flossUK

news@UK

The newsletter of FLOSS UK, the new name for the UK's oldest Open Systems User Group, UKUUG

Published electronically at `http://www.flossuk.org/Newsletter`

Volume 22, Number 1 **ISSN 0965-9412** **March 2013**

Contents

News from the Secretariat

Thank you to all the members who have paid their subscription invoice promptly this year. Those subscriptions outstanding will be chased at the end of March and any not paid at the end of April will not receive the June Newsletter.

We now have everything in place for the forthcoming Spring Tutorials and Conference being held in Newcastle upon Tyne from 19th – 21st March. The event is being kindly sponsored by HP, Bytemark and our Gold Sponsor member SUSE.

A full list of the Tutorials, Conference talks, abstracts and speaker biographies can be found at: **http://www.flossuk.org/Events/Spring2013**.

Places are still available – you can book on-line via the web site.

We have recently held very successful tutorial days in London. Intermediate Python by John Pinner was well attended on 30th January and the series of Perl tutorials by Dave Cross were well received from 12th - 15th February.

We are currently in talks with O'Reilly to bring some more tutorials later in the year.

The UKUUG Annual General meeting will be held this year on Friday 20th September: please put this date in your diary now.

Other dates to remember:

- OpenTech – Saturday 18th May – London
- Bar Camp – Saturday 1st June – Birmingham
- Unconference – Saturday 26th October – London

And looking ahead – Spring 2014 Tutorial and Conference will be held from 18th - 20th March in Brighton – venue to be confirmed.

The next Newsletter will be the June issue and the copy date is Friday 17th May.

Any comment about past or future events, or if you have something to say about our User Group or this Newsletter please contact **newsletter@ukuug.org**.

If you do NOT wish to receive future issues of the Newsletter in hard copy (all issues can be found on our web site in pdf format) please let me know.

Chairman's Report

Kimball Johnson

Spring 2013

Our Spring Conference in Newcastle is very soon now, and the full schedule is available on our website. Topics include Real-time Monitoring, Software Orchestration, Security and more will be covered over the two days. In addition there are four half day tutorials this year, allowing a choice of one from two each in the morning and afternoon of the tutorial day.

There is still time to book for both the conference and the tutorial, I hope you will be able to join us for the enjoyable 3 day programme.

Support for local user groups

Last year Paul talked about this in his chairman's report. The budget is still available for FLOSS UK to assist local user groups by helping them to obtain speakers for their events and assist with travel expenses. In addition we have a small budget for assisting with projects that would benefit the Free Software or Free Hardware communities in some way. If you have an idea and wish some support, please contact Jane on office@flossuk.org and it will be discussed by the council.

Get Involved

FLOSS UK exists to serve its members and we are always on the lookout for people who are keen to get involved in any capacity, whether that be through volunteering to help with organising events, writing newsletter articles, or entirely new activities which haven't been tried before. If you would like to help out in any capacity please do get in touch via `office@flossuk.org`.

Deploying Rails

Tom Copeland and Anthony Burns
Pragmatic Bookshelf
ISBN: 978-1-934356-95-1
240pp.
£ 22.99
Published: July 2012

reviewed by Nick Booker

Deploying Rails, by Anthony Burns and Tom Copeland, is a book detailing a method of developing, deploying and maintaining Ruby on Rails applications.

The first half of the book talks about how to keep a consistent and reproducable development environment, and how to keep that environment as close as possible to production by building both environments automatically using the same scripts and recipes.

The second half talks about monitoring and maintaining the production infrastructure, so you know when things go wrong, can recover quickly from it and can predict when things might go wrong in the future.

Chapter 1 is very much an introduction, and discusses various methods and places you might want to deploy your applications. It introduces the virtues of automating the building of your deployment environments and deployment itself. Finally, it instructs the reader to download the example source code that's used throughout the book.

Chapter 2 introduces Vagrant, a system for quick deployment of a development environment in VirtualBox. It goes through how to install VirtualBox and Vagrant, how to create a configuration for your host and bundle that with your source code, and how to access, delete

and re-create the virtual machine on demand. Future chapters top up your Vagrant technique as needed.

Chapter 3 introduces you to the basics of Puppet, explaining what it does and how to get it working in your Vagrant environment. It goes through installing and configuring the Apache web server, MySQL, your deployment location for the app itself, and Passenger. Lastly it recommends updating your Vagrant base image to include much of this, saving on development set-up time. This knowledge is used and topped up in future chapters, as more and more tools are installed and configured through Puppet.

Chapter 4 talks about Capistrano, how to use it to deploy releases of your application onto the infrastructure you've prepared with Puppet, and where it puts your code.

Chapter 5 covers Capistrano in more depth, with examples of how to use hooks, restrict tasks with roles, work with the output of the remote programs, deploy to multiple environments and run commands on multiple hosts simultaneously.

Chapter 6 represents a shift in the book's focus, from building an environment and deploying your application to it, to monitoring and maintaining the running of your application and its environment. Chapter 6 covers Nagios, helping you set it up to alert you if things start to fill up, a machine becomes overloaded or your application stops responding. This is done through Puppet, as usual, so all the config for this is reproducable from the same repository.

Chapter 7 discusses "Collecting Metrics with Ganglia" – how to keep an eye on trends, to predict when a disk might run out of space or when you might want more resources; and to help tell the difference between an overload from a gradual increase in usage or a sudden spike in traffic.

Chapter 8 talks about how to maintain the application, keeping Apache and Rails application logs in check so they don't fill the disk (using logrotate), and setting up MySQL failover.

Chapter 9 covers how to install and run several versions of Ruby on the same host machine using RVM, configure applications to run under their own "Passenger Standalone" instances to prevent conflicts between needed Ruby versions, and how to use the monitoring tool "monit" to keep those Passenger instances in check.

The final chapter, 10 "Special Topics", covers the use of the "whenever" gem to bundle application-specific cron jobs with the application itself, how to back up MySQL using the "backup" gem, using Ruby Enterprise Edition (a hardened version of Ruby 1.8) and how to secure the SSH daemon.

An appendix "A Capistrano Case Study" goes quickly through a Capistrano configuration for deploying on JRuby, and the final appendix covers the use of Unicorn and nginx, instead of Passenger and Apache, to serve the application, and why you might want to do that.

After reading this book, I have a better understanding of certain tools such as Puppet and Vagrant and how I might use them in a real deployment, in fact I'm likely to refer to this book quite a lot when working on my next Rails project because it solves some of the problems I'm having with current projects.

I think the book could do with a section on getting Puppet up and running on a real bare Linux installation, and suggestions on how to get the Puppet manifests and custom modules

onto a real machine ready to run and keep them in sync (push vs pull for example). The examples use VirtualBox's host filesystem access for this, which real machines don't have.

Other than that the book is very useful, especially if you're struggling with bugs where "it works on my development machine, but it's broken in production", or just want to make your work that bit easier.

Note that the book is designed for Rails applications, though with a little thought you could probably use many of the techniques and the tools suggested within for other environments too.

Learning Rails 3

Simon St. Laurent, Edd Dumbill and Eric J Gruber
O'Reilly Media
ISBN: 978-1-4493-0933-6
416pp.
£ 26.99
Published: July 2012

reviewed by Nick Booker

Learning Rails 3 is an excellent resource for doing exactly that. The authors present a well-illustrated overview of the task of writing Rails 3 applications.

The book is useful from the Preface onwards. Along with the usual 'conventions used in this book' type of information, the preface provides a useful list of things to double-check if any of the example code in the book doesn't work, though I can see this being useful for searching for the causes of your own bugs.

There are too many chapters to cover individually in a short review, yet for less than 400 pages the book provides a surprising amount of depth, being packed with useful information that with other books you may have to go to the official Rails documentation for, but presented in a concise and easy to understand manner. Each visit to a concept is fairly brief, but is more than enough to get a beginner started, and if you're a little more experienced than that you'll probably learn the occasional new trick. Some features are covered that I didn't know existed and hadn't really thought of looking for.

The book introduces the basics of the main concepts early on, and later chapters reinforce these, adding additional detail both in terms of features and how some of those features work and fit together.

The first three chapters deal with getting the Rails stack installed on your computer, generating your first Rails application, generating a simple "Hello World" type controller and view and applying some CSS, introducing the concept of view 'layouts' on the way.

There is an excellent and clear diagram of how the Rails framework fits together, and another ingenious one that illustrates in simple terms how resource routes map onto generated controllers and views.

Chapters 4 to 6 deal with getting data back and forth between the database and the user, covering the ActiveRecord models API, further depth on scaffolding and how the standard controllers map onto REST and the URL routing system, and how to modify the basic forms provided by the scaffolding to use more meaningful field types, including field types that haven't been covered in other books I've read but are frequently very useful. This includes a method-by-method look at a scaffolding-generated controller, what each method does and how it maps onto the basic REST methods and URLs via the routing system. The use of view helpers to DRY up common view patterns is also covered.

Chapters 7-9 build on the previous chapters, focusing on model-based validation, defining and querying relationships between models and more advanced form programming, including uploading and storing files, defining your own field types with Helpers, and stylesheet integration.

Chapter 10 focuses on database migrations and how you use them to create and manipulate your database schemas over time.

Chapters 11 and 12 cover debugging and automated testing, then Chapters 13 and 14 take on users, authentication with OmniAuth and browser session and cookie management.

Routing is covered in greater depth in its own chapter 15, before chapters 16-18 give you a good grounding in how to use SASS, how the asset pipeline works and how to use this to provide browser-side code with JavaScript and a popular JavaScript pre-processor: CoffeeScript.

The penultimate chapter discusses the basics of sending email from your Rails application, and the final chapter gives some initial flavour of what's needed in order to put your application into production plus some suggested further reading on various parts of the Rails stack.

To end the book there are four short appendices, covering "Incredibly Brief" introductions to Ruby, Relational Databases and Regular Expressions, and an extensive glossary at the end.

Overall, I think this book strikes an excellent balance of breadth and depth of content, and I think it provides excellent coverage of all the important beginners' topics for the page count and a little more.

Getting Started with Raspberry Pi

Matt Richardson and Shawn Wallace
Make
ISBN: 978-1-4493-4421-4
180pp.
£ 11.50
Published: December 2012

reviewed by Roger Whittaker

The Raspberry Pi, as I am sure all readers are aware, is a small ARM system designed as an educational tool by the Raspberry Pi Foundation in the hope that it would stimulate the teaching of some real computer science in schools. This came at the same time that criticism of the school ICT curriculum in this country had even come to the notice of Michael Gove, who has promised changes. The climate of opinion has certainly moved in the right direction, but it will take time to see what can really be achieved. Personally I'm cynical about the chances of the original aims of the Pi being achieved in the educational system. But this and other cheap ARM systems, as well as such things as the Arduino, have resulted in a huge explosion of ideas among "people like us" about projects that they make possible.

This book comes from the O'Reilly "Make" stable (home of "Make Magazine") and so is intended in exactly that spirit. It has a slightly different look and feel (slightly informal) from the O'Reilly books that we are used to.

The book is based on Raspbian and LXDE and does not cover other distributions or operating systems for the Pi (and by now there are many, including BSD variants and RiscOS).

It's not a large book, so it has a big job to do: teach the reader some basic Linux, some basic programming as well as Pi-specific stuff about the board's hardware specifics and GPIO.

As well as a brief introduction to Python, the book also covers (in 11 pages) Pygame and the rather clever "graphical programming language" Scratch which it recommends as a way of introducing children to programming.

There is a chapter on connecting the Pi to an Arduino, and two chapters cover I/O through the GPIO pins. There is a nice chapter on using the Pi with a webcam (something that seems to be a common use for the Pi among users I've spoken to).

Despite the obvious problem that all the topics in the book are necessarily brief introductions and the fact that someone coming to the Pi without any Unix or Linux knowledge will necessarily struggle with many things that are not covered here, this is a useful book that will help many people to get started with using the Pi.

Bad Data Handbook

Q Ethan McCallum
O'Reilly Media
ISBN: 978-1-4493-2188-8
264pp.
£ 30.99
Published: November 2012

reviewed by Gavin Inglis

The cover of "Bad Data Handbook" features a goose. Its species has three different working names and there is healthy debate about whether it should be classified as a "white" goose or "grey" goose. Its appearance in this case is quite apt.

The key question of what comprises "bad" data in this book is dealt with during the introduction. The editor extends the definition beyond malformed records, missing values or misshapen file formats, to any data which consumes unnecessary time, or otherwise gets in the way of your current objective. This wide-ranging definition gives the book its shape: a collection of articles on rather diverse topics. They range from eight to twenty-five pages and vary significantly in approach and tone.

Some are very practical. For instance, Kevin Fink discusses simple analysis techniques, mainly constructing histograms with UNIX command line tools. Paul Murrell considers how best to process data formatted for human eyes. His article covers extracting data from multiple Excel spreadsheets using the R programming language.

One of the most directly useful chapters is written by Josh Levy. He considers "plain" text and likely problems with wrong or unknown encodings. Using exploratory code written in Python, we learn how to interrogate a file with a combination of automatic processing and human scrutiny. The chapter concludes with an exercise which requires you to decode details of financing for terrorism!

Adam Laiacano covers data acquisition through website crawling and "screen scraping". The discussion ranges from how to parse the resulting HTML, how to avoid being blacklisted and, memorably, how to pull data from "traffic lights" displayed by an Adobe Flash plugin. The topic of building databases through web crawling is one which recurs later in the book.

Other chapters like Jonathan A Schwabish's "Subtle Sources of Bias and Error" are pitched at a higher, more theoretical level. Some, like Brett Goldstein's "Don't Let the Perfect Be the Enemy of the Good", are a more personal and philosophical account. He takes as a starting point his statistical training at college and the changes he encountered when transferring to industry. The Spencer Burns chapter is a compact but interesting read on complexity associated with stock market data, and the possible effects of misunderstanding it. Steve Francia's "Myths of Cloud Computing" is a fictional, but utterly believable cautionary tale about launching a business with an entirely cloud-based infrastructure.

The more practical chapters sometimes feature example code, but this book is not like O'Reilly's more familiar programming manuals. It is likely to be most attractive to a data professional, with material ranging from nuts-and-bolts basics to management experience

gained in the data trenches.

"Bad Data Handbook" is quite expensive for a thin book with such a variety of content. Its articles sit together a little uncomfortably. There are definite insights to be found inside and some of the chapters are sufficiently general to be of wider interest. It is probably good value for those who work regularly with troublesome data. Certainly one can't ignore a chapter entitled "Blood, Sweat, and Urine", or indeed, "Detecting Liars and the Confused in Contradictory Online Reviews". Feel free to apply its techniques to this particular review.

CSS3: The Missing Manual, 3rd Edition

David Sawyer McFarland
O'Reilly Media
ISBN: 978-1-449-32594-7
538pp.
£ 26.99
Published: December 2012

reviewed by Gavin Inglis

My dad laments the decline of the printed manual in software. Of course he used to buy all his software on cassette. However in today's IT environment one can have an idea in the shower and install a development framework and web server before lunch. It's not surprising that we rely on web documentation, or that we don't understand our technology as thoroughly as we used to. O'Reilly's Missing Manual series aims for friendly but complete coverage of each subject. This one concerns the latest iteration of Cascading Style Sheets.

The CSS3 volume actually begins with a chapter on HTML5. It seems accessible to even a total beginner (although I don't recommend choosing this book as your entry into HTML) but quickly gets down to the specifics of HTML5 coding for the experienced author, such as the new structural tags and the changing fortunes of <b> and <i>. This emphasis on the scaffolding is a good setup for the chapters to follow, and the chapter concludes with advice to look after your DOCTYPE and the various browser perils which may otherwise befall you. It also considers how to handle outdated versions of Internet Explorer.

Once the text turns to creating actual CSS, the pace is brisk. The "Your first styles" exercise includes a Google web font, background images and box shadow. Experienced CSS authors will probably skip this but those with less confidence will gain a lot by following each step closely and taking the time to understand what is changing and why. Likewise, the section on selectors makes no apology for immediately going as deep as pseudo-classes and attribute selectors. In general, Part One is the "heaviest" section of the book in terms of theory, and things open up with the more visual Part Two.

One nice subtlety is that the exercises are designed to accommodate both complete beginners and those who will skim through the book until they hit a point at which their existing knowledge runs out. Later tutorials still direct you to the URL to download example code, so a reader who has skipped the early exercises does not have to go searching for it. Even instructions on how to use your text editor are spelt out very clearly; there is a balance to

be struck and this book gets it just right. Sometimes the text suggests you skip a theoretical chapter and return later after some hands-on coding.

The coverage of CSS3 is very comprehensive, getting down to details such as the specificity formula for style conflicts, or the difference between content-box and border-box sizing. Readers who have been working with CSS for a while will probably have more than one "eureka" moment upon reading the reason for a previously impenetrable quirk of behaviour.

The tone throughout is direct and matter-of-fact. Not much space is wasted, meaning you get a lot of book for your money, but it's a dense learning experience if you're starting from a position of zero knowledge.

Part Three breaks away from the code for a more general discussion of how to approach layout, considering grids and responsive design. Part Four deals briefly with CSS for print, and establishing good practice as a CSS designer.

It's perhaps not surprising – but still depressing – that we still require extended discussions of how to support older versions of Internet Explorer and their bugs. It is, of course, the reader's choice whether to use the bloated code that often results.
You will not find much artistic coaching in "CSS3: The Missing Manual". It is a comprehensive and well-executed guide to the full toolset; what you do with that is up to you.

Version Control with Git

Jon Loeliger and Matthew McCullough
O'Reilly Media
ISBN: 978-1-4493-1638-9
456pp.
£ 26.99
Published: August 2012

reviewed by Quentin Wright

This book is immediately recognisable as it has a large illustration of a long-eared bat on the cover! It's a second edition of what is undoubtedly the best introductory and reference book for the git distributed version control system. Git was originally developed to manage Linux kernel development and thanks to the popularity of the Github web service has become the leading tool for managing distributed software projects.

An outline of the contents of the book follows.

Chapter 1 – Introduction. This chapter summarises the characteristics that Linus Torvalds was looking for as an alternative to Bitkeeper that weren't in any other version control system. There's a brief mention of the antecedents to git along with the first use of git and the commit of the Linux 2.6.12 kernel in April 2005, a mere 17291 files and 6.7 million lines of code!

Chapter 2 – Installing Git. As the name indicates this chapter covers the installation of git for different environments. Curiously Mac OS X is omitted although installation can be achieved by simply downloading and installing a dmg file.

Chapter 3 – Getting Started. This chapter describes typical use from the command line and use of `git show`, `log` and `diff`. It moves on to detail how to remove files, clone a repository and carry out basic git configuration.

Chapter 4 – Basic Git Concepts. A good explanation of git internals, SHA1 and a mention of tags.

Chapter 5 – File Management and the Index. This chapter describes the workings of the git index, how to use `git add` and `git status` and goes on to outline git's object model and files and uses powerful graphical representations as part of the explanation.

Chapter 6 – Commits. Shows how to use git log to explore the commit history and introduces diagrams to illustrate what is going on. How to use `git bisect` and `git blame`.

Chapter 7 – Branches. Rationale for branches. Creating, listing, showing branches. Checking out branches, merging branches. Creating and switching branches. Dealing with detached head branches. Deleting branches. All of these operations are explained with clear examples.

Chapter 8 – Diffs. This chapter takes unix `diff` and introduces and makes comparisons with `git diff`. Then works through a simple example of `git diff`. Shows how both commit and path ranges can be applied to `git diff`.

Chapter 9 – Merges. Starting with at example of the simple merging of two branches this chapter moves on to deal with merge conflicts and shows how to carry out manual conflict resolution. There's an "Alice, Bob and Cal" example of a criss-cross merge. An explanation of different merge strategies: already up-to-date, fast-forward. Resolve, recursive and octopus merge strategies. Specialty merges: ours and subtree strategies. This chapter is particularly useful in its explanation of merge strategies, something that can't be found elsewhere.

Chapter 10 – Altering Commits. Why you might or might not want to and should and shouldn't alter commits. Use of git reset, git cherry-pick and clarification of `git reset`, `revert` and `checkout`. Use of `gitrebase`.

Chapter 11 – The Stash and Reflog. Coping with interrupted work-flow: `git stash`. Using `git reflog` to work out what's been going on!

Chapter 12 – Remote Repositories. Using remote repositories, remote tracking branches. Remote authoritative repositories. Pushing to remote repositories. Adding a developer. Using `gitfetch`. Using `git merge` as against `gitrebase`. An extra explanation of tracking branches and adding and deleting of remote branches. Why you should only push to a bare repository.

Chapter 13 – Repository management. How to use repositories with controlled access. Publishing repositories and anonymous read access. Publishing with http and SmartHTTP. Publishing via git and http daemons. Anonymous write access and why you shouldn't do it. Repository structure and examples from the Linux kernel project. General discussion: "Living with Distributed Development".

Chapter 14 – Patches. Patching as an alternative to git and http protocols, git patch has 3 commands: git format-patch which generates a patch as an email, `git send-email`

which will send a patch through an SMTP feed and `git am` which applies a patch in an email. Patches are good for a specific small change. Again this whole process of patching is tracked through with examples.

Chapter 15 – Hooks. This features a brief discussion of hooks, which are scripts which can be run when specific events occur. Git comes with some standard hooks which are described.

Chapter 16 – Combining Projects. A discussion about ways of getting sub-projects into a git repository and leads into the following chapter about submodules.

Chapter 17 – Submodule Best Practices. How to use submodules. Gives a few use cases including open sourcing a book's code samples.

Chapter 18 – Using Git with Subversion Repositories. How to use git with subversion repositories.

Chapter 19 – Advanced Manipulations. There's some really heavy stuff here that could occasionally come in handy: `gitfilter-branch` can be used to make large-scale changes to a repository maybe before publishing it. Examples are given of expunging a file from a repository and editing a commit message. `gitrev-list` is used to obtain date-based checkouts and retrieving old versions of a file. Also covered are using `git add-p` to stage in hunks and using `git fsck` to locate "lost" commits.

Chapter – 20 Tips, Tricks and Techniques. Even more snippets here! Among them `git gc` for garbage collection – should probably be given more prominence! As should `git grep` which can be used to search the contents of files in a repository.

Chapter 21 – Git and Github. This chapter contains an overview of Github which is of some value. It could have described how to set up ssh access and putting an authentication key, and then a typical use case.

This book is an excellent resource for someone starting with git. It has short chapters on specific topics logically arranged and all commands have clear examples and explanatory text. Chapter 9 on merging is especially useful. One feels that the chapter on Github has been just tacked on for the second edition where it might have been introduced into the main body of the text and enhanced with an explanation of adding a key to github and cloning a repository using its git url.

Probably some of the examples of working with subversion could be removed, but that's probably a reflection of a particular prejudice of the reviewer!

The organisation of the book with its clear structure makes it easy to use for reference and it can be strongly recommended to any user of git.

21st Century C

Ben Klemens
O'Reilly Media
ISBN: 978-1-4493-2714-9
298pp.
£ 22.99
Published: November 2012

reviewed by Paul Waring

Not having touched much C since university – when a trusty copy of K&R was my guide to the language – I was intrigued by the title of this book which claims to cover modern C. A word of warning up front however: this book assumes the use of C99, and in some cases C11. Most compilers support C99, but there is a great big gap in the form of Visual C++. The author's suggestion if you want to compile C on Windows is basically 'install Cygwin'.

The preface and first chapter largely set the scene, and reveal some of the author's prejudices against interpreted languages, C++ and Windows (these show up throughout the book). The chapter on debugging, testing and documenting provides a good overview of these subjects, although detailed coverage would be a book in itself. The third chapter covers packaging – or perhaps 'compiling' would be more accurate. This is worth reading solely for its explanation of how autotools works. This is complemented by the following chapter on version control, which doubles up as a Git tutorial.

The middle section of the book focuses on the language itself. The first chapter in this section covers the thorny issue of pointers, and does a good job of explaining the different ways of allocating memory, as well as pointer arithmetic. Chapter seven covers some C syntax which, according to the author, you can 'ignore'. Some of these are arguable either way (e.g. not returning a value from `main()` is fine if you're happy to specify C99 as a minimum), others are a bit more controversial – such as using `goto` but avoiding `switch`. The chapter on text handling spends a lot of time explaining how Unicode works, although most readers can skip to the final page with the notes on `gettext`.

The penultimate chapter is the most confusing of the book, covering object orientated C. The overall message seems to be that OOP is bad (as are C++ and Java), but that you can accomplish some of the same things if you use bits of C11. The end result appears to be fairly unreadable code, and this chapter can be safely skipped. The final chapter covers some popular libraries such as pthreads and libxml. However, the space available means that this is necessarily a brief overview and you will need to read the separate documentation to move beyond basic examples.

The only potential issue you may find with this book is that the author describes specific tools rather than general techniques. For example, the chapter on version control is really a brief introduction to Git – Subversion users will find little of use here. If you already use the author's preferred tools, or are able and willing to switch, then this is a good all round collection of tips and advice on C programming – not just the language but also the surrounding environment.

Hadoop: The Definitive Guide

Tom White
O'Reilly Media
ISBN: 978-1-4493-1152-0
688pp.
£ 38.50
Published: May 2012

reviewed by Andy Thomas

Big Data. Two words that have been known to strike fear into the hearts and minds of even the most hardened IT professionals. But Big Data is here to stay and for those of us whose work brings us into contact with large datasets and the need to analyse their content within a reasonable timescale, we need get to get over it, embrace it and learn to make the most of the new opportunities it presents for business and research organisations alike. It has been said that from the dawn of civilisation until around the year 2003, the human race generated 5 exabytes (or 1,000,000,000 Gigabytes) of information. Now, in 2013, we are apparently generating this amount of data every two days! Working in research institutions myself, I can well believe this and it can present a huge problem, not only that of storing it and making sure it is backed up but also in examining, processing and using all this data for meaningful purposes.

While several approaches to the problem of processing and analysing data on the terabyte, petabyte scale and beyond exist, the Apache Hadoop project is perhaps the most comprehensive and widely used system now available, combining the Hadoop distributed file system (HDFS) originally developed by Yahoo with Google's MapReduce framework (which in turn draws on Google's earlier work with GFS, The Google File System) which divides computationally intensive tasks into smaller chunks and assigns the work to nodes in a server cluster. Add in the Hive data warehousing system, the Pig query language, HBase for handling structured datasets, Sqoop for transferring existing SQL data into and out of HDFS and ZooKeeper for creating distributed clusters along with many other utilities, you soon realise Hadoop is a vast ecosystem in its own right, designed to be fault tolerant, scalable and flexible.

But many people (including myself) find Hadoop hard to understand initially – turning many common data processing notions on their heads, it is too much to take in at first and you need to put aside much of what you already know of servers, operating systems, etc at the node level, read the documentation and concentrate on the Big Picture. And this is where this book comes to the rescue – exceptionally well-written by a master of plain talk, Tom White's third edition of Hadoop: The Definitive Guide at last makes sense of what can at times seem a mind-bogglingly complex system. Covering not only the history and theory of the various parts of the Hadoop project, the book includes numerous practical examples, snippets of Java, Ruby and Python code illustrating how to interface to the various HDFS APIs and some real-world data analysis examples.

Starting with a description of the genesis of the Hadoop project in chapter one, the reader is then treated to a gentle introduction to MapReduce, using sample weather datasets from the US National Climatic Data Center (NCDC), where we soon start to realise traditional

ways of mining data are going to take a very long time to complete. Next we learn about HDFS and Hadoop I/O with a chapter devoted to each and then we get down to brass tacks with MapReduce, four solid chapters covering the development of MapReduce applications, its internals (covering both the original MapReduce 1 implementation and the newer YARN/MapReduce 2), the different data types that can be used in the MapReduce model and rounding off this topic with a look at some of the advanced features of MapReduce.

Chapter 9 tells us how to set up a Hadoop cluster from scratch followed by a chapter on Hadoop systems admin – I personally found this chapter the most useful to begin with, after I took over the running of an existing Hadoop cluster with no prior knowledge of Hadoop (and before getting involved some time later with setting up new clusters using Cloudera's CDH3 distribution). In chapter 11 we meet Grunt, the interactive shell to Pig which is the first of the five Hadoop optional application programs discussed in the book, and get to learn some Pig Latin.

Acknowledging the fact that there are a lot data analysts out there with well-honed SQL database skills but little knowledge of Java, C++ and Python, Hive was developed by Facebook to allow SQL queries to be run on the vast amount of data this social networking site stores in HDFS, converting the queries into a series of MapReduce jobs and its installation and use is covered in the next chapter. Chapter 13 covers HBase, which is a very large and scalable column-oriented database originally developed by Chad Walters and Jim Kellerman for web search engine use. Like Hadoop, HBase is yet another application that challenges all that you knew about databases for RDBMS it is not nor does it support SQL – designed with a completely different approach to most other DBMS, it supports very fast random data reads, writes and updates.

Running distributed processes over many nodes is inherently vulnerable to all sorts of things going wrong – this is both dictated and predicated by Murphy's First Law. But here ZooKeeper comes to the rescue, providing a set of tools and libraries for creating highly available distributed processing systems and chapter 14 is devoted to installing and running this service. The last of the optional Hadoop applications covered by this book is Sqoop, a system for importing from and exporting data to traditional RDBMS from HDFS, with detailed examples given of doing these operations in conjunction with a MySQL database.

The final chapter covers in some detail several real-world case studies of the use of HDFS and its associated applications at organisations such as Last FM, Facebook and Rackspace, concluding with a close look at the Nutch Search Engine and the Cascading Java library and API abstraction layer for MapReduce. Three short appendices cover the installation of Apache Hadoop, a description of the Cloudera Hadoop distribution (CDH, which include the Apache Hadoop core along with all the optional applications) and a note on how the NCDC weather datasets were prepared for use by Hadoop in chapter 2 of the book.

Written by a senior Apache Hadoop committer who has a particular interest in making Hadoop easier to use and understand, and ambitious in its remit, no review can ever do this book justice. Packed into its 657 pages is all you'll need to know to set up and running Hadoop and HDFS and for using MapReduce and the various applications available to interface with it. If you need to handle Big Data, use Hadoop. And if you use Hadoop, buy this book.

Contributors

Nick Booker lives in Rugby, works with Linux as a Ruby, Python and Perl programmer, and is the technical director of Sunfield Technology. At work he mostly writes web applications and maintains manuals, but also writes scripts for just about everything he can and is gradually replacing himself with a suite of cron jobs. In his spare time he's trying to learn Parkour and French; he swims regularly, is an opportunist photographer and occasionally gets his telescope out for spot of stargazing.

Gavin Inglis works in Technical Infrastructure at the EDINA National Data Centre in Edinburgh. He is a teacher, photographer and musician who has recently discovered the joys of spreadsheets as a recreational tool.

Kimball Johnson is a Systems Developer for Lancashire County Council He has been programming since a very early age, starting with BBC Micros, then MS DOS and Windows Systems, however was enlightened with a copy of Debian GNU/Linux Woody at university. He is always wanting to learn and has recently started to take on embedded programming on a variety of devices, on everything from a Arduino to a Nintendo DS.

Jane Morrison is Company Secretary and Administrator for UKUUG, and manages the UKUUG office at the Manor House in Buntingford. She has been involved with UKUUG administration since 1987. In addition to UKUUG, Jane is Company Secretary for a trade association (Fibreoptic Industry Association) that she also runs from the Manor House office.

Andy Thomas is a UNIX/Linux systems administrator working for Dijit New Media, Imperial College London and a number of small companies. Having started with Linux when it first appeared in the early 1990's, he now enjoys working with a variety of UNIX and Linux distributions and has a particular interest in high availability systems and parallel compute clusters.

Paul Waring is a Council member of FLOSS UK and a director of a wholesale insurance broker. Outside of work he can usually be found filing documentation bugs against various open source and free software projects.

Roger Whittaker works for SUSE supporting SUSE Linux Enterprise Server for major customers in the UK. He is also the UKUUG Newsletter Editor, and co-author of three successive versions of a SUSE book published by Wiley.

Quentin Wright is a Director of Sunfield Technology working with Linux and Ruby and system integration projects. In his spare time in between playing with reluctant motor vehicles he is pre-occupied with Web and Javascript programming.

Contacts

Kimball Johnson
UKUUG Chairman
Preston

Paul Waring
Treasurer
Manchester

Jon Dowland
Council member
Newcastle

Holger Kraus
Council member
Leicester

Ian Norton
Council member
Rochdale

Stephen Quinney
Council member
Edinburgh

Quentin Wright
Council member
Warwick

Roger Whittaker
Newsletter Editor
London

Alain Williams
UKUUG System Administrator
Watford

Sam Smith
Events and Website
Cambridge

Jane Morrison
UKUUG Secretariat
PO Box 37
Buntingford
Herts
SG9 9UQ
Tel: 01763 273475
Fax: 01763 273255
office@ukuug.org