Foreword

Foreword

Google's story is a story of scaling up. It is one of the great success stories of the computing industry, marking a shift towards IT-centric business. Google was one of the first companies to define what business-IT alignment meant in practice, and went on to inform the concept of DevOps for a wider IT community. This book has been written by a broad cross-section of the very people who made that transition a reality.

Google grew at a time when the traditional role of the system administrator was being transformed. It questioned system administration, as if to say: we can't afford to hold tradition as an authority, we have to think anew, and we don't have time to wait for everyone else to catch up. In the introduction to *Principles of Network and System Administration* [Bur99], I claimed that system administration was a form of human-computer engineering. This was strongly rejected by some reviewers, who said "we are not yet at the stage where we can call it engineering." At the time, I felt that the field had become lost, trapped in its own wizard culture, and could not see a way forward. Then, Google drew a line in the silicon, forcing that fate into being. The revised role was called SRE, or Site Reliability Engineer. Some of my friends were among the first of this new generation of engineer; they formalized it using software and automation. Initially, they were fiercely secretive, and what happened inside and outside of Google was very different: Google's experience was unique. Over time, information and methods have flowed in both directions. This book shows a willingness to let SRE thinking come out of the shadows.

Here, we see not only how Google built its legendary infrastructure, but also how it studied, learned, and changed its mind about the tools and the technologies along the way. We, too, can face up to daunting challenges with an open spirit. The tribal nature of IT culture often entrenches practitioners in dogmatic positions that hold the industry back. If Google overcame this inertia, so can we.

This book is a collection of essays by one company, with a single common vision. The fact that the contributions are aligned around a single company's goal is what makes it special. There are common themes, and common characters (software systems) that reappear in several chapters. We see choices from different perspectives, and know that they correlate to resolve competing interests. The articles are not rigorous, academic pieces; they are personal accounts, written with pride, in a variety of personal styles, and from the perspective of individual skill sets. They are written bravely, and with an intellectual honesty that is refreshing and uncommon in industry literature. Some claim "never do this, always do that," others are more philosophical and tentative, reflecting the variety of personalities within an IT culture, and how that too plays a role in the story. We, in turn, read them with the humility of observers who were not part of the journey, and do not have all the information about the myriad conflicting challenges. Our many questions are the real legacy of the volume: Why didn't they do *X*? What if they'd done *Y*? How will we look back on this in years to come? It is by comparing our own ideas to the reasoning here that we can measure our own thoughts and experiences.

The most impressive thing of all about this book is its very existence. Today, we hear a brazen culture of "just show me the code." A culture of "ask no questions" has grown up around open source, where community rather than expertise is championed. Google is a company that dared to think about the problems from first principles, and to employ top talent with a high proportion of PhDs. Tools were only components in processes, working alongside chains of software, people, and data. Nothing here tells us how to solve problems universally, but that is the point. Stories like these are far more valuable than the code or designs they resulted in. Implementations are ephemeral, but the documented reasoning is priceless. Rarely do we have access to this kind of insight.

This, then, is the story of how one company did it. The fact that it is many overlapping stories shows us that scaling is far more than just a photographic enlargement of a textbook computer architecture. It is about scaling a business process, rather than just the machinery. This lesson alone is worth its weight in electronic paper.

We do not engage much in self-critical review in the IT world; as such, there is much reinvention and repetition. For many years, there was only the USENIX LISA conference community discussing IT infrastructure, plus a few conferences about operating systems. It is very different today, yet this book still feels like a rare offering: a detailed documentation of Google's step through a watershed epoch. The tale is not for copying—though perhaps for emulating—but it can inspire the next step for all of us. There is a unique intellectual honesty in these pages, expressing both leadership and humility. These are stories of hopes, fears, successes, and failures. I salute the courage of authors and editors in allowing such candor, so that we, who are not party to the hands-on experiences, can also benefit from the lessons learned inside the cocoon.

Mark Burgess

Author of In Search of Certainty
Oslo, March 2016

next

Preface