiome sciences", Bonnie Hurwitz and John Hartman, 2017-2018
- Worked as a Research Assistant for a project "Syndicate: A Virtual Cloud Storage", Larry Peterson and John Hartman, 2013-2017

- Worked as a Research Assistant for a project “iRODS-FUSE: A FUSE Implementation of iRODS File-system Client”, Nirav Merchant and John Hartman, 2014-2016

PUBLICATIONS & PRESENTATIONS

- A conference paper, “Stargate: Remote Data Access between Hadoop Clusters”, ACM SAC, 2021, in submission
- A user group meeting talk, “iRODS Container Storage Interface (CSI) Driver”, TRiRODS, 2020
- A conference paper, “SDM: A Scientific Dataset Delivery Platform”, e-Science, 2019
- A conference talk, “A Workflow Management Engine in CORD”, ONF Connect, 2019
- A journal paper, “iMicrobe: Tools and data-driven discovery platform for the microbiome sciences”, GigaScience, 2019 (co-authored paper)
- A journal paper, “Libra: scalable k-mer-based tool for massive all-vs-all metagenome comparisons”, GigaScience, 2018
- A workshop paper, “Libra: Improved Partitioning Strategies for Massive Comparative Metagenomics Analysis”, ScienceCloud, 2018

SKILLS

- Programming Languages: C, Java, Python, Javascript, Golang
- Platforms: Hadoop, Ignite, Lucene, Airflow, Node.JS
- Database Systems: MS SQL, MySQL
- DevOps: Docker, Ansible, Kubernetes

PROJECT DETAILS

iRODS CSI Driver: A Kubernetes persistent storage driver for iRODS File-system, Research Project, 2020-Present

iRODS is an open-source grid file-system used by several science clouds, such as Texas Advanced Computing Center (TACC) and Cyverse. iRODS CSI Driver allows Kubernetes users to mount iRODS collections on a directory hierarchy in Kubernetes Pods as persistent volumes. The driver supports iRODS access via three different iRODS Clients (iRODS-FUSE, NFS and Davfs2). The driver implements both dynamic and static volume provisioning modes. The driver will be used in Cyverse Discovery Environment.

Responsibilities:

- Design and implement the driver

Technologies:

- Language: Golang, Python
- Communication Interface: Container Storage Interface
- Environment: Kubernetes, Docker

Source code:

- <https://github.com/cyverse/irods-csi-driver>

CORD Workflow Engine, Summer Intern at Open Networking Foundation, 2019

This work was to develop a workflow engine in CORD. CORD (Central Office Re-architected as a Datacenter) is an open reference implementation of an operating platform for service provider networks. The workflow engine enables rapid development and flexible deployment of workflows. Also, the engine enforces workflows to have clear execution flows for monitoring. The engine was implemented on top of Apache Airflow.

Responsibility:

- Design and implement the system

Technologies:

- Language: Python, Javascript (Node.JS)
- Communication: Socket.IO, Kafka
- Workflow management platform: Apache Airflow

Source code:

- <https://gerrit.opencord.org/gitweb?p=cord-workflow-probe.git;a=summary>
- <https://gerrit.opencord.org/gitweb?p=cord-workflow-controller-client.git;a=summary>
- <https://gerrit.opencord.org/gitweb?p=cord-workflow-controller.git;a=summary>
- <https://gerrit.opencord.org/gitweb?p=cord-workflow-airflow.git;a=summary>

Stargate: Remote Data Access between Hadoop Clusters, Ph.D. Dissertation, 2018-Present

This work is to develop a file-system for scientific computing that enables efficient remote data access between Hadoop clusters. Stargate performs parallel data transfer to make full use of cluster's total bandwidth. Stargate also provides multi-tiered data caching to improve data reusability and minimize WAN traffic. Stargate provides an integration to Hadoop to be used by existing scientific computing applications. Lastly, Stargate co-locates computation, caching, and data transport for efficient remote data access.

Committee:

- John Hartman, Larry Peterson, Beichuan Zhang, and Michelle Strout

Responsibility:

- Design and implement the system

Technologies:

- Language: Java
- HTTP/REST service: Eclipse Jetty, Eclipse Jersey, Jackson
- Distributed caching and messaging: Apache Ignite
- Storage integration: HDFS
- Compute framework integration: Apache Hadoop
- Cloud service: Amazon EC2, Google Compute Engine, Amazon VPC Flow Logs, Amazon S3
- Service deployment: Ansible, Boto3 (for Amazon)

Source code:

- <https://github.com/iychoi/stargate-commons>
- <https://github.com/iychoi/stargate-server>
- <https://github.com/iychoi/stargate-hdfs-client>

Libra: A Comparative Metagenomic Sequence Analytic Tool using Hadoop, Research Project, 2014-2018

This work was to develop a comparative metagenomic sequence analytic tool using Hadoop (MapReduce). Libra performs sequence comparisons between samples using k -mers (as known as k -grams). Libra determines similarity (or dissimilarity) of the samples using statistical metrics such as Cosine-Similarity, Bray-Curtis, and Jenson-Shannon. Considering ever growing metagenomic sequence data volume, Libra was designed to process any size of input. Libra calculates large metagenomic sequences effectively using Hadoop with several techniques - histogram-based partitioning for load-balancing, a distance matrix computation by a scan-line algorithm, and total-order partitioning for inverted index construction. Community access to Libra is provided via iMicrobe portal.

Responsibilities:

- Design and implement the tool

- Setup Hadoop clusters for service

Technologies:

- Language: Java
- Compute framework: Apache Hadoop (MapReduce)

Source code:

- <https://github.com/iychoi/libra>

Websites:

- iMicrobe portal: <https://www.imicrobe.us>
- Hurwitz lab: <http://www.hurwitzlab.org>

SDM (Syndicate Dataset Manager): A Scientific Dataset Delivery Service, Research Project, 2018

This work was to develop a scientific dataset delivery service using Syndicate. SDM cherry-picked Syndicate's features – data acquisition from external storages, user interfaces and multi-tiered caching. SDM provides a simple command-line admin interface for search and mount of datasets. SDM mounts datasets on a directory hierarchy of Linux filesystem or Hadoop filesystem. Therefore, SDM allows scientific workflows to access remote datasets without laborious data staging. The work was to allow efficient and convenient data access between Cyverse in Arizona and Texas Advanced Computing Center in Texas.

Responsibilities:

- Design and implement the service
- Manage exemplary external datasets and their acquisition gateways

Technologies:

- Language: Python, Node.js
- REST service: Express
- Database: SQLite3
- Service deployment: Ansible, Docker

Source code:

- <https://github.com/syndicate-storage/syndicate-dataset-manager>
- <https://github.com/iychoi/sdm-catalogue-service>
- <https://github.com/syndicate-storage/public-dataset-ag>
- <https://github.com/syndicate-storage/data-containers>

Syndicate: A Virtual Cloud Storage, Research Project, 2013-2018

This work was to develop a general-purpose data storage service using existing storage components for a global, scalable, and secure storage service. These include cloud storage for data durability, network caches and content distribution network for scalable read bandwidth, and local disks for local read/write performance. Syndicate's goal was to allow applications to access data independent of where it is stored by maximizing the use of commodity infrastructure. In this project, I implemented various Syndicate gateways to integrate external storage and compute platforms.

Responsibilities:

- Implement acquisition/replica gateways for iRODS and Cyverse DataStore
- Implement user gateways for computing platforms
 - Hadoop filesystem (HCFS)
 - HTTP/REST service
- Implement some features for large data access
 - Prefetching and connection reuse
- Test and debug the system

Technologies:

- Language: C, Python, Java, Javascript
- External storage integration: iRODS, Cyverse DataStore
- Compute framework integration: Hadoop
- Language binding: Javascript via Node-FFI, Java via REST
- REST service: Express
- Deployment: Ansible, Docker

Source code:

- <https://github.com/syndicate-storage>
- <https://github.com/iychoi/iPlantBorderMessageServer>

Website:

- <http://www.syndicate-storage.org>

iRODS-FUSE: A FUSE Implementation of iRODS File-system Client, Research Project, 2014-2016

iRODS is an open-source grid file-system used by several science clouds, such as Texas Advanced Computing Center (TACC) and Cyverse. iRODS-FUSE is a client app that allows users to mount iRODS collections on a directory hierarchy of Linux/Mac filesystem. This work was originally to improve data access performance of existing implementation (version 3.x). However, I found many data corruption bugs and design issues that caused performance degradation, so I redesigned the client. The new design implements buffered I/O, in-memory data/metadata caching, parallel data transfer for read and prefetching. The new design solved the bugs and issues. It showed improved read (68%) and write (91%) performance compared to the old implementation (version 3.1). The new design was accepted by iRODS consortium and officially included from a release version 4.1.4.

Responsibilities:

- Design and implement the client

Technologies:

- Language: C
- File-system interface: FUSE
- Development environment deployment: Docker

Source code:

- https://github.com/irods/irods_client_fuse
- <https://github.com/cyverse/cyverse-irods-dev-docker>

Website:

- <https://irods.org/>
- <https://cyverse.org/data-store/>

BioSpectra, Research Project, 2015

This work was to develop a classification system for metagenomics reads. BioSpectra determines the taxonomy of given metagenomic reads using k -mer-based search. BioSpectra's goal was to identify the microbiome composition of a sample. Considering ever growing metagenomic sequence data volume, BioSpectra performed in parallel and implemented several optimizations that reduce the number and size of search queries. However, the project was not successful due to poor performance.

Responsibilities:

- Design and implement the system

Technologies:

- Language: Java, Python
- Search engine: Lucene
- Database: SQLite
- Message queue: RabbitMQ

Source code:

- <https://github.com/iychoi/biospectra>

H-Synthesizer: Metagenomic Sequence Analysis using Hadoop and Syndicate, Master Thesis, 2013-2014

This work was to develop an integrated system for large-scale metagenomic sequence analysis. The system used Hadoop for scalable computation and Syndicate for data transfer. Using Syndicate, Hadoop computation was able to access remote datasets in iRODS storage transparently. The work consisted of three sub-projects: 1) development of a metagenomic data analysis app, 2) development of a Hadoop file-system plugin for Syndicate, and 3) implementing parallel data transfer for iRODS FUSE client. These sub-projects became separated projects later -- Libra (analysis app), H-Syndicate (a Hadoop integration of Syndicate) and a new design of iRODS-FUSE.

Committee:

- John Hartman, Larry Peterson, David Lowenthal

Responsibilities:

- Implement the genomic data analysis app using Hadoop
- Implement the Hadoop file-system plugin for Syndicate
- Implement the Syndicate Acquisition Gateway for iRODS
- Optimize data read performance of iRODS-FUSE

Technologies:

- Languages: Java, C, Python
- External storage service integration: iRODS
- Compute framework integration: Hadoop

Nabi: A Web Widget Runtime, Minigate, 2011

This work was to develop WAC/W3C compliant web widget runtimes for Android phones and set-top boxes (STB). A W3C web widget is an app written in web languages (HTML, CSS and Javascript). W3C extends W3C web widget specification to provide device access (e.g., access to camera, contacts, or accelerometer). In this project, I researched a web-browser engine (Webkit) and Javascript device APIs.

Responsibilities:

- Research Webkit and WAC/W3G Web Widget Specification
- Design and implement a prototype of Javascript device APIs

Technologies:

- Language: C/C++, Javascript
- Web browser engine: Webkit
- Device APIs implementation: Webkit plugin-architecture (NPAPI)

Nabi IDE: A Web Widget IDE, Minigate, 2012

This work was to develop an Integrated Development Environment (IDE) for development of WAC/W3C compliant web widgets. The IDE was based on Eclipse and was implemented as an Eclipse plug-in. The IDE provided Javascript code-assist features using JSDT, WAC/W3C compliant widget packaging and a configuration editor. The IDE also provided an emulator that enables testing of widgets. The emulator provided many small tools that mock device access to allow full emulation of widgets.

Responsibilities:

- Design and implement the IDE based on Eclipse
- Implement the emulator for testing and debugging

Technologies:

- Language: C, Java, Javascript
- IDE Development: Eclipse Plug-in, Eclipse JSDT
- Web browser engine: Webkit (Chromium Embedded Framework for Windows)

A Blood Collection Monitoring Software, Freelance Job, 2012

This work was to develop a monitoring software for blood collection devices. The software communicated to remote blood collection devices via radio frequency (with a non-standard protocol). The software managed registrations of blood collection devices and controlled blood collection processes.

Responsibilities:

- Design and implement the software

Technologies:

- Languages: C#
- Platforms: Windows, .NET Framework 3.0

Moon Framework: Mobile Application Framework, Entaz, 2008-2009

This work was to develop a framework for feature phone apps. The framework provided various APIs for network, graphics, and file-system for rapid app development. The framework also provided application updates and dynamic library loading. In this project, I led a small development team of 4 software engineers as a team leader.

Responsibilities:

- Design and implement the framework
- Develop compilation (GCC based) and packaging tools
- Analyze target platforms (SKT/KTF/LGT WIPI, BREW 3.x)
- Lead a team and manage project milestones

Technologies:

- Languages: C, Java, ARM Assembly
- Target platforms: SKT WIPI(C), KTF WIPI(Java/C), LGT WIPI(Java/C), BREW 3.x (C)

Moon Rabbit: Mobile Web Browser, Entaz, 2007-2008

This work was to develop a light-weight web browser for feature phones. The web browser supported HTML 3.0 and WML but had to maintain small executable size (~130KB). The browser provided features to minimize network transfers, such as icons (and emoji) and web page templates, because wireless data transfer over 2G network was very expensive. The browser also was able to run apps inside a web page like Adobe Flash. This feature was used to launch mini-games for a portal service, "Free Game Town" serviced by Korea Telecom (previously KTF). Entaz also released 6+ web-based apps using the web browser.

Responsibilities:

- Implement the web browser (rendering, networking, and caching)
- Implement an applet system

Technologies:

- Languages: C, Java
- Target platforms: KTF WIPI(Java/C), BREW 3.x (C)

Takamiya POS Management System, ENSI Information Technology, 2005-2007

This work was to develop a management software for POS (Point Of Sale) systems at Takamiya (a

fishing-equipment store chain). The software provided supply-chain management, warehouse management, product management, retail store management, price-tag printing, and sales statistics reporting.

Responsibilities:

- Design the system
- Implement warehouse management and product management
- Implement a HTML based report printing module
- Implement a barcode printing module
- Implement various statistics reports

Technologies:

- Language: C#
- Database: MSSQL

Mobile Apps, AT Solution, 2004

This work was to develop various feature phone apps such as banking, lottery, stock market, and real-estate apps. I implemented mobile apps using AT Solutions' mobile app framework. The framework provided networking and UI components. I developed 7 different services.

Responsibilities:

- Implement mobile apps

Technologies:

- Languages: C, Java
- Platforms: GVM (C-like), GNEX (C), SKT WIPI (C), SKVM (Java)

2020-09-01

ILLYOUNG CHOI