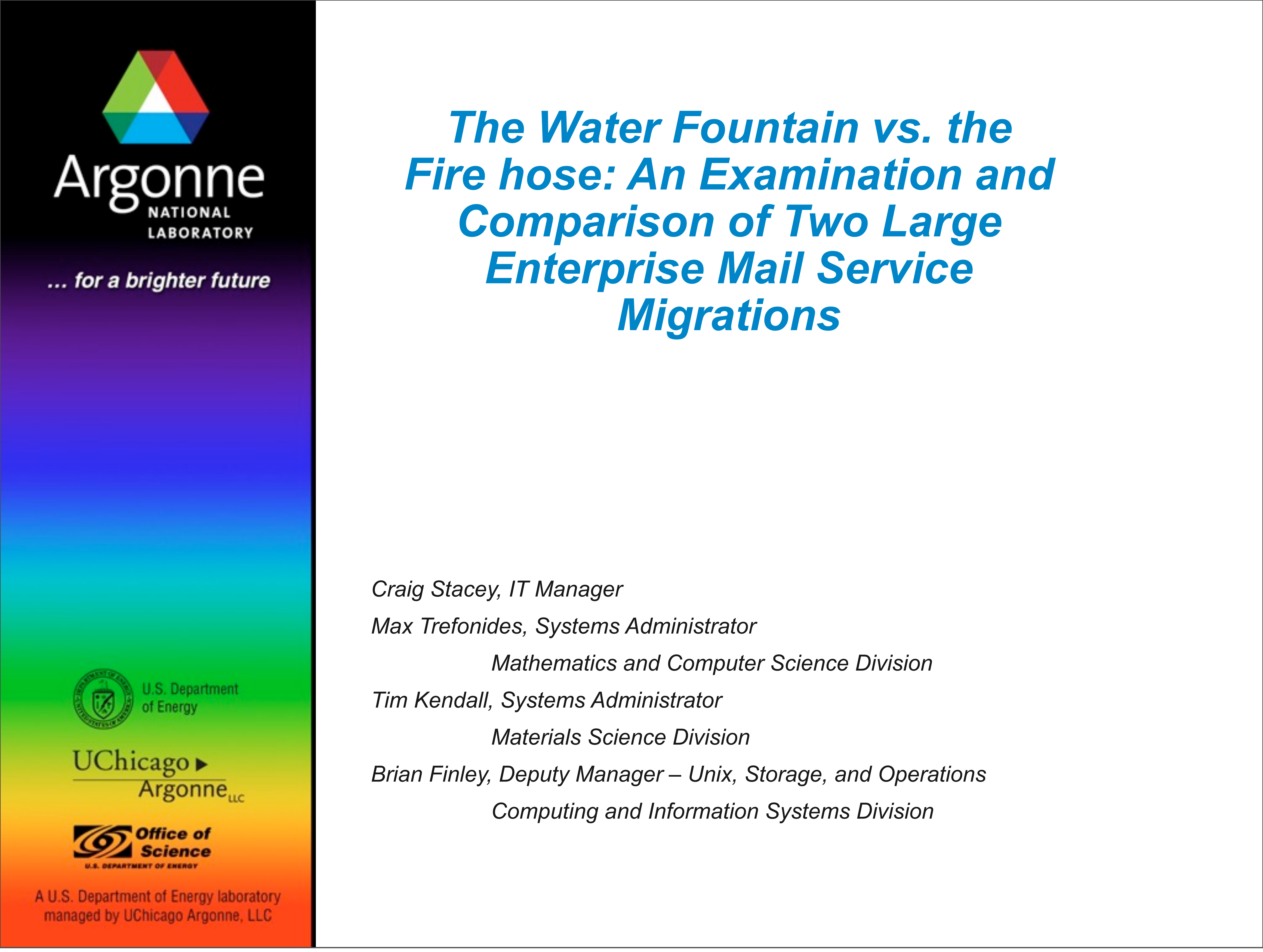


Argonne NATIONAL LABORATORY

... for a brighter future

The Water Fountain vs. the Fire hose: An Examination and Comparison of Two Large Enterprise Mail Service Migrations

Craig Stacey, IT Manager

Max Trefonides, Systems Administrator

Mathematics and Computer Science Division

Tim Kendall, Systems Administrator

Materials Science Division

Brian Finley, Deputy Manager – Unix, Storage, and Operations

Computing and Information Systems Division

U.S. Department of Energy

UChicago Argonne LLC

Office of Science U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory managed by UChicago Argonne, LLC

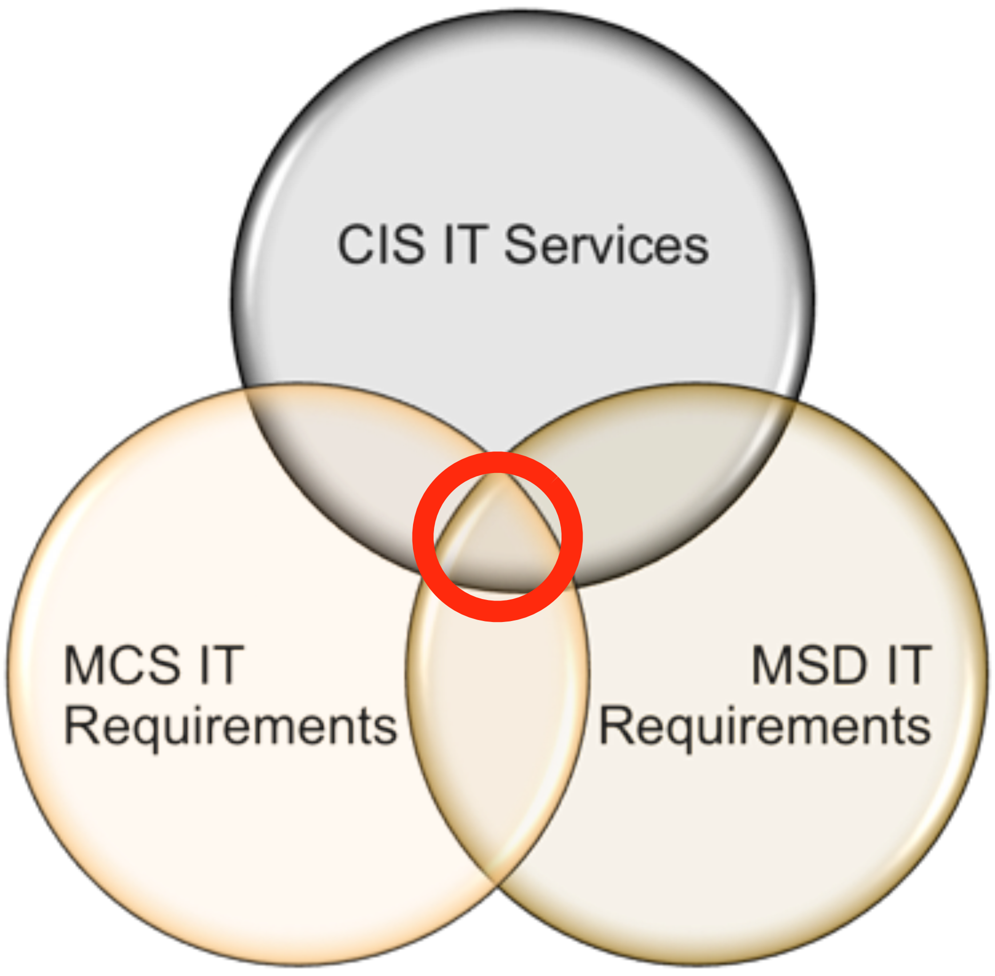
Introductions: Me, Max, Tim, Brian.

I know a number of stand-up comics, and there’s this truism that is known throughout the business. Every comic loves to hear a great “bomb” story. They just love to hear about some other comedian’s worst night on stage. It’s a kind of schaudenfraud that makes you feel better about your own situation. Kind of like how watching Springer or Cops makes everything in your world seem so much happier.

I’m happy to see this is alive and well in the world of systems administration! You all are about to hear to woeful tale of what led up to the worst weekend of my professional career as a sysadmin. So prepare to bask in the tale of poor decisions, rushed implementations, and mistyped config files, and be happy it wasn’t you!

Laboratory Overview of Services

- Central IT Services provided by Computing and Information Systems division
- Programmatic divisions often have IT needs outside this scope
- Occasionally, division-specific IT groups provide services that overlap with CIS.



Argonne's central IT Services group (CIS) provides services for both the Operations and Programmatic sides of the laboratory. Any of the lab's divisions and groups can use these services, typically with no additional cost. Of course, these services don't always overlap with the needs of the programmatic sides of the laboratory.

So, many of the Programmatic divisions also maintain their own IT staffs of varying sizes to support mission-specific computing needs. These groups will also, in some cases, provide general IT services.

Here we see simple Venn diagram that demonstrates two concepts in one. You can look at the three areas as the IT services provided by each group, or you can look at them as representing the IT needs of the group's customers. In either case, there's an overlap of services provided or needed. If we focus on this center intersection (click), we can find the service we're going to talk about today -- e-mail.

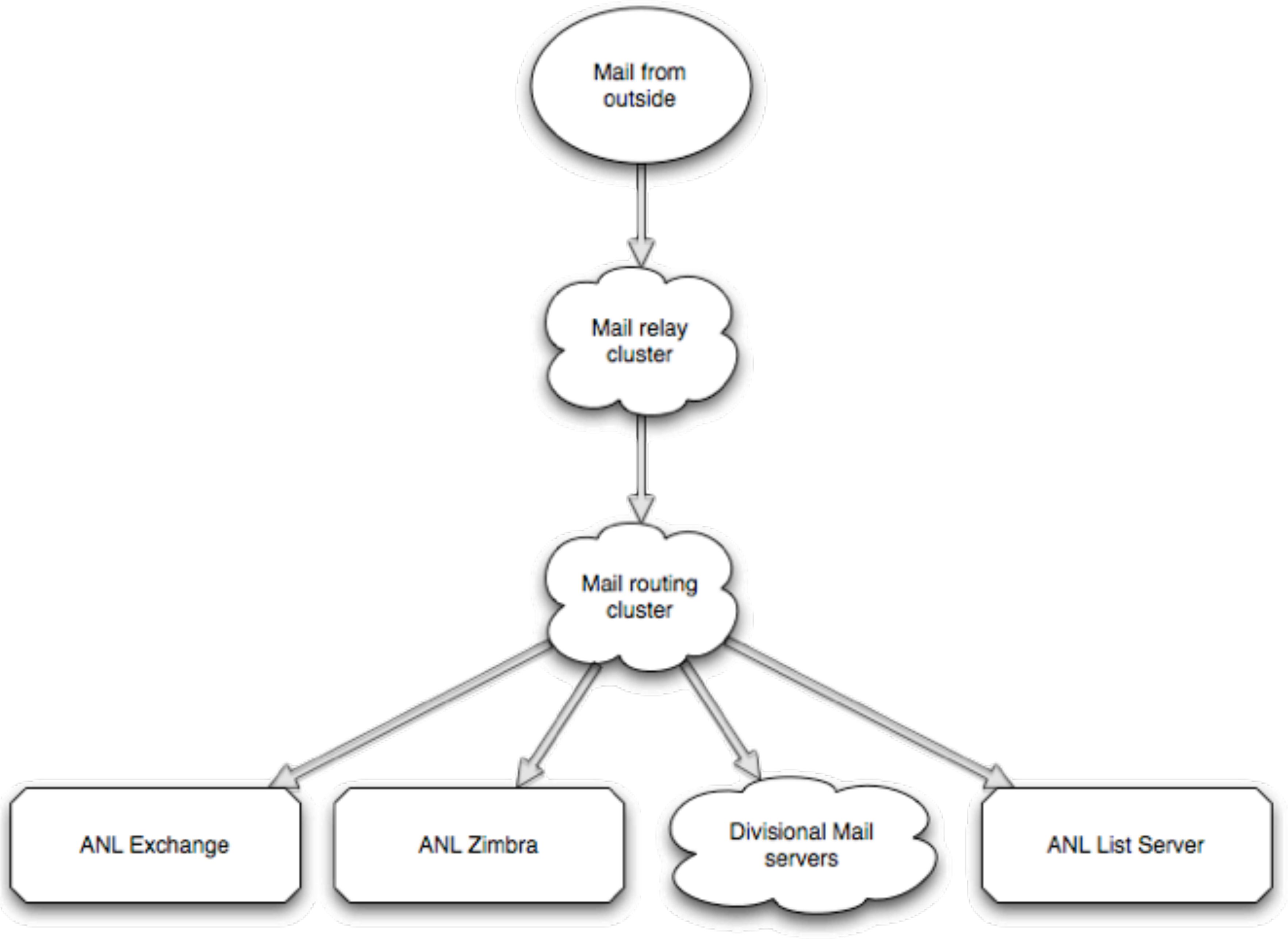
For clarity, when I say "operations", I'm referring to the business of running Argonne National Laboratory. The groups who are concerned with the day to day operation of the lab, and not involved in research. When I say "programmatic", I'm referring to the divisions who do the actual research funded by the various programs.

MCS (Mathematics and Computer Science) and MSD (Materials Science Division) are two such divisions, each has its own IT group, and each provided e-mail services for their users.

MCS maintains an IT staff of 7-12 people depending on what you consider IT staff and how many students we have.

MSD maintains an IT staff of 3 people.

Laboratory Mail diagram



Talking about e-mail, let's look at how things work at the lab, in general.

The lab's central mail service provides everything from external-facing mail relays to mailbox services.

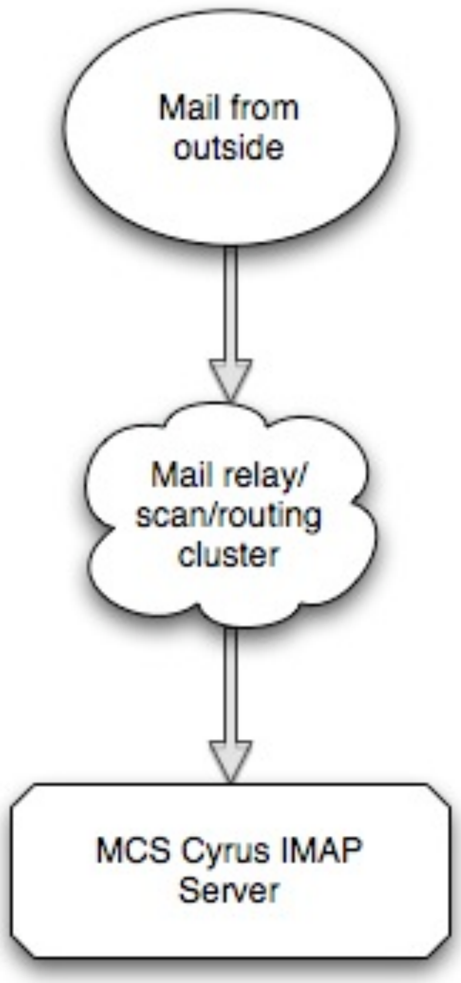
For a long time, the only real production mail service offered by CIS was Exchange, though in 2008 production-level Zimbra support was offered.

Mail is scanned at the relay cluster for spam and malware, then passed on to the routing servers for distribution to mailboxes, divisional mail servers, or list servers.

MCS Mail Migration

The title of this paper is the Water Fountain vs. the Firehose. We're going to talk about MCS's approach first, and, well, you can guess which approach we used.

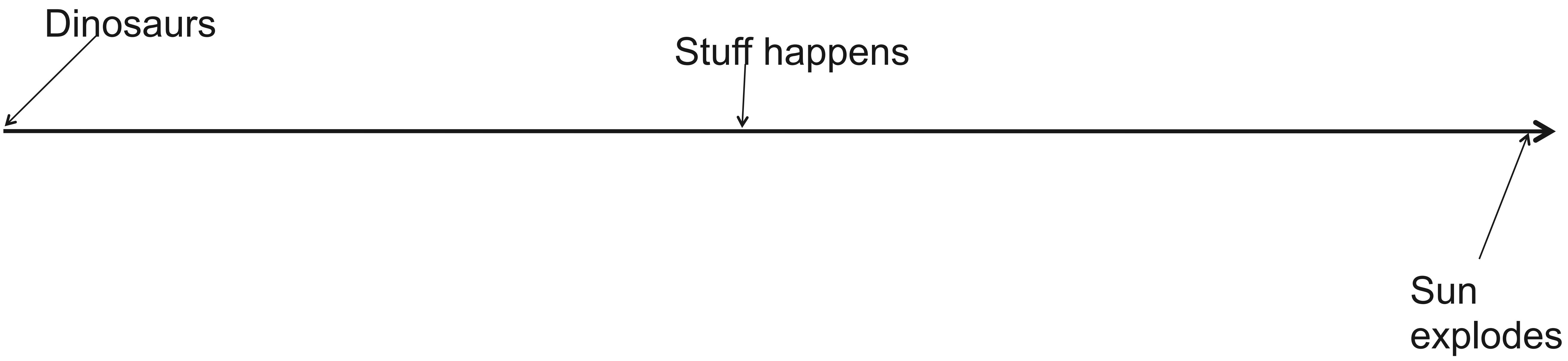
MCS Mail Delivery Overview & Diagram



MCS's mail infrastructure had always been historically outside the scope of ANL's mail system. At the time of the transition, this is how it looked. Pretty straightforward.

The key part of this diagram, and the focus of the next little bit of this talk is the bottom box, the IMAP server. It was an IBM RS/6000 PowerPC 604e AIX box, installed into production service in 1998, running an older version of Cyrus IMAP.

Timelines (the long view and the short view)



Dinosaurs

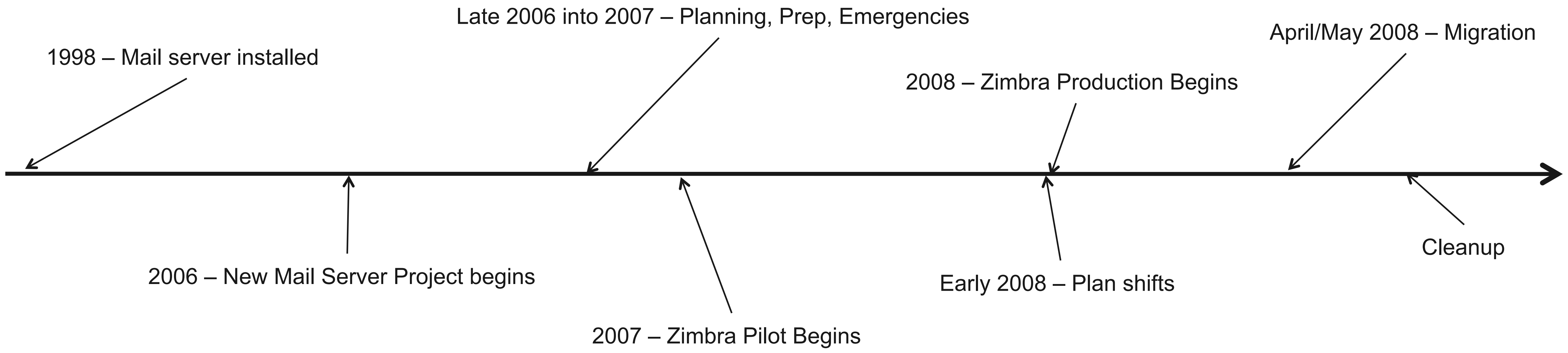
Stuff happens

Sun
explodes

Okay, too long.

We start with a long view of our timeline. <click> At the far left, <click> we have the Jurassic period, and on the right, the end times. The important stuff is in the middle. <click>

Timelines (the long view and the short view)



1998 – Mail server installed

2006 – New Mail Server Project begins

Late 2006 into 2007 – Planning, Prep, Emergencies

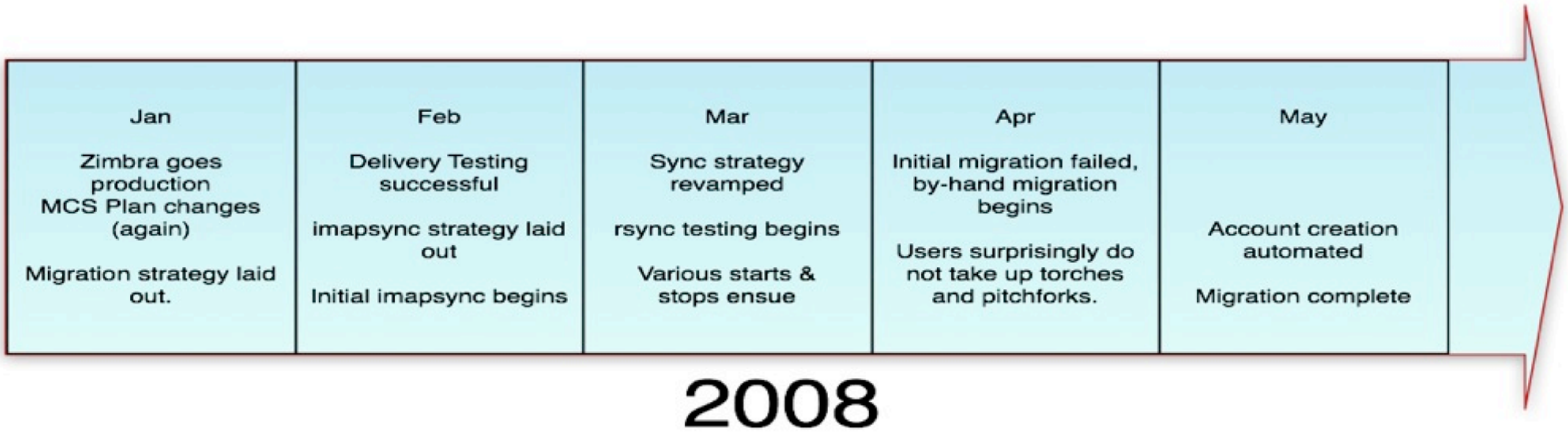
2007 – Zimbra Pilot Begins

2008 – Zimbra Production Begins

Early 2008 – Plan shifts

April/May 2008 – Migration

Cleanup



Jan	Feb	Mar	Apr	May
Zimbra goes production MCS Plan changes (again) Migration strategy laid out.	Delivery Testing successful imapsync strategy laid out Initial imapsync begins	Sync strategy revamped rsync testing begins Various starts & stops ensue	Initial migration failed, by-hand migration begins Users surprisingly do not take up torches and pitchforks.	Account creation automated Migration complete

2008

In 1998, we stood up a successful IMAP server. Too successful, it turns out, since it never really crept onto our radar again until 8 years later, when we began a new project to replace it. Unfortunately, emergencies kept interrupting the planning for this endeavor, and thus it still sat on the back burner. During this time, the Lab stood up a Zimbra server as a pilot program. We were intrigued by its group calendar functionality, as our users were requesting just this very service. And because it wasn't tied to running outlook, our heavy base of Linux users could make use of it.

By 2008, the pilot switched to production, and so did our planning. We switched our strategy to employ the Zimbra service for user mailboxes as well, and began a new plan on how to get the existing data from our old server (cliff) to the new one (zimbra). By this time, getting off the old mail server was getting higher and higher a priority. Services were failing, mailboxes were too big, and loads were climbing.

I'll go over this shorter timeline in the next few slides, but what I wanted to point out is how we'd gone from a very stretched timeline to a very compressed one. And as we'll learn later in the talk, it got even more compressed.

Research

Argonne National Laboratory

We did the research on this. All of our reading indicated this was going to be a simple operation. After all, it was all mailbox data, both the old and the new servers spoke IMAP. We could, with enough lead time, move all the data in advance of the switch to the new server.

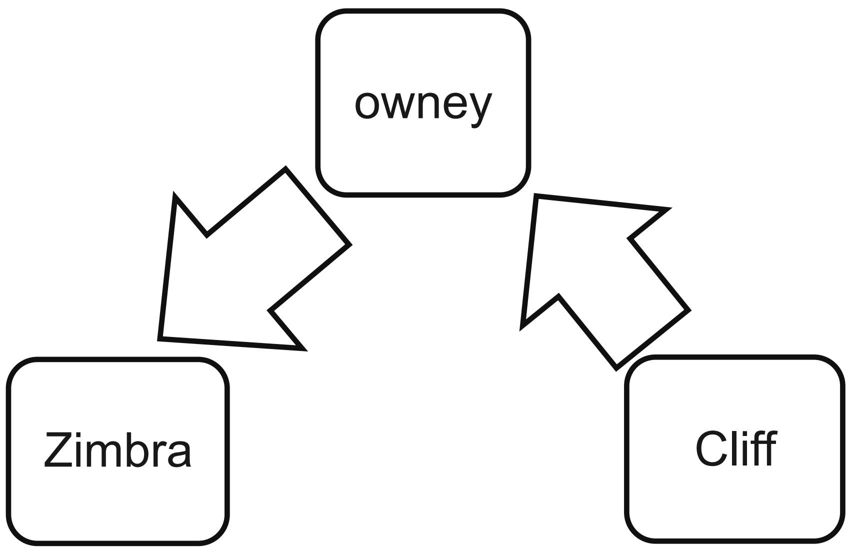
We would be heroes.

This was going to be simple.

I think we all know where this line of thought was heading.

The Plan

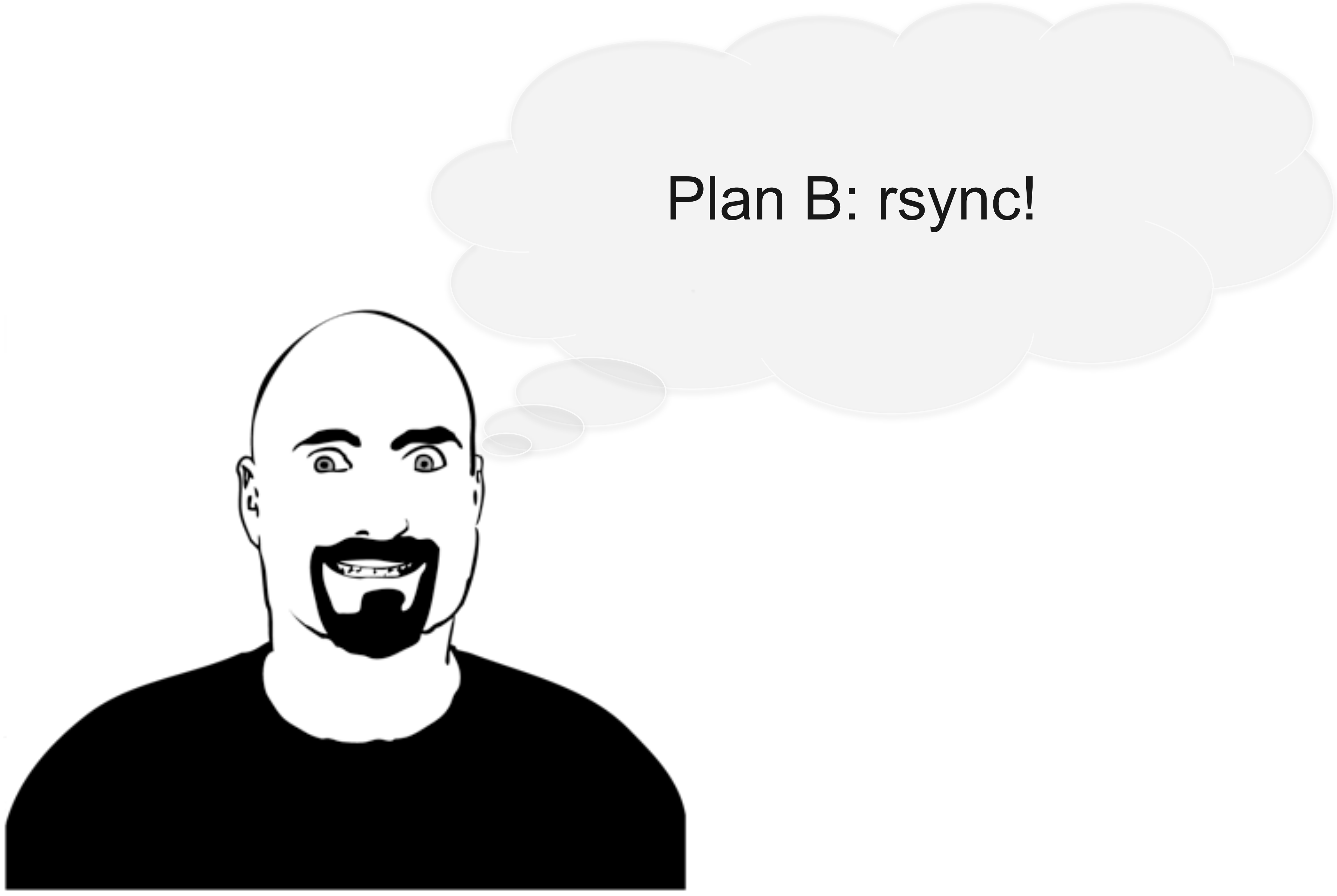
Plan A: Use
imapsync to move
user data.



Plan A was underway. We began the imapsync process. It was not without its pitfalls. First up, the age of the old mail server precluded us from running the imapsync scripts on it – its perl was old precluding it from being able to open an SSL IMAP connection on Zimbra. Also, it was overtaxed as it was, so we had a newer linux box handle the imapsync process. In order to avoid bringing the mail service to a crawl while we were working on the sync, we found the optimum number of concurrent syncs. Unfortunately, that number was two. Any more, and the mail server was slowing to a crawl or refusing connections.

Thus, it was a very slow process. Indications were that the actual sync would not finish in anything near an acceptable time period. So, while the syncs continued, we looked into other methods of moving the data.

The Plan



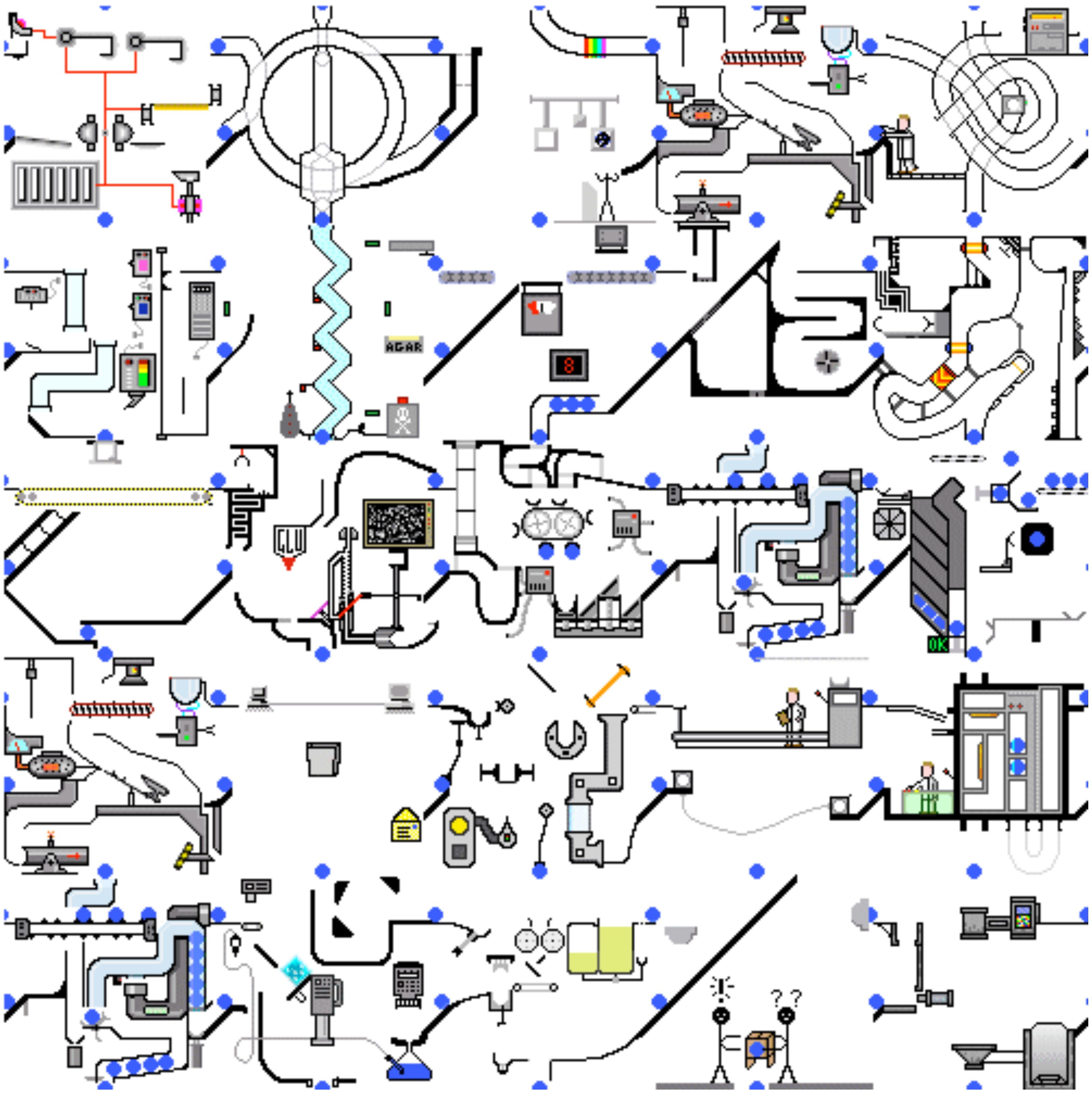
Thus was born Plan B. We would rsync the data from our mail server onto a disk the new server could mount, and then use zimbra's import tools to convert them into user mailboxes. After the data sync, we would use imapsync to get any new messages and set flags on **all** messages.

Because we didn't want to bog down the production zimbra service, we rsynced to a test server first and implemented our mailbox conversion tools on that. Once were were done, we would mount the disk on the production server, and perform the import there.

Surely, this plan could not fail.

Again with the learning.

Nothing is Easy



Despite what Staples would have you believe, there is no easy button. Everything is more complex beneath the surface.

We'd find some plan that worked on paper, but in implementation did not hold up as expected. A partial laundry list of failures:

•The disk to which we rsynced was slower than we expected. We felt the bottleneck was because the disk was NFS mounted instead of directly mounted. We planned to mount the disk directly on the production server as a logical volume from the SAN.

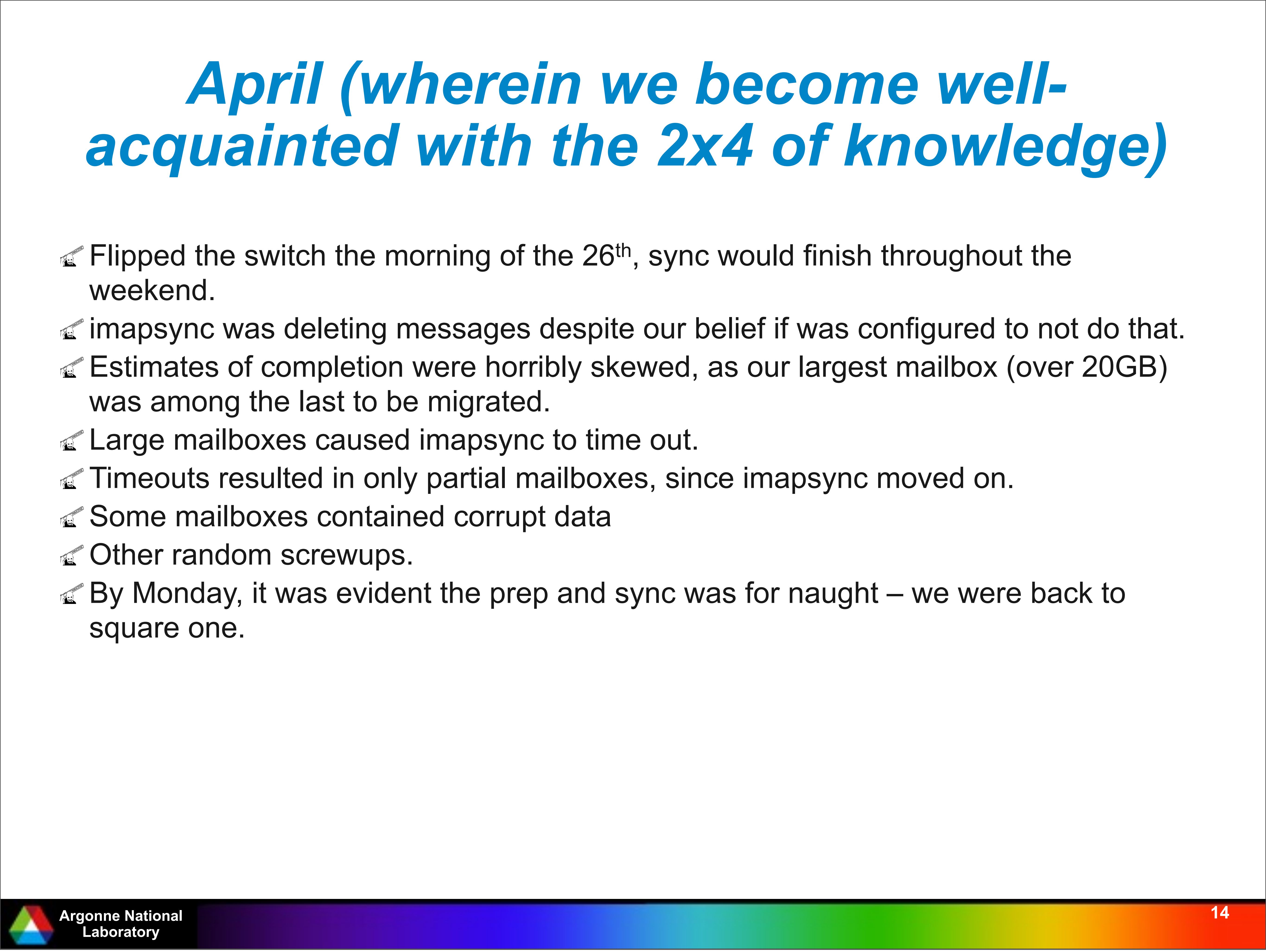
•Alas, the architecture of the SAN prevented us from doing this, requiring us to NFS mount this data on the production server as well.

•Some mailboxes produced name collisions with other mailboxes. Because Zimbra uses a flat namespace for its mailbox and calendar folders, you cannot have a mailbox and calendar with the same name.

•rsyncs were aborted and restarted at various points in the process due to filling disks, log files run amuck, network issues, machine crashes, etc.

•In the end, instead of a 4 month window in which to execute this transfer, various aborts and restarts put us into a situation wherein we had roughly two weeks to accomplish the migration and validation of data.

•It's worth noting here that the April 25th deadline was not completely arbitrary – it was already scheduled to be a maintenance weekend lab-wide, so users had an expectation of downtime. Also, our old mail server's ability to continue to provide service for more than another month was seriously doubted.



April (wherein we become well-acquainted with the 2x4 of knowledge)

- Flipped the switch the morning of the 26th, sync would finish throughout the weekend.
- imapsync was deleting messages despite our belief if was configured to not do that.
- Estimates of completion were horribly skewed, as our largest mailbox (over 20GB) was among the last to be migrated.
- Large mailboxes caused imapsync to time out.
- Timeouts resulted in only partial mailboxes, since imapsync moved on.
- Some mailboxes contained corrupt data
- Other random screwups.
- By Monday, it was evident the prep and sync was for naught – we were back to square one.

Starting from square one yet again, we had our method down. The rsynced data was freshly imported onto the production server, and our imapsync scripts were running, picking up the stragglers. Spot checks showed things were going as expected, though slowly. Users with large mailboxes were taking significantly longer than other users, but this was to be expected.

As the final weekend in April approached, we went into the morning of Saturday, April 26th with optimism that we'd overcome the pitfalls we'd been seeing.

<click> That morning, we discovered the imapsync was taking much longer than we'd hoped. We made the judgment call that we'd flip the delivery switch on delivery, get all the pieces in place, and continue the imapsync through the weekend. With all the pieces in place, mail was now being delivered into the new mailboxes, and we restarted the sync process.

<click> We were once again to be visited by the 2 x 4 of knowledge, as spot checking the logs later that night showed some behavior we should not be seeing. We had configured the imapsync script to be nondestructive – no mail would be deleted from the destination mailbox, only new messages would be added. However, it turns out that configuration was not working, and mail that had been delivered throughout the day would get deleted once that user's imapsync was run.

<click> Our estimation of when we would complete the syncs were skewed because they were based on alphabetical progress through the user list. Alas, our largest mailbox was the second-to-last mailbox to be synced, and many other of the largest mailboxes were skewed toward the end of the alphabet.

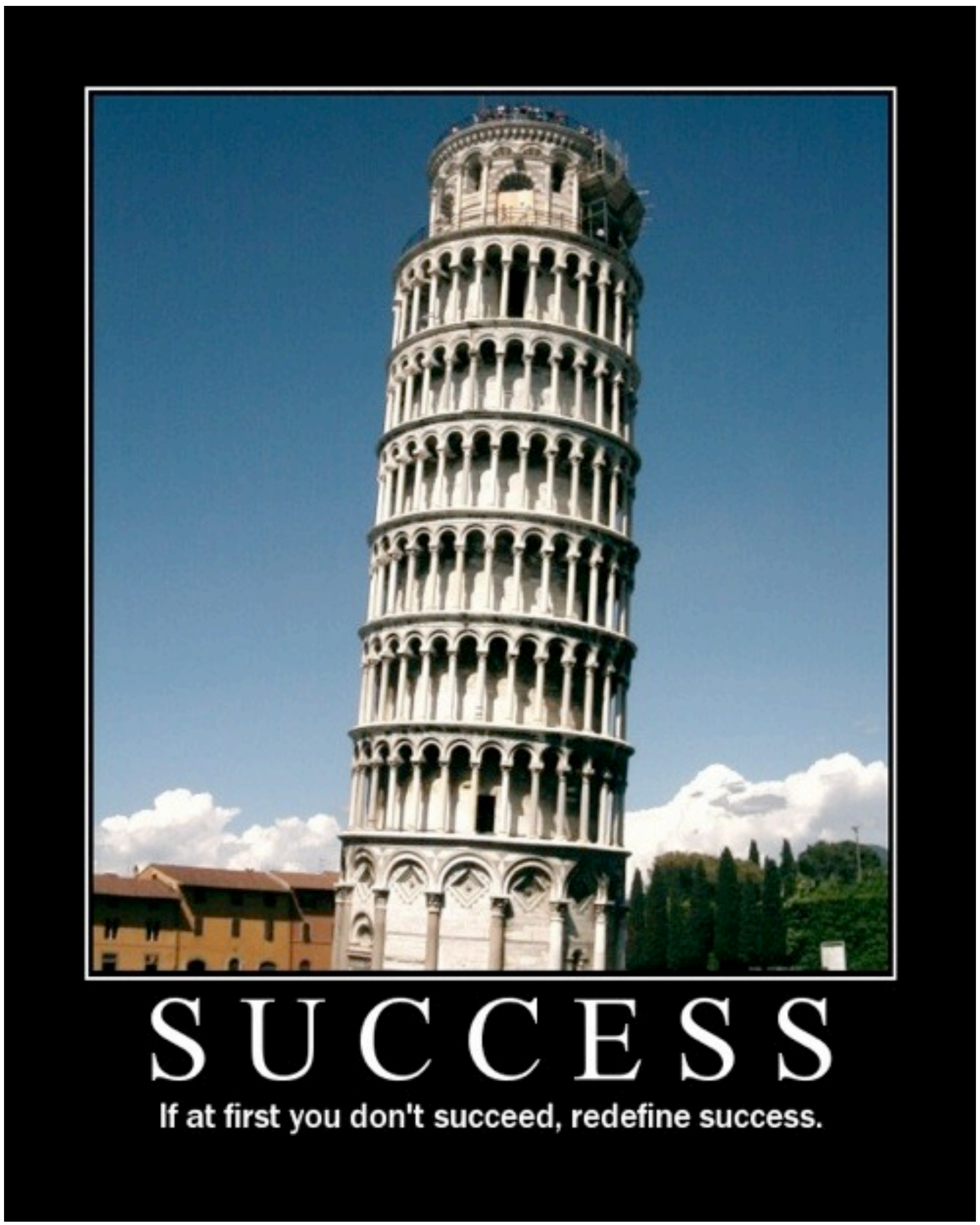
<click> IMAP would time out on the larger mailboxes, causing partial mailbox migration and flag setting, aborting the user early and moving on to the next one.

<click> Likewise, some mailboxes would only partially transfer as the sync would abort on the first corrupt message.

<click> We let users have access to their new mailboxes on Sunday evening by dropping a message in their old mailboxes containing instructions on how to reconfigure their mail clients. A long sleepless weekend led to a typo in that message, causing us to have to change configs such that the incorrect instructions would work.

<click> On Monday, after seeing the complaints, we sent notice to the users that the previously synced mailboxes were not likely to be current, and that we'd give them access to their old mailboxes so that they could move their data by hand. We would assist any user who wanted help.

May



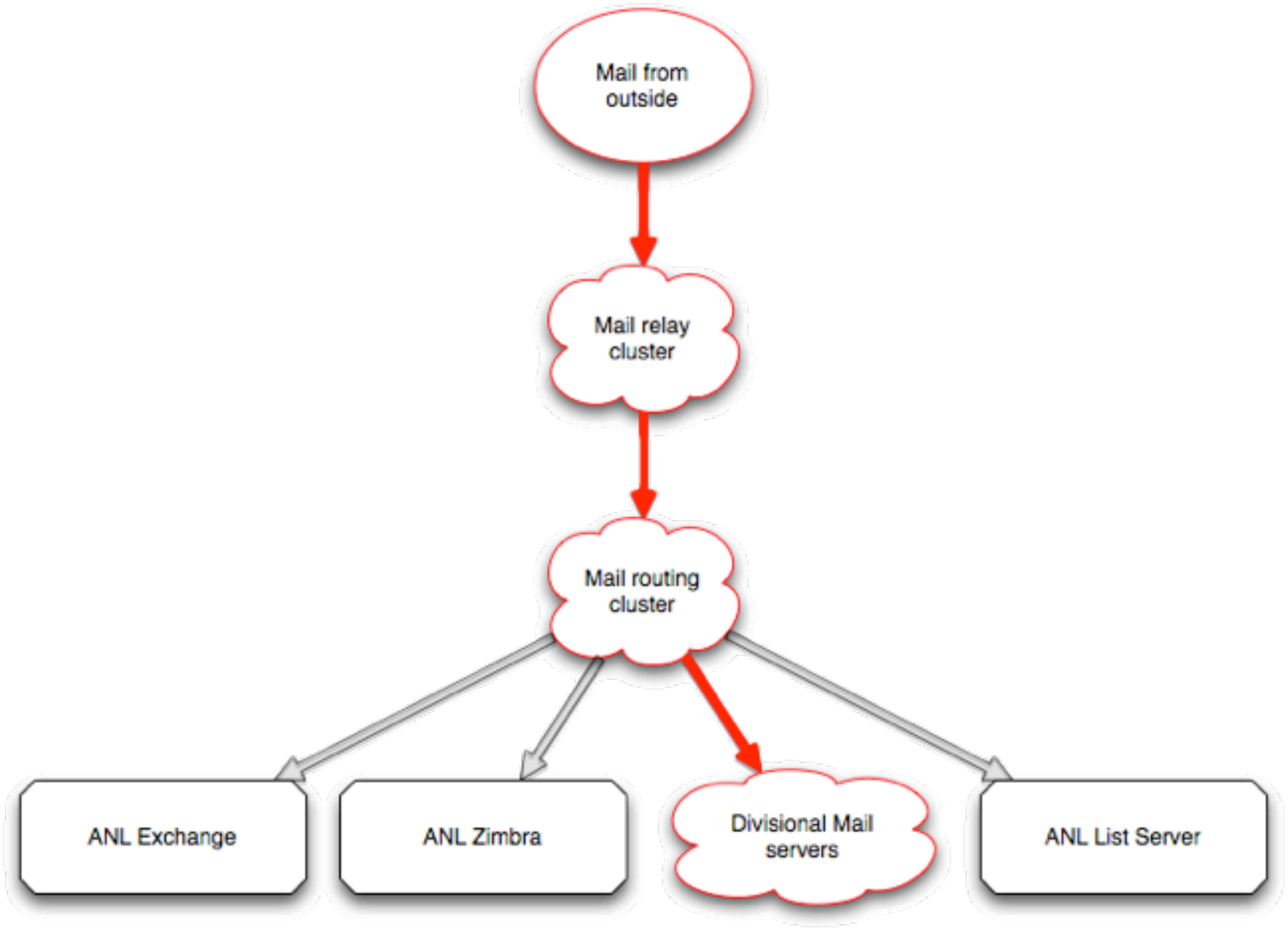
In May, we finished the migration. Largely, it's what we should have done from the start. Specifically, we should have simply switched delivery, and let the users move any mail they wanted to keep.

There were other lessons we learned after MSD's migration, which I'll get into later in the talk.

MSD Mail Migration

This part of the talk is much shorter, mainly because it went largely without incident.

MSD Delivery Overview



As noted here, MSD was largely using the lab’s mail infrastructure, just not for user mailboxes. Ultimately, all they were looking to do was switching from running their own mailbox server to using the lab’s services.

They had considered running their own mailbox server, but based off of MCS’s experience and faith in the new Zimbra service, they felt enough of a comfort level to go that route as well.

While a handful of users would ultimately end up on the Exchange server, the only real change in the “before and after” for this diagram is moving the final red arrow from the divisional mail server to the ANL Zimbra server.

The Importance of Learning From the Mistakes of Others

- Meetings were held prior to Zimbra decision.
- Discussions took place after MCS's migration.

After decades of development, Dr. Bradford finally reveals the world's first horseless carriage.

Prior to any migrations, MSD, MCS, and CIS got together at MSD's behest to discuss using Zimbra as a production mail service. At this point, MCS had already committed to moving to Zimbra, but had yet to perform the switch. We all got together again after MCS had finished its migration and discussed what we'd learned in the process.

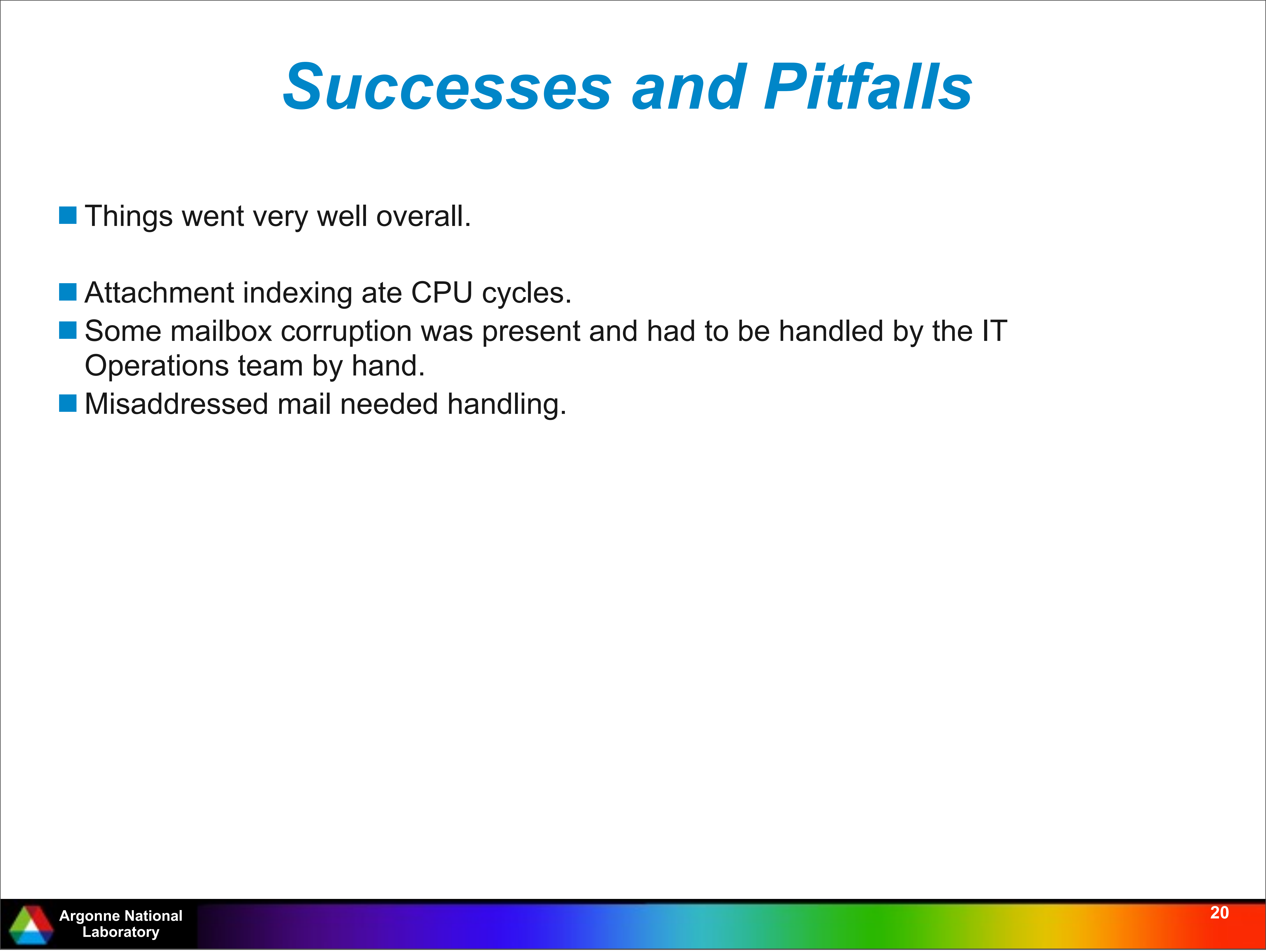
MCS unequivocally recommended a phased approach if at all possible, for obvious reasons.

The Plan

- Use new tools available in Zimbra to do the bulk of the heavy lifting.
  - Add external IMAP account to Zimbra account
  - Let Zimbra server slurp up the tasty messages
  - Drag imported mailboxes up into the main folder to preserve structure
- A minority of users were migrated to Exchange instead, to facilitate planning with groups outside the division who use Exchange.
  - These users were migrated piecemeal, by hand, and on a separate schedule.

In the intervening weeks between MCS's and MSD's migration, a new version of Zimbra appeared which contained the ability to download mail from other IMAP accounts. This new feature made MSD's path clear – create a mailbox on the new server, have the user add their old account to it, and let the server do the work. Once the mail was downloaded via IMAP, mailboxes could be trivially dragged from one account to the next, replicating the user's folder structure.

Beautiful in its simplicity, it allowed MSD to migrate users very gradually, at their own pace, largely without problem.



Successes and Pitfalls

- Things went very well overall.

- Attachment indexing ate CPU cycles.
- Some mailbox corruption was present and had to be handled by the IT Operations team by hand.
- Misaddressed mail needed handling.

The plan went off largely as expected, with some minor pitfalls.

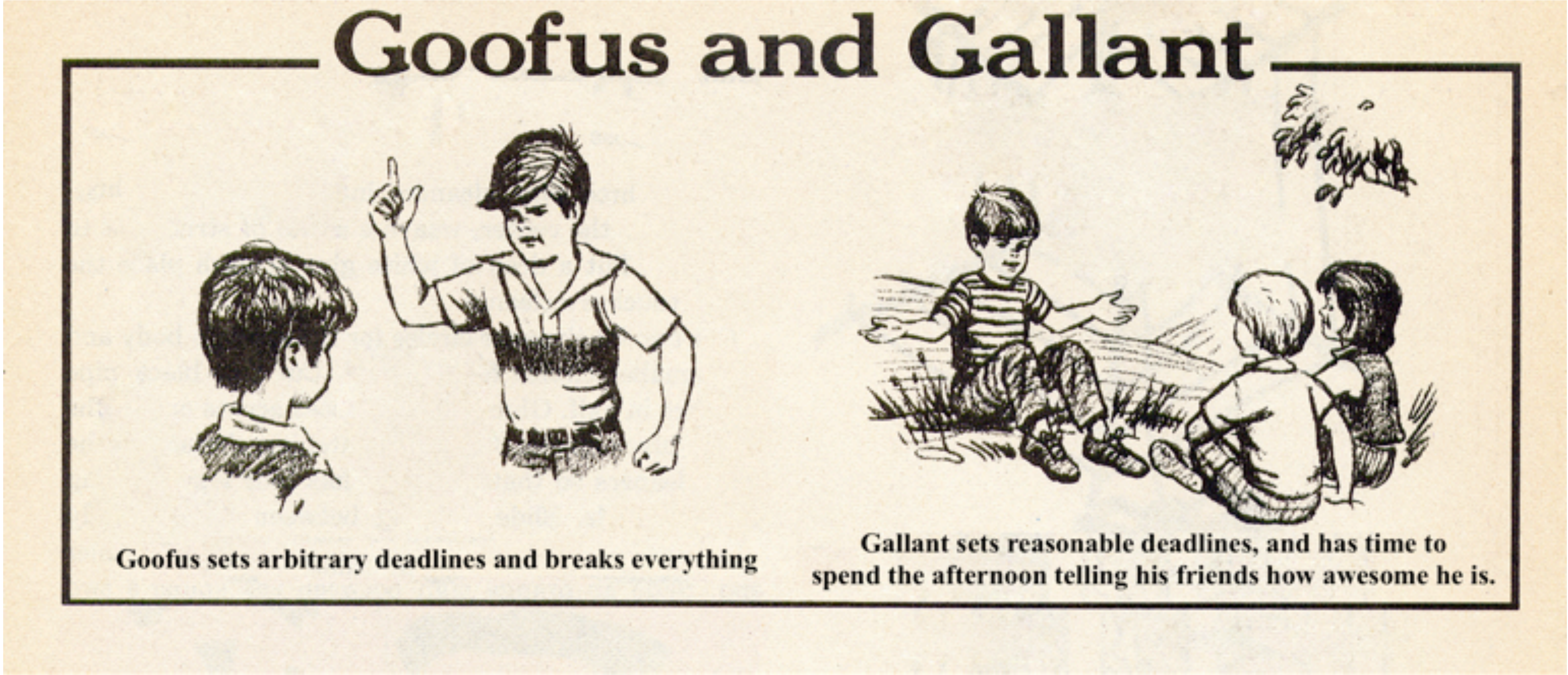
When MCS handled its migration, they were the only real users. Any negative impact on the Zimbra service was only going to be felt by MCS.

MSD joined in the fun after hundreds of mailboxes were already on the system. Other issues began to arise because of this. Some examples:

•Indexing of attachments was turned on, however as MSD added more and more users, we began to see load averages climb and the system bordered on being unusable for a time. Turning off attachment indexing resolved this issue.
•Some users had corrupt mailboxes, requiring intervention before their mailboxes could be imported.
•Some users were receiving mail addressed for the fully qualified divisional address instead of the @anl.gov alias. Because MSD does not run their own relay, these mails would not reach their users once the old mail server was turned off.

Generally, however, this was a success, and went pretty much as planned.

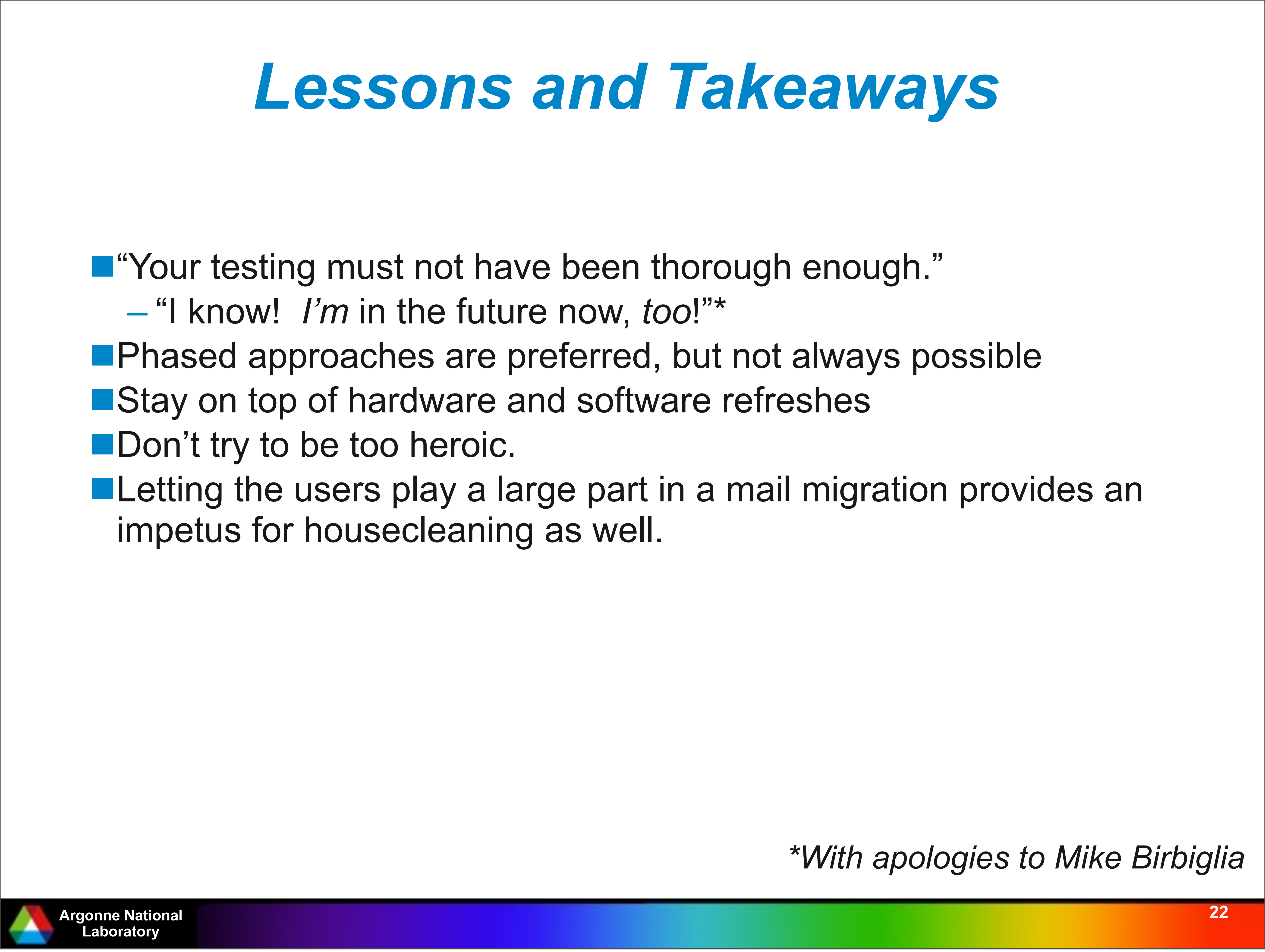
Comparing the two experiences


Goofus and Gallant
Goofus sets arbitrary deadlines and breaks everything
Gallant sets reasonable deadlines, and has time to spend the afternoon telling his friends how awesome he is.

Hindsight is 20/20. It's easy to say what we did wrong in MCS. From the surface, in fact, it would seem all we needed to do was wait it out and this new Zimbra feature would have made our work so much easier.

Deeper examination shows this not to be the case, however. The Zimbra tool MSD used relied on IMAP, the same way imapsync does. Testing after the fact confirmed that the problems we experienced with timeouts in imapsync were also present in this server-side IMAP poll from Zimbra. Certainly, smaller mailboxes would have been transferred this way very easily, but larger ones would still time out, and corrupt mailboxes would still need handholding.

Likewise our urgencies were different. MSD was battling against a filling disk, whereas in MCS we were looking at the possibility of our aging server not surviving the transition.



Lessons and Takeaways

- "Your testing must not have been thorough enough."
  - "I know! *I'm* in the future now, *too*!"*
- Phased approaches are preferred, but not always possible
- Stay on top of hardware and software refreshes
- Don't try to be too heroic.
- Letting the users play a large part in a mail migration provides an impetus for housecleaning as well.

**With apologies to Mike Birbiglia*

One of the comments we commonly received on this paper basically boils down to <click> "You obviously didn't have thorough enough testing." To which my response is <click>"I know. *I'm* in the future now, too!"

<click>It's not unexpected that a slower paced, staged migration is preferred. Alas, sometimes life does not provide this option, and you can occasionally be faced with a much more compressed schedule.

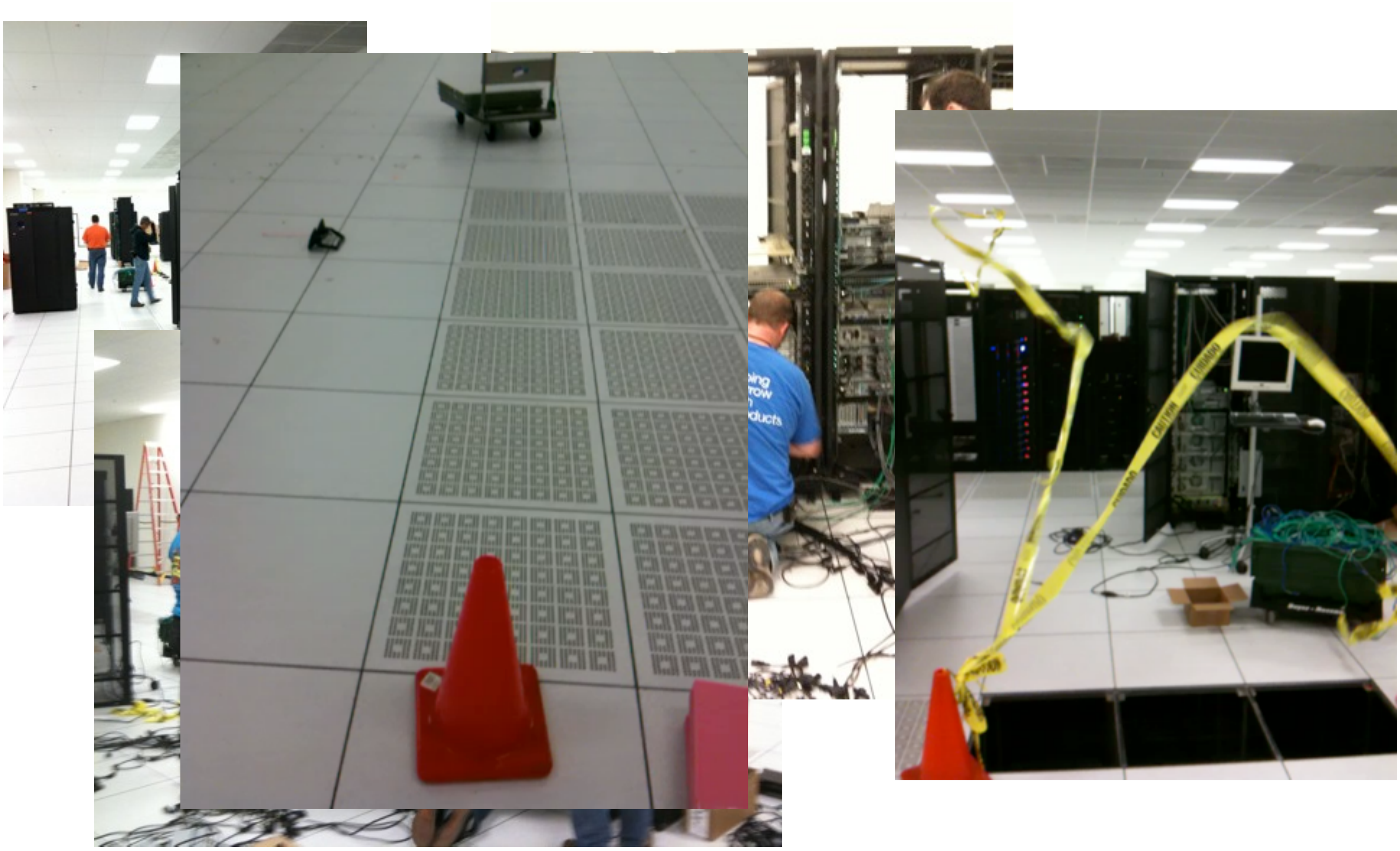
<click>The biggest lesson learned in MCS is to not let a well-functioning system allow you to be complacent in your hardware and software refreshes. Putting off the inevitable is only going to worsen the matter. In our case, it wasn't simply a complacency, but a larger confluence of events including limited funding, turbulent staffing changes, reorganizations, and, as always, a lack of time, that allowed the situation to reach the point it did. But the longer a system is left alone because it's running just fine increases the likelihood that it will grow roots and become far too embedded to be extricated smoothly. Regardless of budgets and purchases, stay on top of documentation, keep the care and feeding of all your systems in everyone's consciousness so no one piece gets left behind in a sea of undocumented kludges.

<click> Don't be a hero. We can get this Scotty on the Enterprise thing going, wherein we have the reputation of being a miracle worker. But, in reality, every once in awhile Scotty should have just told Kirk to cram it, or else the exceptional becomes the norm, and anything less is unacceptable. Have some faith in your users and their ability to deal with things.

I take every opportunity I can to brag about our users. We've got such a fantastic group of people we support in MCS, they genuinely understand Systems Administration is not easy, and that a large portion of what we do goes on under the hood and isn't noticed. This is most probably not universal, so it may not be a lesson for others to take away, but it is one for us.

<click> By eventually having the users lead their own migrations, we accomplished some cleanup as well. Certainly beneficial for us.

Epilogue & Datacenter Pr0n



- We took the lessons learned from this paper, and the other papers cited within, and applied it to a physical migration we just endured. We moved from a 3000 square foot datacenter to a shiny new 25,000 square foot facility. The move went very well, in a nicely staged manner. We even got to devise a prototype for datacenter air hockey.
- The mail server we were so afraid would crash in April of 2008 was turned off for good in October. It will never bother us again. This is its main CPU board, which will hang as a trophy of conquest in my office.