How to Build a Bad Research Center

David A. Patterson

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2013-123
http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-123.html

June 15, 2013

How to Build a Bad Research Center

David Patterson

June 15, 2013

Abstract

This paper contains my advice on building and running research centers based on being involved in a dozen of them over nearly 40 years. In keeping with my past advice pieces on talks and careers, where the goal was to be informative and entertaining, I first lay down eight commandments to create a *bad* research center before describing how to do the opposite: multidisciplinary, five-year centers that rely on biannual retreats and shared open space to guide the research.

Be forewarned that to persuade readers of the benefits of this approach I see no option but to cite impact on research results and on student and faculty careers I have witnessed in my decades at UC Berkeley. (Hint: the impact was positive.)

My intent is to publish this piece in a journal, but early feedback is that it will have to be less satirical and a lot shorter, hence this technical report to share the full history.

Introduction

A major center just completed,[i] so I finally have time to collect my thoughts on centers. I've been part of a dozen centers in computer systems, often as director (see Table I). By center, I mean a research project with at least three faculty, a dozen students, and a common vision. This piece is from the perspective of an academic in computer systems, but I hope it has wider applicability, even beyond academia. I do not advocate centers for *all* research; a lone researcher is best for many topics.

Why care about the Berkeley experience? Alas, establishing credentials is a lot like bragging so let me apologize in advance, which I'll also need to do later. The *US News and World Report* ranked universities for the computer systems field four times since 2002. In every survey, our peers rank UC Berkeley first. In addition, the National Research Council published a study of information technology research that led to multibillion-dollar industries.[ii] UC Berkeley was associated with seven such industries, more than any other university, primarily for its system projects.[1]

The Eight Commandments for a Bad Center

Following the template of my earlier pieces on "How to Give a Bad Talk"[iii] and "How to Have a Bad Career,"[iv] I offer Eight Commandments on "How to Build a Bad Research Center." I later suggest how to avoid bad centers.

Bad Commandment 1. Thou shalt *not* mix disciplines in a center.
It is difficult for people from different disciplines to talk to each other, as they don't share a common culture or vocabulary. Thus, multiple disciplines waste time, and therefore precious research funding. Instead, remain pure.

Bad Commandment 2. Thou shalt expand centers.
Expanse is measured geographically, not intellectually. For example, in the US the ideal is having investigators from 50 institutions in all 50 states, as this would make a funding agency look good to the US Senate.[2]

Bad Commandment 3. Thou shalt *not* limit the duration of a center.
To demonstrate your faith in the mission of the center, you should be willing to promise to work on it for decades. (Or at least until the funding runs out.)

Bad Commandment 4. Thou shalt *not* build a graven prototype.
Integrating results in a center-wide prototype takes time away from researchers' own, more important, private research.

---

[1] The seven industries (and Berkeley projects) are timesharing (Project Genie), client server (Berkeley Unix), VLSI design (SPICE), RISC processors (RISC), relational databases (Ingres), RAID disk arrays (RAID), and portable communication (Berkeley Wireless Research Center). Note that the founding of most industries required multiple research projects.

[2] Only the Big Science dreams of physicists have a chance of covering all 435 US congressional districts.

Bad Commandment 5. Thou shalt *not* disturb thy neighbors.
Good walls make good researchers; isolation reduces the chances of being distracted from your work.

Bad Commandment 6. Thou shalt *not* talk to strangers.
Do not waste time convening meetings to present research to outsiders; following the next commandment, reviews of your many papers supply sufficient feedback.

Bad Commandment 7. Thou shalt make decisions as a consensus of equals.
The US Congress is a sterling example of making progress via consensus.

Bad Commandment 8. Thou shalt honor thy paper publishers.
Researchers of research measure productivity by the number of papers and the citations to them[v]. Thus, to ensure center success, you must write, write, write and cite, cite, cite. If the conference acceptance rate is 1/X, then obviously you should submit at least X papers, for otherwise chances are that your center will not have a paper at every conference, which is a catastrophe.

Alternatives to a Bad Research Center[3]

Creating alternatives to bad centers requires breaking all eight commandments.

While I use my Berkeley experience for concrete examples, I polled the alumni from Table I and found that projects at CMU, Google, Harvard, Wisconsin, and UC San Diego are breaking these commandments as well.

Good Commandment 1. Thou *shalt* mix disciplines in a center.
Good Commandment 2. Thou *shalt* limit the expanse of a center.
Table II shows the multiple disciplines of each center in Table I. The rapid change in underlying technologies, and hence our fields, leads to new opportunities. While industry has more resources, it is often difficult for companies to innovate across conventional interfaces. The psychological support of others also increases the collective courage of a group. Multidisciplinary teams, which increasingly involve disciplines outside computer science,[vi] have greater opportunity if they are willing to take chances that individuals and companies will not. I believe in our fast moving fields there are more chances for impact *between* disciplines than *within* them.

Proposers often promise nearly anything to increase the chances of getting funding, with little regard to running a center should funding be procured. For example, many believe that including numerous faculty and institutions increases chances of funding. One excuse is simply the difficulty of evaluating research; as it takes time to judge center impact, there is little downside to proposing unwieldy centers. One study suggests following both of these good commandments. After examining 62 NSF-funded centers in computer science, the researchers found that multiple

---

[3] This section is based on Section 1.4, "Why Multidisciplinary Five-Year projects," from *The Berkeley Par Lab: Progress in the Parallel Computing Landscape*, D. Patterson, D. Gannon, & M. Wrinn, (eds) 2013.

disciplines increase chances of research success, while research done in multiple institutions—especially when covering a large expanse—decreases them:

> *The multi-university projects we studied were less successful, on average, than projects located at a single university. … Projects with many disciplines involved excelled when they were carried out within one university.* [vii]

A downside to multidisciplinary research is that it *does* take time to understand the differences in culture and vocabulary. If you believe multidisciplinary centers offer the best chance for having impact, however, then the benefits outweigh the costs.

Years	*Title*	*Profs: Director, Co-PIs*	*Number of Students (ACM Fellows)*
1977-1981	X-Tree: A Tree-Structured Multiprocessor	Despain, Patterson, Sequin	12 (Jim Goodman, Bart Miller)
1980-1984	RISC: Reduced Instruction Set Computer	Patterson, Ousterhout, Sequin	17
1983-1986	SOAR: Smalltalk On A RISC	Patterson, Ousterhout	22 (David Ungar)
1985-1989	SPUR: Symbolic Processing Using RISCs	Patterson, Fateman, Hilfinger, Hodges, Katz, Ousterhout	21 (Susan Eggers, Mark Hill, Jim Larus, David Wood)
1988-1992	RAID: Redundant Array of Inexpensive Disks	Katz, Ousterhout, Patterson, Stonebraker	16 (Peter Chen, Garth Gibson, Mendel Rosenblum, Margo Seltzer)
1993-1998	NOW: Network of Workstations	Culler, Anderson, Brewer, Patterson	25 (Mike Dahlin, Amin Vahdat)
1997-2002	IRAM: Intelligent RAM	Patterson, Kubiatowicz, Wawrzynek, Yelick	12
2001-2005	ROC: Recovery Oriented Computing Systems	Patterson, Fox	11
2005-2011	RAD Lab: Reliable Adaptive Distributed Computing Lab	Patterson, Fox, Jordan, Joseph, Katz, Shenker, Stoica	30
2007-2013	Par Lab: Parallel Computing Lab	Patterson, Asanovic, Demmel, Fox, Keutzer, Kubiatowicz, Sen, Yelick	40
2011-2017	AMP Lab: Algorithms, Machines, and People	Franklin, Jordan, Joseph, Katz, Patterson, Shenker, Stoica	40
2013-2018	ASPIRE Lab: Algorithms and Specializers for Provably optimal Implementations with Resilience and Efficiency	Asanovic, Alon, Bachrach, Demmel, Fox, Keutzer, Nikolic, Patterson, Sen, Wawrzynek	40

Table I. Patterson's Research Centers. The director is listed first in the 3rd column and the 4th column parenthetically names alumni who are ACM fellows. The CS faculty (thus far) is in the National Academy of Engineering: Brewer, Culler, Demmel, Jordan, Katz, Ousterhout, Shenker, Stonebraker, and myself. Demmel, Jordan, and I are in the National Academy of Sciences as well. Also in NAE are student alumni Eggers and Rosenblum. Alumni winners of ACM Doctoral Dissertation Awards include Gibson, Katevinis (RISC project), Rosenblum, and Ungar.

Good Commandment 3. Thou *shalt* limit the duration of a center.

My career has been based on five-year research centers, as Table I attests. This sunset clause arose from three observations:

- *To hit home runs, it's wise to have many at bats*. Fortunately, people remember research home runs and not the near misses. My experience has been that the chance for home runs is more a function of the number of research *projects* than of the *years* spent, so shorter projects give you more chances for success.
- *It's hard to predict information technology trends much longer than five years*. We start a center based on our best guess of what new opportunities will present themselves in 7 to 10 years, which is the right target for a 5-year research center. It's much, much harder to guess what the opportunities will be in 15 to 20 years, which you'd need for longer projects.
- *US Graduate student lifetimes are about five years*. It's much easier to run a center if there is no turnover of the people. As we are in academia, at the start each new project we recruit a new crop of students, as they will not graduate for five years.

You need a decade after a center finishes to judge if it was a home run. Just 8 of the 12 centers in Table I are old enough, and only 3 of them—RISC, RAID, and the Network of Workstations[4] center—could be considered home runs. If slugging .375 is good, then I'm glad that I had many 5-year centers rather than fewer long ones.

Title	*Prototype Disciplines*	*Applications*
X-Tree	Architecture, VLSI	--
RISC	Architecture, VLSI, ECAD, Compilers	--
SOAR	Architecture, ECAD, Compilers	--
SPUR	Compilers, OS, Architecture, VLSI, Circuits	--
RAID	Dependability, Architecture, OS, DBMS	--
NOW	OS, Architecture, Networking	Search Engine, Image Archive, HPC
IRAM	Architecture, VLSI, Compilers, OS	--
ROC	Dependability, SW Engineering	Satellite Tracking
RAD Lab	Networking, Machine Learning, OS	Twitter-clone, SAT vocabulary learning game
Par Lab	Patterns, Numerical Algorithms, Compilers, OS, Architecture, VLSI	Browser, Health, Music, Speech, Vision
AMP Lab	Machine Learning, OS, DBMS, Networking	Genomics, Participatory Sensing, Urban Planning
ASPIRE Lab	Patterns, Numerical Algorithms, Compilers, OS, Architecture, VLSI, Circuits	Genomics, Machine Learning, Graph Processing, Multimedia, Vision, SW Radio

Table II. The multiple disciplines of Table I. The Par Lab selected applications at the *start* of the project rather than as an afterthought, which AMP and ASPIRE labs imitated. The RAD Lab tried open space, which Par, AMP, and ASPIRE adopted. SPUR was the first with retreats, and did the rest.

---

[4] The NOW project showed clusters of workstations helped everything from encryption to sorting. The Inktomi search engine was built on NOW. The startup Inktomi Inc. in turn proved the value of clusters of many low cost computers versus fewer high-end servers, which Google and others in the Internet industry later followed.

A downside to sunset clauses is that it's easier to recruit students and sustain funding as a center grows in reputation than to recruit to or to fund a new center. However, if the goal is to maximize home runs versus recruiting or funding, in our fast moving fields it's better to declare victory after five years and look afresh for new opportunities, and then to recruit the right team to pursue them.

Good Commandment 4. Thou *shalt* build a center-wide prototype.
A common feature of the centers in Table I is the collective creation of a prototype that demonstrates the vision of the center, which helps ensure that the pieces are compatible. While systems students like building their part of the center vision, they do not necessarily want to work with others to make everything fit together. The educational power of such multidisciplinary centers comes from students of different backgrounds working together, however, as the experience improves their understanding and taste in research. This process also enhances students' system building skills by requiring them to do a real prototype, not just toy demos. Such prototypes can even lead to open source projects, which simultaneously aid technology transfer *and* expand the workforce building the prototype. In fact, open source success generally means more developers outside versus inside the center.

One downside of such centers is that faculty must convince students of the benefits of spending some of their time working for the common good versus working just on one's own piece. However, once they start seeing how the research of others benefits *their* results, they no longer require encouragement to do so.

Good Commandment 5. Thou *shalt* disturb thy neighbors.
Researchers have found that innovation is enhanced if all participants work in a single open space less than 50 meters across, as it encourages spontaneous discussions across disciplines.[viii] The goal is for the space to support concentration *and* communication. For example, for the last four centers of Table I, the faculty gave up their private offices to be embedded with students and postdocs in open space, where only the meeting rooms have walls.[ix] Faculty access draws students from their home offices to the lab, which increases chances of interactions.

A downside of shared space is the cost of remodeling to create an attractive open space. This is a one-time capital expense, since following projects will use it, but even so, the cost was equal to just two students over the life of a center. Shared open space is certainly more beneficial to a center than a few more students.

Good Commandment 6. Thou *shalt* talk to strangers.
A key to the success of our centers has been feedback from outsiders. Twice a year we hold three-day retreats with everyone in the center plus dozens of guests from other institutions. You can think of it as having a "program committee" that meets with everyone for six days a year for five years, or a month of feedback per center. Their value is indicated by 100% of the respondents in my alumni poll using them.

Having long discussions with outsiders often reshapes the views of the students *and* the faculty, as our guests bring fresh perspectives. More importantly, at the end of each retreat we get frank, informed, and thorough feedback from our valued guests. Researchers desperately need insightful and constructive criticism, but rarely get it. During retreats, we're not allowed to argue with feedback when it is given; instead, we go over it carefully on our own afterwards. In fact, we warn that we are taking their advice seriously, so be careful what you ask for.

Retreats fulfill other important roles:

- They provided real milestones, which are scarce in academia. It's one thing to promise your advisor you'll get something done by January, but quite another when you're scheduled to give a talk on it to respected visitors.
- They build esprit de corps, in that everyone in the center spends three days together. There is free time to play as well as work, so students in these centers often become lifelong friends. In fact, I was delighted to learn from the poll that perhaps due to their shared experience, there is even a sense of connection between the generations of systems alumni.
- Over their five-year center lifetimes, all students get ten chances to present posters or give talks, which gives them as much exposure as they desire *and* improves their communication skills.

A downside to retreats is again cost, which is about as much as one or two students. As before, exchanging retreats for a few more students would be ill advised. You also need to be careful which outsiders you invite to the retreat to be sure that they are really engaged and that they will offer useful and constructive criticism.

Good Commandment 7. Thou shalt find a leader.

To make progress, you need to find someone willing to spend the time to pull people together, to create a shared vision, to build team spirit, and in who investigators believe will make decisions in the best interest of the center. When I am the director[5], I am playing the game to be on the winning team rather than to be the most celebrated coach. Hence, I recruit faculty who are team players, leaving prima donnas for others to manage. I try to lead by example, working hard for the center's success while hoping that my teammates will be inspired to follow. My guiding motto is

> "There are no losers on a winning team, and no winners on a losing team."[6]

Good Commandment 8. Thou shalt honor *impact*.

While papers are surely associated with high impact research, they are not a direct measure of it. High impact research can even be difficult to publish. The first paper on the LLVM compiler, winner of the 2012 ACM Software Systems Award, was

---

[5] While I've been drafted several times, five others have taken the reigns in my 12 projects. In addition, many other Berkeley projects in which I didn't participate follow these commandments.

[6] Attributed to University of North Carolina basketball coach Dean Smith by his colleague Fred Brooks, Jr.

initially rejected by the main compiler conference (PLDI), and the first paper on the World Wide Web by Tim Berners-Lee was rejected by Hypertext '91[x]. Turing Award winner Jim Gray explained the phenomena in his note to Jim Larus about an initially rejected paper that was eventually published and cited 145 times[xi]:

> *Well, the cohort-scheduling paper is in good company (and for the same reason).*
> *The B-tree paper was rejected at first.*
> *The Transaction paper was rejected at first.*
> *The data cube paper was rejected at first.*
> *The five-minute rule paper was rejected at first.*
> *But linear extensions of previous work get accepted.*
> *So, resubmit! PLEASE!!!*

Nor are best paper awards necessarily predictors of high impact. One study counts citations of best paper awards, documenting the modest influence of most of them.[xii] For example, the conference program committee for the first MapReduce paper, which led to the 2012 ACM Infosys Award, named *two* other papers as best! [xiii] Test of time awards, given 10+ years after the conference, are a much better indicator of success—the first RAID paper won three[7] —but they are trailing indicators.

For my successful projects, the early indicators weren't papers or awards, but non-researchers trying to use our ideas. For RISC and NOW, the first adopters were startups or at least young companies. For RAID, it was hungry companies trying to gain market share. The market leaders were happy with the status quo; they developed related products only *after* others had commercial success.

Educational Impact of Centers

Some may be surprised to learn that centers can improve undergraduate education, which is a principle of research universities. Less than a decade after seeing how to pipeline microprocessors in the RISC project, universities taught undergraduates to design their own. To develop applications for the RAD Lab, we offered specialty courses to teach modern tools and development methods. We then reinvented our software engineering course, which has long been problematic in CS departments.[xiv] It grew quickly from 35 to 240 Berkeley students, and inspired massive open online courses[8] where 10,000 earned certificates in 2012. These classroom innovations led to three textbooks, which further expand educational impact. [xv] [xvi] [xvii]

Most faculty and students are only familiar with single-investigator research, so it is hard to convince them how much centers can enhance graduate education. Good research centers are really federations of subprojects that share a common vision, with each subproject having one or two professors and several students. Each professor is an expert in a different field, which facilitates multidisciplinary

[7] It received the ACM SIGMOD Test of Time Award in 2008, the ACM SIGOPS Hall of Fame Award in 2011, and the Jean-Claude Laprie Award in Dependable Computing from IFIP in 2012.

[8] CS169.1x and CS169.2x are MOOCs from UC Berkeley and EdX.

research. These subprojects give students the same faculty attention as single-investigator project, but in addition students:

- Learn invaluable lessons by working and negotiating with dozens of smart, hard working, and stubborn collaborators, some of who go on to notoriety.
- Acquire good taste in research that comes from working on a common prototype, which leads to clear-eyed dissertations.
- Get a user base for new tools and approaches. Harvesting this feedback improves tools, user interfaces, semantics, and learning how to build tools.
- Have senior students as role models in the open space, which can lead to explicit mentoring.[9] Imitating good examples can lead to unusually mature behavior by first year students; one even gave a talk at CMU.
- Have fewer feelings of loneliness that is all too common in the PhD process, due to the esprit de corps that comes from the open space and retreats.
- Can be re-energized by praise from retreat visitors, as they may discount faculty compliments as mere cheerleading to get students to work harder.
- If a center funds the majority of faculty's students, their advisors awake each day thinking about the collective work of their students, rather than having their thoughts scattered across N different centers for N students.

I can't be modest while espousing the successes of this model, so as I warned earlier, let me yet again apologize in advance. When DARPA cut the research funding to universities in the last decade[xviii], our pitch to companies was that we couldn't keep producing such sterling systems students without the funding for multidisciplinary centers. We now know that this argument was more influential than our research agenda when Google and Microsoft decided to fund the RAD Lab.[xix] [xx] Happily, our claim proved to be true. For example, one student from the first cohort who did their research exclusively in the open space of the RAD Lab came up with the idea for the Spark cluster-computing framework[xxi] after overhearing machine-learning students in the lab gripe about MapReduce. He has just graduated, and not only did he get job offers from these and all the other companies he visited, he got them from the top universities he interviewed with as well, including CMU, MIT, and Stanford.

In case you were curious, these projects do not just produce single superstars. For example, the Par Lab averaged three students per paper and each student averaged three first-author papers. More importantly, at the end-of-project party, Burton Smith opined: "These are the best graduate students I have ever seen." This is high praise from a Microsoft Fellow and NAE member. What Smith said in 2013 was obviously gratifying, but I was struck that this is exactly what Mark Weiser, the father of ubiquitous computing, said at the end-of-project event for SPUR in 1989.

This tradition of nurturing promising Berkeley systems students as part of multidisciplinary centers goes back even further, to at least 1965. Project Genie[xxii] created the timesharing system prototype Cal TSS, one of the first time-sharing

[9] Since not everyone finishes a PhD in five years, we still have senior role models for junior students in new projects.

systems, and the team modified the Scientific Data Systems 930 hardware to support paging and memory protection. It was later commercialized and sold as the SDS 940 computer, and operating systems influenced by Cal TSS include TENEX for the DEC PDP-10 and Unix. Project Genie's most impressive legacy, however, may be that it produced three Turing Award winners in systems: Butler Lampson, Chuck Thacker, and Ken Thompson. Indeed, the faculty leader of the project told me he started a company based on the project in part to keep the team together, because it was "the best group of graduate students I have ever seen."[xxiii]

Reflections on Five-Year, Multidisciplinary Research Centers

One person congratulated our success at producing great systems students by saying we must have a terrific admissions committee. As all the top schools get great students, I doubt they would reach the same heights if not for our center culture. I only know computing, but I think students in any discipline would flourish in these centers. As my alumni poll suggests, good centers are not limited to a few places.

If such centers are good for faculty as well as for their students, then by now we should be able to tell. One measure is election to NAE and the National Academy of Sciences: only 5 to 10 computer scientists are selected each year for NAE (with half from industry), and just 2 or 3 for NAS. Thus far, 9 of the 24 Berkeley CS faculty who worked in the centers of Table I are in NAE (≈40%), and 3 are also in NAS. In fact, the majority of the Berkeley CS faculty in NAE or in NAS worked on these projects. In case you were wondering, they were elected *after* joining these centers.

Whereas early computing problems were more likely to be solved by a single investigator within a single discipline, I believe that the fraction of computing problems requiring multidisciplinary teams will increase. If so, then learning how to build good centers could become even more important for next 40 years than it has been for the past 40.

Acknowledgements

I wish to thank everyone who worked with me on the projects listed in Table I and the following people who offered feedback on early drafts: Krste Asanovic, Luiz Barroso, Chris Celio, Krsital Curtis, Rich Clewett, Alan Fekete, Armando Fox, Mike Franklin, Jim Larus, Roy Levin, Yunsup Lee, Kari Pulli, Evan Sparks, Amin Vahdat, and Matei Zaharia.

References

[i] *The Berkeley Par Lab: Progress in the Parallel Computing Landscape*, D. Patterson, D. Gannon, & M. Wrinn, (eds) 2013.

[ii] *Continuing Innovation in Information Technology*, National Research Council Press, 2012.

[iii] D. Patterson, http://www.cs.berkeley.edu/~pattrsn/talks/BadTalk.pdf, 1983.

[iv] D. Patterson, http://www.cs.berkeley.edu/~pattrsn/talks/BadCareer.pdf, 1997.

[v] See http://en.wikipedia.org/wiki/Impact_factor

[vi] Patterson, D. "Computer Scientists May Have What It Takes to Help Cure Cancer," *New York Times*. December 5, 2011.

[vii] J. Cummings & S. Kiesler. "Collaborative research across disciplinary and organizational boundaries,'' *Social Studies of Science* 35.5 (2005): 703-722.

[viii] T. Allen & G. Henn (2006) *The Organization and Architecture of Innovation: Managing the Flow of Technology*, Butterworth-Heinemann, 152 pages

[ix] D. Patterson, (2009) "Viewpoint: Your students are your legacy,'' *CACM*, 52.3 30-33.

[x] http://ibiblio.org/pjones/blog/the-story-behind-the-hypertext-91-demo-page-and-unc-and-me/

xi J. Larus & M. Parkes, (2002) "Using Cohort-Scheduling to Enhance Server Performance," *Proc. 2002 USENIX Annual Technical Conference*, p.103-114, June 10-15.

[xii] Best Papers vs. Top Cited Papers in Computer Science, arnetminer.org/conferencebestpapers

[xiii] J. Dean & S. Ghemawat, (2004) "MapReduce: simplified data processing on large clusters," *Operating Systems Design & Implementation Conference*, San Francisco, CA.

[xiv] A. Fox & D. Patterson. (2012). "Viewpoint: Crossing the software education chasm." *CACM* 55(5):44-49.

[xv] J. Hennessy & D. Patterson. (2011) *Computer architecture: a quantitative approach, Fifth Edition*. Morgan Kaufmann.

[xvi] D. Patterson & J. Hennessy. (2013) *Computer Organization and Design: The Hardware/Software Interface*, *Fifth Edition*. Morgan Kaufmann.

[xvii] A. Fox & D. Patterson. (2013) *Engineering Software as a Service: An Agile Approach Using Cloud Computing*, *First Edition*. Strawberry Canyon.

[xviii] E. Lazowska & D. Patterson. (2005). An endless frontier postponed. *Science*, 308(5723), 757.

[xix] A. Eustace, Private Communication, 2013

[xx] J. Larus, Private Communication, 2013.

[xxi] M. Zaharia, M. Chowdhury et al. (2010) "Spark: cluster computing with working sets." *Proc. 2nd USENIX conference on Hot Topics in Cloud Computing*.

[xxii] P. Spinrad and P. Meagher. (2007), Project Genie: Berkeley's piece of the computer revolution, *Forefront*, Fall.

[xxiii] M. Pirtle, Private Communication, 1992.