

[Home](#)[Shop Video Training & Books](#)[Radar](#)[Safari Books Online](#)[Conferences](#)[More Topics ▼](#)

We're in the process of moving Radar to the new oreilly.com. Check it out.



## Operations: The New Secret Sauce

by [Tim O'Reilly](#) | [@timoreilly](#) | [+Tim O'Reilly](#) | July 10, 2006



I spoke last week with [Debra Chrapaty](#), the VP of Operations for Windows Live, to explore one of the big ideas I have about [Web 2.0](#), namely that once we move to software as a service, everything we thought we knew about competitive advantage has to be rethought. Operations becomes [the elephant in the room](#). Debra agrees. She's absolutely convinced that what she does is one of the big differentiators for Microsoft going forward. Here are a couple of the most provocative assertions from our conversation:

1. Being a developer "on someone's platform" may ultimately mean running your app in their data center, not just using their APIs.
2. Internet-scale applications are pushing the envelope on operational competence, but enterprise-class applications will follow. And here, Microsoft has a key advantage over open source, because the Windows Live team and the Windows Server and tools team work far more closely together than open source projects work with companies like Yahoo!, Amazon, or Google.

Let me expand on these two points. Did you ever see the children's book, [Cloudy, With a Chance of Meatballs?](#) I summarize my conversation with Debra as "Cloudy, with a

chance of servers.” (I started with a pun on the title, but the book description from Amazon is particularly apt: “If food dropped like rain from the sky, wouldn’t it be marvelous! Or would it? It could, after all, be messy. And you’d have no choice. What if you didn’t like what fell? Or what if too much came? Have you ever thought of what it might be like to be squashed flat by a pancake?” But I digress. Back to Debra.)

People talk about “cloud storage” but Debra points out that that means servers somewhere, hundreds of thousands of them, with good access to power, cooling, and bandwidth. She describes how her “strategic locations group” has a “heatmap” rating locations by their access to all these key limiting factors, and how they are locking up key locations and favorable power and bandwidth deals. And as in other areas of real estate, getting the good locations first can matter a lot. She points out, for example, that her cost of power at her Quincy, WA data center, soon to go online, is 1.9 cents per kwh, versus about 8 cents in CA. And she says, “I’ve learned that when you multiply a small number by a big number, the small number turns into a big number.” Once Web 2.0 becomes the norm, the current demands are only a small foretaste of what’s to come. For that matter, even server procurement is “not pretty” and there will be economies of scale that accrue to the big players.

Her belief is that there’s going to be a tipping point in Web 2.0 where the operational environment will be a key differentiator. I mentioned the idea that Web 2.0 has been summed up as “Fail Fast, Scale Fast,” and she completely agreed. When it hit its growth inflection point, MySpace was adding a million users every four days — not at all an easy feat. As these massive apps become the norm, unless you can play in a game where services can be highly stable, geodistributed, etc., you won’t be in the game. And that’s where she came to the idea that being a developer “on someone’s platform” may ultimately mean running your app in their data center. Why did Fedex win in package delivery? They locked up the best locations with access to airports, warehousing, etc. so they had the best network. A similar thing will happen with packet delivery.

Who are the competitors of the future in this market?

Microsoft, Google, Yahoo! and the telcos were the folks she called out, with a small nod to Amazon’s platform aspirations. (Sure enough, in true [news from the future](#) style, into my inbox comes [Jon Udell’s review of a new service called OpenFount](#):

“Openfount’s big idea is that a solo developer ought to be able to deploy an AJAX application to the web without worrying about how to scale it out if it becomes popular. If you park the application’s HTML, JavaScript, and static data files on Amazon’s S3 storage service, you can make all that stuff robustly available at a cost that competes favorably with conventional hosting.”)

Debra also talked about the importance of standardization, so that increasing capacity is as seamless as possible. “We’ve got to make it like air and water.” In that regard, another very interesting point we discussed was [Ian Wilkes’ thought that database tools were still weak when it came to operational provisioning](#). That was where she came to the second point above, that Microsoft has a key competitive advantage here. Internet-scale applications are really the ones that push the envelope with regard not only to performance but also to deployment and management tools. And the Windows Live team works closely with the Windows Server group to take their bleeding edge learning back into the enterprise products. By contrast, one might ask, where is the similar feedback loop from sites like Google and Yahoo! back into Linux or FreeBSD?

As Shakespeare said, “The game’s afoot.” Debra put more servers into production in the last quarter than she put in place in all of the previous year, and she thinks this is just the beginning. Operations used to be thought of as boring. It’s now ground zero in the computing wars.

P.S. When I circulated a draft of this message to the Radar team, Nat wrote: “Open source definitely has this. Linux, FreeBSD, Apache, Perl, Python, and Ruby there’s a huge crossover between large-scale deployers and core project members. There’s been a lot of talk about how the Linux kernel is now really only developed by people employed full-time by big companies. If you look at Yahoo! or Google, they have a ton of kernel and language people working for them. I think there’s a pretty tight loop there.”

I replied, “I know this is true on the core of many projects, but is it true with regard to tools for managing *operations*, which was Debra’s point?”

Nat replied: “Deployment tools have *never* been open source’s strong point: OS has always been about the developer, rarely about the deployer. cf the hacker’s disdain

for IT who get stuck with deployment and management. That said, there are some open source tools like [nagios](#) (for system monitoring) and [capistrano](#) (for rails deployment). The feedback loop there tends to be that the people writing the tools are the ones with the deployment problem. The downside is that if your need isn't met by the tool, it may be hard to get the developer to add it. (That's why [Hyperic](#) is on good terms with Nagios: the Nagios developer will never add the features that Hyperic has.)”

He continued: “The deployment tools tend to be commercial offerings in open source, where Red Hat, IBM, et al. give away the open source operating system and charge like a wounded bull for the management tools. Walking around LinuxWorld Boston two years ago convinced me of this: *everyone* had management tools. Third party management tools suffer because of the lack of integration. Red Hat at least can pair the management people with the kernel people and get the integration they want. I'm not ready to believe that the Windows server story is 10/10. I'd say the open source story is only 5/10. There's a lot more to be done.”

I do think Microsoft has an advantage here over Linux and LAMP stack. But more to Debra's point, it is outmoded to think of a software stack alone as the platform. Microsoft's competition in this arena is not Linux, but Google, and the other Web 2.0 platform players, who have their own operational competencies, and as far as I know, are *not* releasing them to the open source community. What's more, I'm not even sure that the open source community understands just how important this whole area is going to be, so even if the tools and techniques were released, I'm not sure how strong the uptake would be.

Your thoughts?

---

## Related

- Operations is a competitive advantage... (Secret Sauce for Startups!)
- Early History of the Web Pipes Concept
- Cultivating Server Farms
- Open source community

## Explore O'Reilly Radar

- Building and deploying large-scale machine learning pipelines
- Creative computing with Clojure
- Parsing HTML with Perl

# collaboration strategies for the enterprise • The Future of AngularJS

Powered by

tags: [operations](#), [Web 2.0](#)



Comments for this thread are now closed. ✕

**Comments**   **Community**   1 **Login** ▾

**Recommend**   **Share**   **Sort by Best** ▾



**Rachel** • 9 years ago

In my opinion you've forgotten to mention to [marktplaas.nl](#), I think also will get to be a great competitor in this market.....do you think so??

^ | ▾ • [Share](#) ▸



**akoper** • 9 years ago

I think Tim makes some good points with his piece. I have been reading in the NY Times and WSJ about the big data centers the big Internet companies are building and how the Internet business may not be the high-profit-margin, non-capital-intensive industry people thought it was at first.

I'm going to post my second thought first and my first thought second.

---

[see more](#)

^ | ▾ • [Share](#) ▸



**Bill Dav** • 9 years ago



**Bill Day** · 9 years ago

Very interesting, thanks for the write-up Tim.

I blogged about A3 and FolderShare earlier this evening. See: <http://billday.com/2006/07/24/network-storage-and-file-sharing/>

Maybe what we really need is a combination of both? "A3 BitTorrent"?

^ | v · Share ›



**Steve Judkins** · 9 years ago

Operations is only a secret sauce if someone finds your application valuable enough to want to clone your operations. Why are the Google File System (GFS) and datacenter operations interesting? Because it has been used to successfully attack three nagging problems with running web applications at scale: commoditizing hardware, commoditizing software, and commoditizing (access to) power. The keyword being "commodity". We are witnessing the transformation to a mature industry where software is deployed at scale (in factory like settings) and companies must become operationally efficient with some of their biggest fixed costs. The prospect of powering hundreds of thousands of servers is a daunting challenge.

I don't think there is any reason for people to be alarmed about this and I certainly don't think this

[see more](#)

^ | v · Share ›



**Dion Hinchcliffe** · 9 years ago

Tim:

Great job raising awareness of this trend. While I think operations will be a strategic issue and there will be problems until a common application hosting standard is created, I do think it will soon be a commodity item. Nevertheless, for now, choosing

your operations provider will be something that needs to be done very carefully.

My take on operations in the Web 2.0 era, as well as a brief review on some companies doing this now,

---

[see more](#)

^ | v • Share ›



**Luke Kanies** • 9 years ago

I agree that operations, far beyond deployment into full lifecycle management, are critical. Ironically, the new generation of web application entrepreneurs seems to be both ignoring and causing problems of scale -- PubSub's founder commented that doubling his server's RAM was his scaling plan -- and its requisite management problems. Rails is a great framework but it's easy to build resource-hungry applications with it that are very difficult to manage.

Given how important operations are, I'm surprised at how difficult it is for entrepreneurs in this space, like myself, to get the attention of VCs or media. I expect that my proposal about my product [Puppet](#) was the only one to OSCON that attempted to address

---

[see more](#)

^ | v • Share ›



**Joe** • 9 years ago

Too long a response for the box - so I posted it on my blog and set a trackback.

<http://www.rhonabwy.com/wp/2006/07/11/operational-effectiveness/>

^ | v • Share ›



**genehack** • 9 years ago

I think the thing that's really getting missed about



JMason's comment is the "also see" bit at the end. People *have* been working on and thinking about the operations side, and it's led to tools like cfengine, puppet, bcfg, etc. They're fairly complicated to deploy (as are AD and WSS and the like), but they're cross-platform, OSS, vendor-neutral -- and once you've got them working, *extremely* powerful.

^ | v • Share ›



**Adrian Cockcroft** • 9 years ago

I have a few somewhat unrelated comments.

A comment on systems management, my observation is that while everyone wants better systems management, no-one wants to pay for it, it is intrinsically expensive to build because of the permutational complexity of support, and every user has a different permutation. The end result is that the best tools cost too much and still don't solve your problem, so everyone minimises their permutations and rolls their own tools for that subset. A common abstraction model like the [Grid Reference Architecture](#) helps, and Open Source building blocks make it easier to roll your own.

---

see more

^ | v • Share ›



**Tim O'Reilly** • 9 years ago

Greg -- I did read the Baseline article, and recommend it to anyone interested in this topic.

Julian -- BitTorrent still has a big impact. But you're right that distributed computing hasn't taken off quite as much as I would have expected. When I first started thinking about "the internet operating system", it was very high on my radar. (And I even funded a startup in the space, which never got follow-on venture funding, and failed.) But it definitely is happening still in the science community, and one



day an easy-to-use general-purpose platform may emerge.

^ | v • Share ›



**jbond** • 9 years ago

Randy H.: What happened to distributed computing?

Indeed. And what happened to decentralisation and peer to peer?

The big problems are interesting. But actually more useful is to solve the small ones. Take a copy of Drupal, run it on cheap shared hosting. Then work out how to cope when your usage jumps by 1000. Isn't it ironic that many, many sites crumple under the Slashdot and Digg effects?

^ | v • Share ›



**Greg Biggers** • 9 years ago

Tim, you've likely already seen this--

[baselinemag.com](http://baselinemag.com) (via slashdot) has a story up on bits of Google internal ops. They quote Schmidt talking about the same principle you raise: operations as a core (i.e. not outsourced, not purchased from another software or hardware company) competence for high-scale web2.0 companies.

From [baselinemag.com](http://baselinemag.com): (italics mine)

Google buys, rather than leases, computer equipment for maximum control over its infrastructure. Google chief executive officer Eric Schmidt defended that strategy in a May 31 call with financial analysts. *"We believe we get tremendous competitive advantage by essentially building our own infrastructures,"* he said.

^ | v • Share ›



**BEETLE!** • 9 years ago

"We learned that when you multiply a small number



*I've learned that when you multiply a small number by a big number, the small number turns into a big number."*

She's not thinking small enough. Where I work, small is somewhere less than  $3.7 \text{ E } -16$  or so... multiply that all you like.

What happened to distributed computing? The average desktop PC now literally outpaces supercomputers of 10 years ago. I can't believe people are considering filling football fields with computers, with the environmental difficulties we face today.

Please, please try to think of your solutions in ways that can take advantage of the same resources that are so liberally spread about as 'client' computers. "Web 2.0" would make some kind of sense in a world where Thin Client computing had won ten years ago, but it didn't.

Now thanks to the ever adventurous PC gamer and the "invisible hand of the market" which dictates to Microsoft and Intel that they must collaborate to push out more faster chips to the same buyers over and over, we have a forest of 'server-class' equipment right under our noses. And yet some can't see the forest for the trees.

^ | v • Share >



**Randy H.** • 9 years ago

I work at Microsoft on the "enterprise" side of the house and I found this piece on Windows Live fairly enlightening. I have been in discussions with many of the old MSN folks about what it takes to scale our OS and how we build scalable systems for the web. To say that it a challenge is a huge understatement, just as many people underestimate the complexity of Google's infrastructure.

Considering this, I think that some of the commenters should not be too quick to judge Microsoft's ability to build scalable systems. There is a tremendous track record (certainly since Windows 2000) of high-scale

applications running on the Microsoft platform. From a price/performance perspective, Microsoft has been a leader in this area for many years. There is plenty of room to improve (always), but Microsoft has proof of scale in both the enterprise and the web. MSN and now Windows Live have significantly larger numbers of users than most services on the Web. While IBM and Oracle are cited for their expertise in scaling systems, to my knowledge neither IBM nor Oracle runs any system (including OS, application software, integration and management tools) on the scale of Hotmail, MSN Spaces or MSN Search anywhere in the world.

I'm certainly open to any examples that would prove my statement wrong, but I would encourage people to keep an open mind about Microsoft's expertise and technology in the area of scalable systems.

^ | v • Share >



**Tim O'Reilly** • 9 years ago

ssavitzky -- you miss the point. Of course Google is doing well. The question is whether the expertise they've build on top of open source is flowing back into the open source projects (specifically in the area of operations and deployment.) This isn't a question of whether Microsoft is advantaged against Google, but about whether they are advantaged against Red Hat because of the change in the application environment to massively hosted applications.

^ | v • Share >



**ssavitzky** • 9 years ago

The only advantage Microsoft really has in operations is that the operations staff can get at the source code for the OS they're deploying on top of. But that's an advantage only if their competitors are also using Microsoft software. For the most part they aren't.

Google and Livejournal, to give two examples I'm

familiar with, are built on top of an open source infrastructure and doing very well.

^ | v • Share ›



**Tim O'Reilly** • 9 years ago

rektide -- I think you hit the nail on the head. The OSS vendors have tended to keep their deployment tools proprietary value-add. Right after the original conversation with Ian Wilkes (linked above) that sparked this line of thought for me, I asked Marten Mickos about Ian's requests, and he said, "Yes, we have that. It's part of the MySQL Network." (i.e part of the commercial-only service offering.)

I should be clear that I have no moral objection to this strategy -- as I've long said, companies need to make their own choices about how to maximize the value of their work -- but I do question whether it will have long term consequences for the health of the open source ecosystem.

^ | v • Share ›



**rektide** • 9 years ago

HERE HERE!! Cheers Tim!

and where is open source? we've let RH and SuSE build their own deployment systems, and let OpenLDAP and Kerberos sit and rot unused. OSS has zilch for infrastructure. ironic that such a network oriented small-pieces culture would be so poorly interconnected. i loath the filthy bastard of AD, but the fact that there's still no linux deployable equivalent (build of real standards, of actual openldap&kerberos) is, well, downright bloody pathetic. and having done the kerberos+ldap dance, i can tell you, hand rolling it is not a pretty path.

and yes, dpkg needs more deployment oriented tooling. namely, more complex proxy servers.

^ | v • Share ›



**Bryan** • 9 years ago

I think a lot of people are missing the basic thrust of the article: **operations**.

Operations such as deploying, patching, and upgrading your app across many systems, maintaining a robust directory and authentication solution for all users, keeping platforms patched, performing backups & restores, detecting and responding to errors & hardware failures.

MS does have an advantage in these areas, given that it has standardized (but proprietary) solutions to all these operational issues. Justin Mason commented that many admins roll their own solutions to such issues - and if you're going for long-term sustainability & scalability, that's precisely the problem. You don't *\*want\** to have to support any more than the absolute minimum number of such hand-rolled operational solutions. Ideally you'd like to bring in operational personnel who *\*already\** understand how the basic block-and-tackle operations mechanisms work - not guys who need to be trained in your company's specific tinkertoy contraption.

^ | v • Share >



**Dominic Mitchell** • 9 years ago

I'm so glad that people are starting to talk about this. For a long while now, I've been extremely frustrated by the fact the *nobody talks about deployment*. "Elephant in the room" is a complete understatement. There are articles and books on how to write code until you're blue in the face. But the operational side of things is simply not discussed. What's worse, this leads to everybody reinventing things, badly. You'll note that the successful people tend to be those who have solved their deployment problems.

But it's worth noting deployment is a *really* difficult thing to solve in general. There are so many choices to be made which are absolutely specific to your circumstances. I think what's needed is some kind of "Deployment Patterns" book. To be read by both sysadmins and programmers and try to bridge the gap between the two.

I actually think that Open Source is in an extremely good position to deal with these issues. Because the

good position to deal with these issues. Because the code is open, it can be gently levered into any position to suit. Like all such things though, it needs somebody to step up and take the gauntlet.

^ | v • Share ›



**Jeff Carr** • 9 years ago

The fact is that Google has not been successful in creating a web-based enterprise software application yet, but neither has anyone else. 37 Signals won't even consider doing it, according to Jason in his post "Growing in vs. growing out"

(<http://37signals.com/svn/archi...>)

Frankly, only Microsoft appears to have the depth of talent and cash reserves to take on the enormous challenge of Web 2.0 Enterprise-scale software. Personally, I'm hoping like hell to be involved in it's birth. It's an exciting time to be in this industry, isn't it?

^ | v • Share ›



**Steve Loughran** • 9 years ago

Tim,

I felt operations were a barrier on a small project, let alone a big one, because once you have a split between dev and deployment, you have problems. Take a look at the "When web services go bad" presentation on my site from 2001, or search for "making web services that work", a paper I wrote at the same time.

Nowadays I work on grid-scale deployments using SmartFrog, which is OSS and ready for use today. It

---

[see more](#)

^ | v • Share ›



**Sid Steward** · 9 years ago

Tim- thanks for the thoughtful article. Must a developer run her app in *someone else's* data center? How about creating a data center cooperative? As a shareholder, the developer can feel better using *her own* data center. Seems to me like a natural extension of the free software culture.

*Cloudy with a Chance of Meatballs* is in our son's library. Great illustrations.

^ | v · Share ›



**Alex** · 9 years ago

Tim - You are right, she did not expressly reference Google in regards to advantage, but if she is not referencing some MS-specific software advantage then really isn't her argument kind of circular?

"Companies that have large server deployments are good at deploying large amounts of servers"

^ | v · Share ›



**Jnan Dash** · 9 years ago

Having spent 26 years at IBM (developing DB2) and at Oracle (server group) on servers for highly scalable systems, I am familiar with operational robustness issues. Microsoft claiming highly scalable systems seems a bit strange. They have spiritual commitment but are far behind the likes of IBM or Oracle in dealing with highly scalable systems. This SaaS (hosting highly scalable Web 2.0 applications) is like going back to the future, in terms of highly scalable, reliable, repeatable, measurable, 24x7 available, with high performance systems. Google, Yahoo, and Amazon have years of experience in building such scalable systems with Linux plus lots of proprietary glue. At Foldersa we deploy Oracle RAC as the database for high scalability even with a .Net framework.

^ | v · Share ›



**jbond** · 9 years ago

It very much depends on what you're trying to do. Take a look at something like Digg or Slashdot.

There are lots of us who would love to have their scaling problems and don't actually have ambitions to grow any larger. And sites like that are perfectly capable of being built and run on OS with quite modest hardware. And there's now quite a large body of people around who know how to manage that size of problem. Even something of the scale of MySpace is within reach and at least understandable.

But step it up to Amazon, eBay, Yahoo or a corporate processing operation like Amex or Fedex and you're in another game entirely. So great. Let Microsoft compete for that business and I can ignore the issues completely.

^ | v • Share ›



**Kevin Farnham** • 9 years ago

Having worked on what was intended to be a high volume business document archive and presentation product suite built using Microsoft technology, and struggling for a few years (2000-2002) to scale the application for thousands of users -- I was quite shocked to find out that [MySpace.com](http://MySpace.com) runs on Microsoft technology. At the end of my time working on the document archive project, I was bringing in .NET for new development, and it looked promising. About 6 months after I left, tests using the new software successfully supported some tens of thousands of simulated users, running on a 32-processor Unisys box running Windows 2000. It was quite a change from where we started, with me running to client sites because their system began to fail as the number of users approached 100.

---

see more

^ | v • Share ›



**Tim O'Reilly** • 9 years ago

Alex -- Debra didn't say they had an advantage over Google, but rather that players with the kind of world class operational infrastructure that the big internet