

David C Wang

CONTACT

San Francisco, CA
415.937.8385
davidwang at alum.mit.edu

<http://www.linkedin.com/in/dcwangmit01>
<https://github.com/dcwangmit01>
<http://davidwang.com>

OBJECTIVE

Full-time position in software engineering management or technical leadership.

Passionate about building software systems and applying methods that minimize operational burden. Engineer and manager with enough experience developing and operating 24/7 public cloud services to have learned endless empathy for operations and quality.

EDUCATION

Massachusetts Institute of Technology

Cambridge, MA

Bachelor of Science degree in Electrical Engineering and Computer Science (6-2).

EXPERIENCE

Architect, Engineering

Cisco Systems

San Francisco, CA

May 2015 – Current.

- Cloud Architect working on video distribution systems, enabling cable providers to stream live and video-on-demand content to any device on any network.
- Wrote code to manage and deploy Elasticsearch datastores running on Kubernetes, taking full advantage of auto-wiring, self-healing, and persistence. Litmus test for success is if the system survives any number of, combination of, or all node failures, and self-heal without intervention so long as a minimum number of Kubernetes nodes exist or are eventually returned to the cluster (Elasticsearch, Logstash, Kibana)
- Core contributor to Openstack Kolla-Kubernetes, a project to run Openstack services as containers on top of Kubernetes. Goal is to apply the self-healing and auto-wiring capabilities of Kubernetes to address the Openstack operations problem, because Openstack is hard to install, harder to operate, and nearly impossible to upgrade. As of August 2016 for the Openstack Newton release, became the top contributor for personally-written lines-of-code deletions and additions to the kolla-kubernetes project, as well as the top contributor from all of Cisco Systems to Openstack. Influenced the kolla-kubernetes project to remove host networking (thus enabling auto-wiring and self-healing) as well as create an operations-first focus by exposing all operable steps through CLI implementation. Refactored the CLI and implemented major features such as multi-node support, multi-node persistence, and support for various storage providers including GCE disks, AWS EBS, or custom Ceph RBD. (Docker, Kubernetes, Openstack, Ceph)
- Deployed and managed Kubernetes clusters in hosted GKE, turnkey AWS, as well as custom bare-metal with self-hosted Ceph RBD. Extensive experience with packet.net bare-metal as a service. (Bare-Metal, AWS, GKE, GCE)

Director, Engineering

Basis Science / Intel Corp

San Francisco, CA

September 2013 – May 2015. Basis, an Intel-acquired company, produces health & fitness wearables that continuously sample biometric signals including heart rate, perspiration, temperature, and motion. Our product is a top-selling wearable at Amazon and BestBuy.

- Managed the team responsible for the development and operations of the Backend Cloud Services, where all customer data ends up for storage, processing, and is made available through APIs.
- Within 5 months, resolved a series of scaling issues that effectively denied service for all customers. Within 11 months, we mitigated our #1 scaling bottleneck by migrating all time-series data away from our MongoDB cluster and into a Cassandra cluster. Our processing times have sped up by 100x due to the reduction in database write-lock wait time. We continue to migrate away from the operationally expensive MongoDB and are considering a traditional RDBMS system, which will be a better fit for how we reference our non-time-series data.
- Increased team happiness and retention by prioritizing infrastructure improvements that restored sustainable pace by improving system quality, maintainability, and reducing downtime.
 - Moved the team to re-implement our configuration and deployment system from basic Fabric scripts to a Python SaltStack automation system. Minor deployments no longer require downtime, and major deployments have reduced from 18 hours to 2 hours, meaning no more all-nighters for the team. Deployments are also occurring multiple times per week instead of once every 2 to 3 months. We have also reduced risk by modifying the infrastructure and process to support data snapshotting and

rollback (Cassandra and LVM snapshots). This brings us several steps closer to CICD, but we must focus next on continuous test to fill out the CICD pipeline.

- System monitoring previously relied only on Customer Support tickets. Moved the team to implement a Nagios monitoring system using network polling and agent-based metrics gathering that alerts through Pagerduty. The monitoring and alerting system traverses an escalation path within minutes if a system reboots, a service dies, or a network partition occurs. This significantly reduces team stress, catches production issues early, and increases team velocity by reducing the time we spend cleaning up production issues.
- My team generally implements and productionizes support for new features such as automatic Activity Detection (run, bike, walk), Sleep Staging Detection (REM, light, deep), and support of a new hardware device platforms.
- Implemented a rigorous and streamlined interview process, and hired over half of the team.

Manager, Engineering

Rackspace Hosting Inc.

San Francisco, CA

June 2012 – February 2013. Rackspace is a provider of cloud computing services.

- Responsible for the team and product behind the Rackspace Cloud Monitoring Service. Our 15 member engineering and operations staff developed and operated the highly redundant, distributed monitoring service which spans multiple data centers. Metrics gathering for the Cloud Monitoring Service consisted of distributed network pollers, as well as a multi-platform host-based agent. The collected time-series metric data is redundantly distributed to and processed at each data center, going through parallel processing pipelines, rules engines, and then finally archived in an eventually consistent Cassandra cluster. We used a multi-DC Zookeeper cluster for distributed locking, coordination, and de-duping the redundantly processed data before triggering notifications.
- Our development and deployment practice was nearly a complete CICD pipeline, which included continuous build, test, and semi-automated deploy. Code deployments to production occurred several times a day.
- Implemented new Jira Agile and Confluence instances in the team workflow, which enabled basic work tracking, prioritization, and documentation.

Manager, Engineering

Premier Retail Networks

San Francisco, CA

Senior Software Engineer

June 2005 – June 2012. Premier Retail Networks is the world's largest in-store video advertising network. Worked on all aspects of the network, including content distribution, playback, device control, monitoring, and reporting.

- Manager of 10+ local and offshore development resources tasked with DevOps and Professional Services; handling new retailer deployments, field updates, investigations, lab management, prototypes, and other incomings. The team included software developers, systems engineers, quality, and operations staff. We managed all customers including Walmart, Sam's Club, Costco, BJ's, and various dining chains. Our Walmart footprint alone was a massively distributed network covering 4k stores, each with 15 IPTV players driving more than 60k screens, across a mix of retailer-owned hardware spanning several generations of Dell PowerEdge and IBM blade servers, as well as several generations of Windows OS's. Each retail deployment had its own retailer-driven network, hardware, and software requirements, and we had to manage these heterogeneous hardware and software deployments at scale.
- Managed the development and deployment of the latest-generation of video advertising networks into Costco and BJ's retailer chains. Instead of IPTV streaming, the latest generation of video advertising networks are based on players that pre-cache into local storage, which allows for a pull-based, more reliable playback and control experience.
- As a software engineer, lead the prototype effort for the world's first in-store IPTV video streaming solution, and then followed it through implementation, productization, and deployment into thousands of Walmart stores, serving 140 million US viewers each week.
- Represented PRN as a technical expert by traveling to customer headquarters to resolve high-priority technical issues. Traveled to partner companies to discuss and plan current and future product features and roadmaps.
- Implemented a private cloud in our Lab, using open-source Linux KVM virtualization. This was before AWS and Rackspace offered cloud services.
- Drove inter-departmental improvements to software engineering processes and tools. Improved build and release processes, as well as issue lifecycle. Drove the migration of source control to Perforce, and the issue management system to Jira.

Wireless Engineer

Aruba Networks

Sunnyvale, CA

August 2002 – June 2005. Aruba is an enterprise-class wireless solution company. The products consist of high

performance L2/L3 switches integrated with wireless 802.11 thin access point technology.

- Independently initiated, designed, implemented, and tested an x86 Linux-based virtualized access point platform to cost-effectively emulate hundreds of Aruba hardware APs for scalability testing. Implemented the entire project, including building cross-compilation tool-chains and embedded libraries, patching an architecture specialized MIPS-only kernel to run as x86 user-mode-linux, resolving kernel and userspace byte order issues, and integrating the platform into the software build system. Successfully resolved all hardware dependencies via software implementation, which include emulating the hardware flash device, writing a virtual wireless driver, and emulating the ethernet and bonding devices. (Makefile, C, Linux Kernel Programming, Linux Device Drivers, Perforce).
- Independently responsible for the quality assurance of the management, monitoring, and policy-enforcement features of the wireless solution. Provided guidance on air-management algorithms and product feature set. Independently designed and implemented a fully automated test framework which condensed weeks of manual testing into nightly runs. Wrote object-oriented Perl libraries to simplify and speed the development of test programs. Modified the Linux Atheros madwifi kernel driver to allow the precise generation of arbitrary radio packets, while setting parameters such as frame rates and preambles. Wrote packet generator tools for sales engineers to demonstrate the features of our product. (802.11 Link-Layer, Object-oriented Perl, Shell Scripting, Makefile, Perl Expect, Cisco CLI, VLANs, C, Linux Kernel Programming, Perforce).

Wireless Engineer **Instant802 Networks (now Devicescape)** **San Francisco, CA**

July 2001 - June 2002. Principal software engineer of the OpenAP open-source project which replaced the firmware of commodity 802.11b access points with an open-source embedded Linux operating system.

- Created a custom build environment and custom Linux distribution, which runs on SMC/UsRobotics access points with 1MB of flash ROM and 4MB of RAM. Experienced with embedded libraries, utilities, and linking issues. Current features include dhcp client, boa web server, pcmcia-utils, bridge-utils, wireless-tools, and standard 802.11 access point services. Modified 802.11 wireless drivers to implement multipoint-to-multipoint wireless bridging. Presented and released most work to the open-source community at the Bay Area Wireless Users Group. (C, Linux Kernel Programming, 802.11 Link-Layer, Bridging, Embedded Systems, Embedded Libraries and Utilities, Shell Scripting, Makefile, CVS, Custom Build Environment).
- Member of a team engineering enterprise-class wireless LAN management software based on the Windows .Net platform. Wrote the backend system for the management of enterprise wireless networks, which allows for real-time user monitoring, troubleshooting, auditing, configuration, and management of wireless access points in addition to the identification of unauthorized wireless entities. (C#, .Net Framework, Visual Studio .Net, Visual SourceSafe, SNMP).

Software Engineer **Akamai Technologies** **Cambridge, MA**

Fall 1998 - Spring 2001. Akamai is a distributed and global internet content & application delivery company.

- Developed software to monitor thousands of servers across the Akamai distributed network for numerous internet services, such as web object hosting and multimedia streaming. Designed and implemented a multi-threaded, C++ monitoring agent that reports real-time network status and Akamai QoS performance to the network operations command center. Performed network socket programming on client and server.
- Lead the development effort to manage the configuration and scheduling of monitoring tasks on thousands of monitoring agents. Designed and implemented a scheme that intelligently derives agent configurations from minimal database-stored information.
- The monitoring agent has been deployed on thousands of servers, and provides data used for the derivation of critical alerts and troubleshooting of Akamai system components. These efforts have saved millions of dollars through elimination of company dependence on third party monitoring. This work has become a critical component of revenue generating services such as EdgeSuite and Firstpoint. (Linux, C++, C, Perl, Java, Multi-Threaded Programming, Sockets, CGI, DBI, SQL, MySQL, Perforce, CVS, Network Programming, Knowledge of Network Systems).

Software Researcher **MIT Lab for Computer Science** **Cambridge, MA**

Summer 1998. Worked in a research group under Professor Tom F. Leighton, founder of Akamai Technologies. Helped develop a large-scale distributed caching system for reducing network traffic and eliminating network bottlenecks. Wrote network traffic simulators and monitoring interface. Demonstrated the feasibility of technologies used in startup company Akamai Technologies. (Unix, C, Perl).

SKILLS

Docker, Kubernetes, Openstack, AWS, Rackspace, Google Cloud, GKE, GCE, Elasticsearch, Kibana, Logstash,, Python, Golang, C++, C, C#, Perl, Java, Java Swing, Scheme, Linux Kernel Programming, .Net Framework, Multi-Threaded Programming, Sockets, Shell Scripting, Makefile, gcc, Object-oriented Perl, STL, PHP, HTML, CGI, DBI, SQL, MySQL, msSQL, Expect, SNMP, LaTeX, Unix, Linux System Administration, Win9x/NT/2000/XP/.Net, MS-DOS,

Cisco CLI, CVS, Perforce, Visual SourceSafe, Visual Studio .Net, Embedded Systems, uCLibc Embedded Libraries, Busybox Embedded Utilities, Network Programming, Layer 3 Network Systems, Layer 2 Network Systems, 802.11 Link-Layer, Bridging, VLANs, Jira, UML, KVM, VmWare, Certified Agile Scrum Master and Product Owner, Rally

ADDITIONAL INFORMATION

Honors	John Morrison Scholarship (four year merit-based college scholarship).
Background	US citizen born in Cincinnati, Ohio and raised in Harrisburg, Pennsylvania.
Interests	Yoga, Crossfit, Music, NPR, violin, guitar, tennis, cycling, running, swimming, endurance events.
Software	DD-WRT on Linksys wrt54g and Netgear WNR3500L, iPhone, Android, Facebook applications,
Interests	Linux KVM virtual machines
Patents	PCT/US2007/022617 - Method and system for playlist synchronization PCT/US2007/002562 - Method, apparatus and system for dynamic grouping and content distribution PCT/US2007/022602 - Method and system for improved transition between alternating individual and common channel programming via synchronized playlists