Name: Ken Birman **Affiliation:** Cornell University, Dept of Computer Science **Address:** Prof. Kenneth P. Birman Dept of Computer Science (5119B Upson Hall) Cornell University, Ithaca NY 14853	**Preferred topics:** A. Programming Models for the Cloud B. Network Support for the Cloud

Current research activities related to Cloud Computing

Relative to topic A above (your numbers 3, 4 and 5):
I've spent the past year or so building a new cloud computing platform called Isis^2, although working without any funding for this specific topic, and with rather limited testing options. The goal is to make high assurance cloud computing practical tomorrow. Today, in contrast, it is easy to build flexible services, but they have very weak assurance properties: they scale, and recover quickly from crashes, but often give incorrect answers by any definition of correctness. I've targeted my new system towards developers familiar with Azure, Amazon EC2, or similar platforms. Isis^2 will be released under FreeBSD licensing early in 2011.

The idea underlying Isis^2 is to embed group computing into a modern language (I'm using C#) in a clean and elegant way, bringing consistency, fault-tolerance and security to programmers with a skill set typical of a first semester undergraduate taking a college level object-oriented programming class. The experience is a bit like GUI programming : one extends a framework. In addition to making it easy to replicate code or data and to run computations fault-tolerantly and in parallel, the system offers a new way to perform MapReduce styled queries interactively, with consistency and fault-tolerance semantics much stronger than available in MapReduce.

My work (obviously) runs contrary to CAP and other commonly accepted principles for cloud computing. This abstract is a bit short to explain the reasoning, but I believe that while CAP is correct (after all, Gilbert and Lynch have a CAP theorem), it may have overlooked an important technical option, namely IP multicast. By getting IP multicast to work in cloud settings, one can replicate data or synchronize actions at the speed of light: often, with a single UDP packet, irrespective of group size. My hypothesis is that this capability is a true game changer.

As an offshoot of this work, I've collaborated with Robbert van Renesse and Dahlia Malkhi on a study of consistency in settings where service membership adapts to load, as in the cloud. We call our model "Dynamically Reconfigurable Services" (DRS); it merges State Machine Replication with Virtual Synchrony. This has advantages over both virtual synchrony and reconfigurable Paxos; indeed, it yields a new protocol we call "virtually synchronous Paxos." A paper can be downloaded from Cornell (http://www.cs.cornell.edu/ken) or Microsoft Research. The model is constructive and we hope to use Cornell's NuPRL theorem prover to explore it using automated formal methods.

Relative to topic B above (your number 2)
As part of a consortium called NEBULA, funded under the NSF FIA program, my group is exploring ways of building better routers on behalf of cloud-hosted applications with strong assurance properties. Cisco is our main industry partner. My focus is on using Isis^2 on core Internet routers, to enhance availability (we're aiming for 5-nines or better), using replication (e.g. of the BGP service) to mask failures. A student has developed a novel way to use TCP to connect to such a fault-tolerant BGP service; it greatly reduces the overheads of prior non-breakable TCP approaches, and eliminates the complexities and resynchronization delays associated with BGP's so-called graceful restart. Of course this is just one of several ideas, but typical of what we're doing with the NEBULA team.

Science Basis for Highly Assured Cloud Computing
Ken Birman

My belief is that the widespread industry adoption of very weak consistency models for the cloud, a reflection of the CAP conjecture and theorem but also of the poor performance of cloud synchronization and data replication protocols, has given us a cloud computing technology with unacceptable consistency and security problems. In effect, today's cloud computing offers a grab bag of tools and tricks of the trade that do lead to scalable, highly available solutions, but in ways that deprive us of consistency, security (which is often dependent on consistency of underlying data) and even performance. The issue is highlighted if one thinks about a medical computing system, which must be reactive on the basis of accurate, current data. Today's cloud offers us reactiveness, but only if we accept that the behavior of the system will often reflect stale or inconsistent data.

As we look towards the future, it is impossible not to see a wave of mission critical applications shifting towards the cloud for economic considerations. This includes systems for banking, control of the smart power grid, medical computing (not merely records management but also active care of patients who may need continuous monitoring and continuous control of various devices, such as insulin pumps), military computing… the list is long. To realize the true potential of the cloud, we need to find ways to host these mission-critical applications on the cloud. Failing to do so poses a real threat: those applications could move to the cloud, "ready or not". So the imperative is clear.

To bridge the gap we need a *Science Basis for Highly Assured Cloud Computing*. This would include a theoretical infrastructure that embraces scalability, in distinction from today's distributed computing theories, which often implicitly assume that no "service" would ever have more than 3 to 5 members. It would include an engineering infrastructure that offers realizations of best of breed solutions in easily used forms. And we need to work hand-in-hand with domain specialists who are creating software for tomorrow's medical care systems, power grids and other critical uses, so that we can understand how to apply our ideas to their problems.

I am keen to play a role in this effort. This endeavor has the attributes of a science because it demands a principled, rigorous underpinning, that can be used predictively to answer questions about hypothetical systems, and that also offers compelling answers to today's open questions. For example, consider the very basic issue of *stability*. One can argue that CAP reflects an assumption that replicated data will often be stale and that synchronization will be impossibly slow. This in turn reflects (I believe) the assumption that IP multicast is not a useful option in cloud settings, hence all communication is via one-to-one unicast: a slow way to replicate data. And IP multicast, in turn, is ruled out mostly out of fear: fear that when used aggressively, it can trigger meltdowns and thrashing on the scale of the whole data center.

To calm such worries, we need a way to deploy IP multicast and yet to prove that the resulting solution will be stable and safe. But to do so, in a climate of fear, and with widespread acceptance of CAP, we face a scientific challenge. Needed is a convincing theory of scalable stability and safety, and convincing engineering demonstrations. Given practical ways to replicate at much higher speeds and with less communication cost, and reassurance that doing so would be safe, industry might be convinced to adopt a better (more consistent, more secure model), even if only because it would be a cheaper model to operate (one multicast replacing *n* unicasts).

This abstract is limited to one page, but I would argue that the same set of issues underly questions of security, because today's security tools depend on the consistency of the data on which they base decisions, and that data is often stale or outright incorrect. They have obvious implications for responsiveness. Thus, advances on the basic questions could ignite advances over a wide front.