Edge Analytics in the Internet of Things

High-data-rate sensors are becoming ubiquitous in the Internet of Things. GigaSight is an Internet-scale repository of crowd-sourced video content that enforces privacy preferences and access controls. The architecture is a federated system of VM-based cloudlets that perform video analytics at the edge of the Internet.

Mahadev Satyanarayanan
Carnegie Mellon University

Pieter Simoens
Ghent University

Yu Xiao
Aalto University

Padmanabhan Pillai
Intel Labs

Zhuo Chen, Kiryong Ha, Wenlu Hu, and Brandon Amos
Carnegie Mellon University

Many of the "things" in the Internet of Things (IoT) are video cameras, which are proliferating at an astonishing rate. In 2013, it was estimated that there was one surveillance cameras for every 11 people in the UK.[1] Video cameras are now common in police patrol cars[2,3] and almost universal in Russian passenger cars.[4] The emergence of commercial products, such as Google Glass and GoPro, point to a future in which body-worn video cameras will be commonplace. Many police forces in the US are now considering the use of body-worn cameras.[5] The report of the 2013 NSF Workshop on Future Directions in Wireless Networking predicts that "it will soon be possible to find a camera on every human body, in every room, on every street, and in every vehicle."[6]

What will we do with all this video? Today, most video is stored close to the point of capture, on local storage. Its contents are not easily searched over the Internet, even though there are many situations in which timely remote access can be valuable. For example, at a large public event such as a parade, a lost child

might be seen in the video of someone recording the event.[7] Surveillance videos were crucial for discovering the Boston Marathon bombers in 2013. In general, an image captured for one reason can be valuable for some totally unrelated reason. Stieg Larsson's fictional work, *The Girl with the Dragon Tattoo* (Alfred A. Knopf, 2010), embodies exactly this theme: a clue to solving a crime is embedded in the backgrounds of old crowd-sourced photographs.

This richness of content and the possibility of unanticipated value distinguishes video from simpler sensor data that has historically been the focus of the sensor network research community. The sidebar presents many hypothetical use cases for crowd-sourced video. An Internet-scale searchable repository for crowd-sourced video content, with strong enforcement of privacy preferences and access controls, would be a valuable global resource. Here, we examine the technical challenges involved in creating such a repository. In particular, we propose GigaSight, a hybrid cloud architecture that is effectively a content delivery network (CDN) in reverse.

GigaSight: A Reverse CDN

A key challenge for the cloud is the high cumulative data rate of incoming videos from many cameras. Without careful design, this could easily overwhelm metropolitan area networks (MANs) and ingress Internet paths into a centralized cloud infrastructure, such as Google's

Crowd-Sourced Video: Use Cases

Here we present some hypothetical use cases for crowd-sourced video.

Marketing and Advertising

Crowd-sourced videos can provide observational data for questions that are difficult to answer today. For example, which billboards attract the most user attention? How successful is a new store window display in attracting interest? Which clothing colors and patterns attract the most interest? Are there regional preferences?

Theme Parks

Visitors to places like Disneyworld can capture and share their experiences, including rides, throughout the entire day. With video, audio, and accelerometer capture, the recreation of rides can be quite realistic. An album of a family's visit can be shared via social networks such as Facebook or Google+.

Locating People, Pets, and Things

A missing child was last seen walking home from school. A search of crowd-sourced videos from the area shows that the child was near a particular spot an hour ago. The parent remembers that the child has a friend close to that location. She calls the friend's home and locates the child.

When a dog owner reports that his dog is missing, a search of crowd-sourced videos captured in the last few hours may help locate the dog before it strays too far.

Public Safety

Where are the most dangerous intersections, with an accident waiting to happen? Perhaps an accident hasn't happened yet, but it could just be a matter of time. Video analytics of intersections could lead to the timely installation of traffic lights.

Many other public safety improvements are also possible: uneven sidewalks that are causing people to trip and fall; timely detection of burned-out street lights that need to be replaced, new potholes that need to be filled, and icy road surfaces and sidewalks in need of immediate attention.

Fraud Detection

A driver reports that his car was hit while it was parked at a restaurant. However, his insurance claims adjuster finds a crowd-sourced video in which the car is intact when leaving the restaurant.

Other law and order opportunities abound. For example, when a citizen reports a stolen car, his description could be used to search recent crowd-sourced video for sightings of that car to help locate it.

---

large datacenters or Amazon's Elastic Compute Cloud (EC2) sites. In 2013, roughly one hour's worth of video was uploaded to YouTube each second. That corresponds to 3,600 concurrent uploads. Scaling well beyond this to millions of concurrent uploads from a dense urban area will be difficult. Today's high-end MANs only have a bandwidth of 100 Gbps. Each such link can support 1080p streams from only 12,000 users at YouTube's recommended upload rate of 8.5 Mbps. A million concurrent uploads would require 8.5 Tbytes per second.

To solve this problem, we propose GigaSight, a hybrid cloud architecture that uses a decentralized cloud computing infrastructure in the form of virtual machine (VM)-based *cloudlets* (see Figure 1).[8] A cloudlet is a new architectural element that arises from
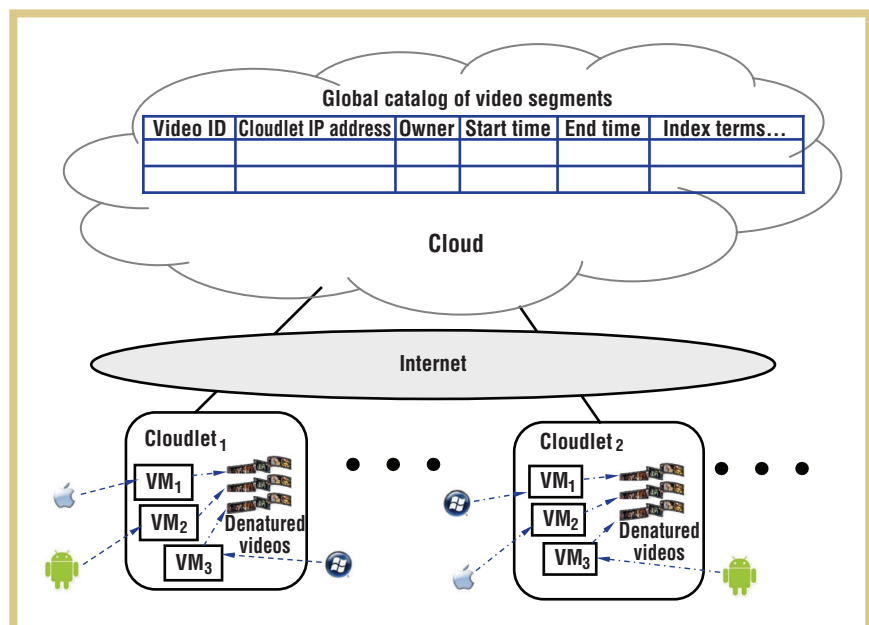

Figure 1. The GigaSight architecture. It uses a decentralized cloud computing infrastructure in the form of VM-based cloudlets.

the convergence of mobile computing and cloud computing. It represents the middle of a three-tier hierarchy: mobile device, cloudlet, and cloud. A cloudlet can be viewed as a "data center in a box" that "brings the cloud closer." Although cloudlets were originally created to address end-to-end latency in interactive applications, the use of cloudlets in GigaSight is based solely on bandwidth considerations.

In the GigaSight architecture (described in detail elsewhere[9]), video from a mobile device only travels as far as its currently associated cloudlet. Computer vision analytics are run on the cloudlet in near real time, and only the results (recognized objects, recognized faces, and so on), along with metadata (such as the owner, capture location, and timestamp) are sent to the cloud. The tags and metadata in the cloud can guide deeper and more customized searches of the content of a video segment during its (finite) retention period on a cloudlet.

An important type of "analytics" supported on cloudlets is automated modification of video streams to preserve privacy. For example, this might involve editing out frames or blurring individual objects within frames. What needs to be removed or altered is highly specific to the owner of a video stream, but no user has time to go through and manually edit video captured on a continuous basis. This automated, owner-specific lowering of fidelity of a video stream to preserve privacy is called *denaturing* (discussed further in the next section).

It is important to note in Figure 1 that cloudlets are not just temporary staging points for denatured video data in route to the cloud. With a large enough number of cameras and continuous video capture, the constant influx of data at the edges will be a permanent stress on the ingress paths to the cloud. Just buffering data at cloudlets for later transmission to the cloud won't do. Because video will be streaming 24/7, there will never be a "later" when ingress paths

are unloaded. The potential bandwidth bottleneck is at the access and aggregation networks and not in the core network with its high-speed links.

Preprocessing videos on cloudlets also offers the potential of using content-based storage optimization algorithms to retain only one of many similar videos from colocated cameras. Thus, cloudlets are the true home of denatured videos. In a small number of cases, based on popularity or other metrics of importance, some videos can be copied to the cloud for archiving or replicated in the cloud or other cloudlets for scalable access. But most videos will reside only at a single cloudlet for a finite period of time (typically on the order or hours, days, or weeks). In a commercial deployment of GigaSight, how long videos remain accessible will depend on the storage retention and billing policies.

Note that Figure 1 is agnostic regarding the exact positioning of the cloudlets in the network. One option is to place numerous small cloudlets at the network edge. An alternative is to place fewer but larger cloudlets deeper in the network—for example, at the metropolitan scale. Our analysis suggests that small cloudlets close to the edge is the better alternative.[9]

Denaturing

Denaturing must strike a balance between privacy and value. At one extreme is a blank video: perfect privacy but zero value. At the other extreme is the original video at its capture resolution and frame rate. This has the highest value for potential customers, but it also incurs the highest exposure of privacy. Where to strike the balance is a difficult question that is best answered individually, by each user. This decision will most probably be context-sensitive.

Denaturing is a complex process that requires careful analysis of the captured frames. State-of-the-art computer-vision algorithms enable face detection, face recognition, and object

recognition in individual frames. Activity recognition in individual sequences is also possible. However, preserving privacy involves more than blurring (or completely removing) frames with specific faces, objects, or scenes in the personal video. By using other objects in the scene, or by comparing videos from other users taken at the same place or time but with different privacy settings, we might still deduce which object was blurred and thus of value to the person who captured the video. The user's denaturing policy must also be applied to videos that were captured by others at approximately the same time and place. Simply sending the denaturing rules to the personal VMs of other parties is undesirable; this would expose at a metalevel the sensitive content.

One possible solution, proposed by Jianping Fan and colleagues,[10] is to send weak object classifiers to a central site where they are combined with a global concept model. This model could then be returned to the personal VMs. Of course, this approach requires videos to be temporarily saved in the personal VM until the central site has received any video uploaded at the same time and place. Any video that is uploaded later could then simply be discarded, to avoid keeping videos in the personal VM for too long.

In its full generality, denaturing might involve not only content modification but also metadata modification. For example, the accuracy of location metadata associated with a sequence of video frames might be lowered to meet the needs of *k-anonymity* in location privacy.[11] Whether the contents of the video sequence will also have to be blurred depends on the video's visual distinctiveness—a scene with the Eiffel Tower in the background is obviously locatable even without explicit location metadata.

In the future, guidance for denaturing might also be conveyed through social norms that deprecate video capture of certain types of scenes or in certain

locations. A system of tagging locations or objects with visual markers (such as QR codes) could indicate that video capture is unwelcome. We can imagine video capture devices that automatically refrain from recording when they recognize an appropriate QR code in the scene. In addition, denaturing algorithms on a cloudlet might also strip out scenes that contain such a code. In the long run, we can envision the emergence of an ethics of video capture in public. Many broader societal issues, such as the ability to subpoena captured but encrypted video, add further complexity. Clearly, denaturing is a deep concept that will need time, effort, and deployment experience to fully understand. GigaSight opens the door to exploring these issues and to evolving a societally acceptable balance between privacy and utility.

In the GigaSight prototype, a personal VM on the cloudlet denatures each video stream in accordance with its owner's expressed preferences. This VM is the only component, apart from the mobile device itself, that accesses the original (nondenatured) video. Figure 2 illustrates the processing within a personal VM. Denaturing is implemented as a multistep pipeline.

In the first step, a subset of the video frames is selected for actual denaturing. Our initial experiments showed that denaturing is too compute-intensive to perform at the native video frame rate.[9] Consequently, the denaturing process results in two output files: a low frame-rate "thumbnail video" file that provides a representative overview of video content for indexing and search operations, and an encrypted version of the original video. Both outputs are stored on the cloudlet, outside the personal VM. The encryption of the full-fidelity video uses a per-session AES-128 private key that is generated by the personal VM. If a search of the thumbnail video suggests that a particular segment of the full-fidelity video might be of interest, its personal VM can be requested to decrypt and denature
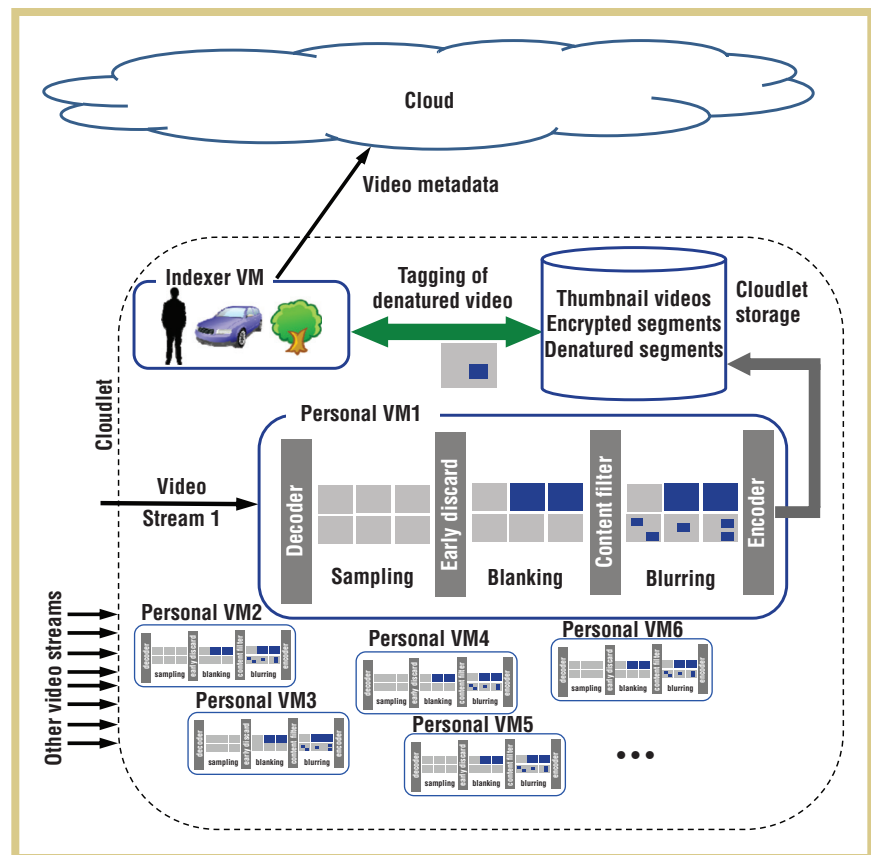

Figure 2. Cloudlet processing in the GigaSight prototype. Denaturing is implemented as a multistep pipeline.

that segment. This newly denatured video segment can then be cached for future reuse.

After sampling video frames, metadata-based filters with low computational complexity are applied. This early-discard step is a binary process: based on the time, location, or other metadata, the frame is either completely blanked or passed through unmodified. Then, we apply content-based filters that are part of the preference specifications for denaturing. For example, face detection and blurring using specified code within the personal VM might be performed on each frame. Figure 3 illustrates the output of such a denatured frame.

Indexing and Search

The indexing of denatured video content is a background activity performed by a separate VM on a cloudlet. To handle searches that are time-sensitive (such as locating a lost child) or to search for content that is not indexed, custom search code encapsulated in a VM can directly examine denatured videos. For each tag produced by the indexer, an entry is created in a dedicated tag table of the cloudlet database. Each entry contains the tag, the ID of the video segment, and a confidence score. For example, an entry "dog, zoo.mp4, 328, 95" indicates that our indexer detected with 95 percent confidence a dog in frame 328 of the video zoo.mp4. After extraction, these tags are also propagated to the catalog of video segments in the cloud.

The throughput of indexing depends on the number of objects that must be detected. Because this number is potentially very high, we propose first
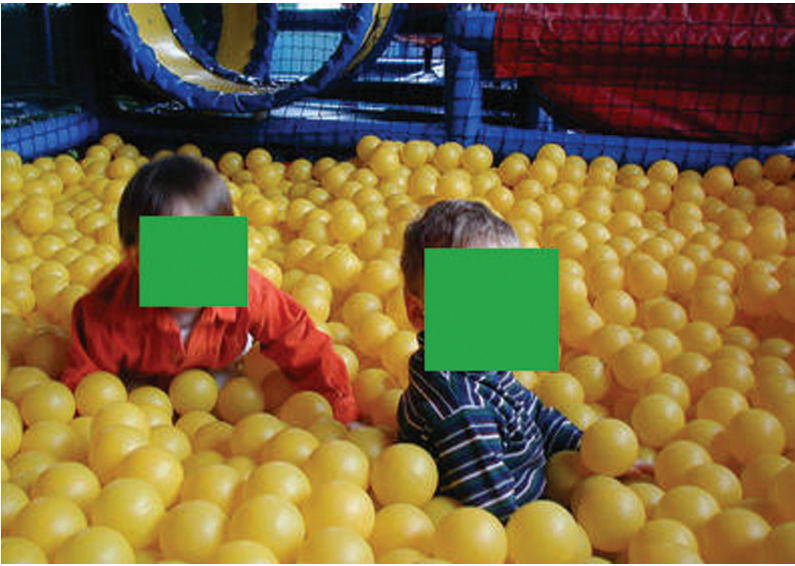
Figure 3. An example of a denatured video frame, based on the following query: "Images taken yesterday between 2pm and 4pm during a school outing to the Carnegie Science Center in Pittsburgh, showing two children in a room full of yellow balls and one of the children wearing his favorite blue striped shirt."

applying only classifiers for the most popular objects sought in the database. Classifiers of less popular objects could be applied on an ad hoc basis if needed. As a proof of concept, the GigaSight prototype uses a Python-based implementation of Jamie Shotton and his colleagues' image categorization and segmentation algorithm,[12] with classifiers trained on the MSRC21 dataset mentioned in their work. This enables the detection and tagging of 21 classes of common objects such as airplanes, bicycles, birds, and boats.

GigaSight uses a two-step hierarchical workflow to help a user find video segments relevant to a specific context. First, the user performs a conventional SQL search on the cloud-wide catalog. The query might involve metadata such as time and location, as well as tags extracted by indexing. The result of this step is a list of video segments and their denatured thumbnails. The identity (the host names or IP addresses) of the cloudlets on which those video segments are located can also be obtained from the catalog.

Viewing all of the video segments identified in the first step might overwhelm the user. We therefore perform a second search step that filters based on actual content to reduce the returned results to a more relevant set. This step is computationally intensive but can run in parallel on the cloudlets. This step uses early discard, as described by Larry Huston and his colleagues,[13] to increase the selectivity of a result stream. Using a plug-in interface, image-processing code fragments, called *filters*, can be inserted into the result stream. These code fragments let user-defined classifiers examine video segments and discard irrelevant parts of them, thus reducing the volume of data presented to the user. We provide a suite of filters for common search attributes, such as color patches and texture patches. For more complex image content, the user can train his or her own filters offline and insert them into the result stream.

To illustrate this two-step workflow, consider a search for "any images taken yesterday between 2pm and 4pm during a school outing to the Carnegie Science Center in Pittsburgh, showing two children in a room full of yellow balls and one of the children wearing his favorite blue striped shirt." The first step of the search would use the time and location information and the "face" tag to narrow the search. The result is a potentially large set of thumbnails from denatured videos that cover the specified location. From a multihour period of video capture by all visitors, this might only narrow the search to a few hundred or few thousand thumbnails. Using a color filter tuned to yellow, followed by a composite color/texture filter tuned to blue and striped most of these thumbnails can be discarded. Only the few thumbnails that pass this entire bank of filters are presented to the user. From this small set of thumbnails, it is easy for the user to pick the result shown in Figure 3.

GigaSight only processes video that is voluntarily shared. Datasets gathered via crowd-sourcing often exhibit a sampling bias toward popular events or news, and the "long tail" is much less covered. We believe this content bias will be lower in GigaSight, because many of its data sources (police on-body cameras, automobile dashboard cameras, and so on) involve continuous capture of video. Conversely, pruning the collection of videos at locations and times, where much redundant footage is available, would limit the richness of data collected by GigaSight. An "uninteresting" detail that is eliminated in the pruning could be exactly the crucial evidence for an important future investigation, such as one of the scenarios in the sidebar.

Automotive Environments

The GigaSight architecture is especially relevant to automobiles. For the foreseeable future, cloud connectivity from a moving automobile will be 3G or 4G. An important question is whether cloudlets should be placed in automobiles or at cell towers. We see value in both alternatives, as shown in Figure 4. This

architecture can be viewed as a mapping of Figure 1 to the automotive context.

Continuous capture and real-time analytics of car-mounted video cameras can help to improve road safety. For example, if the computer vision analytics on your automobile's cloudlet recognizes a pothole, dead animal, or fallen tree branch, it can transmit the coordinates of the hazard (including a brief video segment) to its cell tower cloudlet. The cloudlet can share this information promptly with other cars associated with that cell tower. With advance warning of the hazard, those cars can proactively shift lanes to avoid the hazard. Such transient local knowledge can also be provided when an automobile first associates with a cell tower. There is a subtle trust issue implicit in this capability: a hazard report is assumed to be truthful. Requiring a video segment to support the report can help increase confidence, but it is difficult to verify when and where that video segment was captured. This trust issue points to the need for certified sensor output, where the provenance and value of the sensed data is not in doubt.[14,15]

An automobile cloudlet could also perform real-time analytics of high-data-rate sensor streams from the engine and other sources, alerting the driver of imminent failure or to the need for preventive maintenance. In addition, such information can also be transmitted to the cloud for integration into a database maintained by the vehicle manufacturer. Fine-grain analysis of such anomaly data might reveal model-specific defects that can be corrected in a timely manner.

Cloudlet Size and Placement

The scalability of GigaSight depends on the specific configurations of cloudlets and their locations in the Internet. We have analyzed the tradeoff between cloudlet computational capacity and the number of cloudlets, with the goal of maximizing both the number of simultaneous users per cloudlet (N) and the number of denatured and indexed
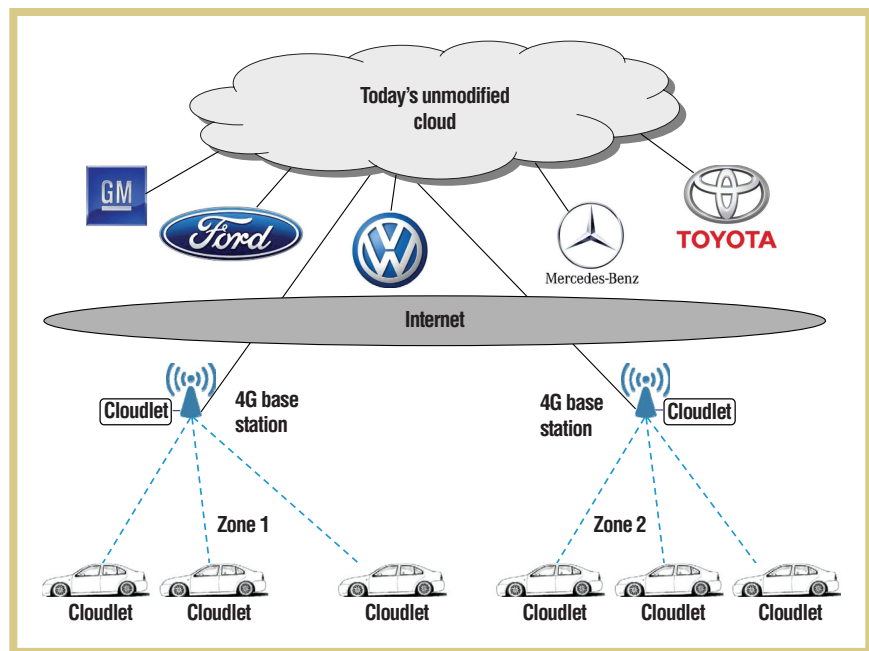

Figure 4. GigaSight for cars. Continuous capture and real-time analytics of car-mounted video cameras can help to improve road safety when shared via cell tower cloudlets.

frames that each user contributes per unit of time (F).[9] This conceptual tradeoff is illustrated in Figure 5. For values of $F < F_E$, the number of users supported is limited to N_E users. For values $F > F_E$, the architecture is compute bound and $N < N_E$.

Using measurements from the GigaSight prototype and extrapolating hardware improvements over a five-year timeframe, the analysis compares the two alternative design strategies, illustrated in Figure 6. In Figure 6a, many small cloudlets are deployed close to the edge of the Internet. In Figure 6b, a single large cloudlet covers a city-sized area. The analysis concludes that edge cloudlets (Figure 6a) scale better than MAN cloudlets (Figure 6b) from the viewpoint of performance. In five years, the analysis estimates that one edge cloudlet would be able to support approximately 120 users with real-time denaturing and indexing at the rate of 30 frames per second. However, because management costs decrease with centralization, a more holistic analysis might suggest an optimum size that is
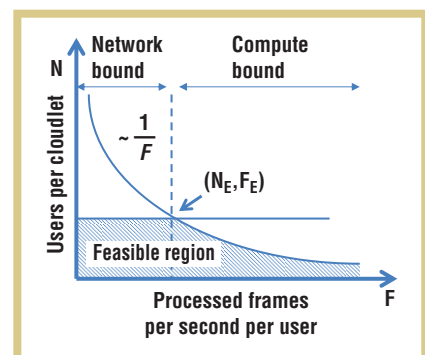

Figure 5. Cloudlet sizing tradeoff. This figure illustrates the tradeoff between the number of users (N) and the processed framerate per user (F). The shaded region represents the range of feasible choices with fixed network and processing capacity.

somewhat larger than that suggested by performance considerations alone.

The central theme of GigaSight is processing edge-sourced data close to the point of capture in space
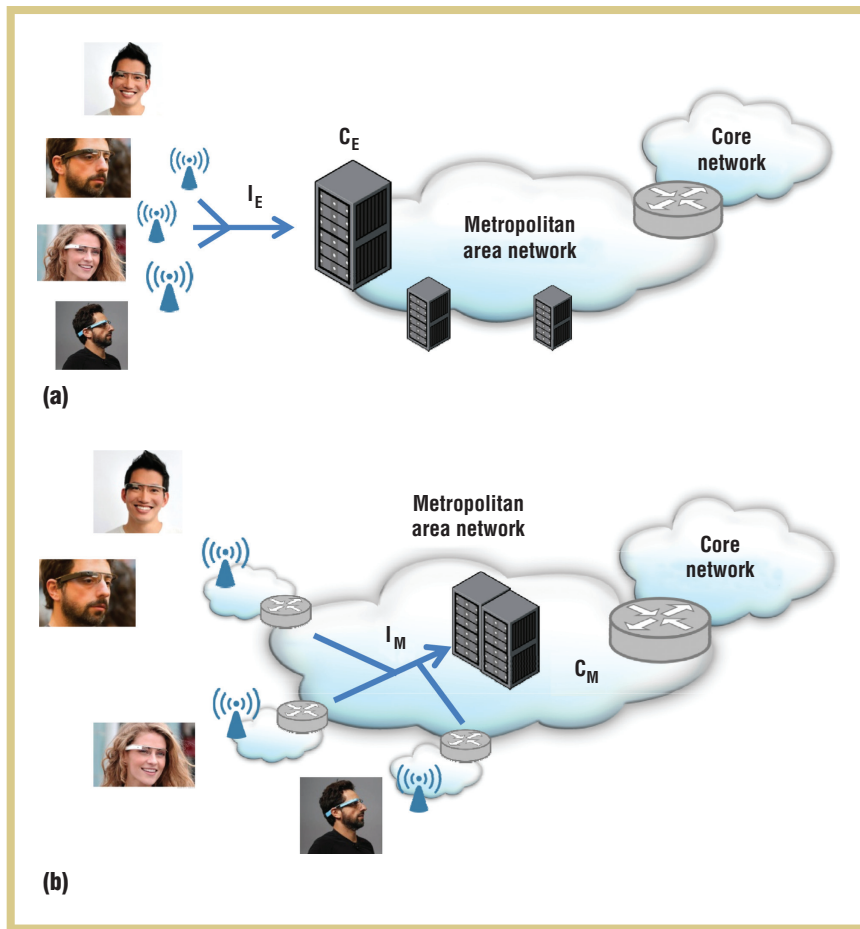
Figure 6. Cloudlet placement: (a) smaller edge cloudlets (C_E) versus (b) larger metropolitan area network (MAN) cloudlets (C_M). Note that I_E and I_M are the ingress bandwidth from the edge and MAN, respectively.

These are just a few of examples of IoT data sources that have high data rates. Many of the points made in this article apply directly to this broad class of sensors. ▣

REFERENCES

1. D. Barrett, "One Surveillance Camera for Every 11 People in Britain, Says CCTV Survey," *The Telegraph*, 10 July 2013; www.telegraph.co.uk/technology/10172298/One-surveillance-camera-for-every-11-people-in-Britain-says-CCTV-survey.html.

2. H. Haddon, "N.J. Mandates Cameras for Its New Police Cars," *Wall Street J.*, 11 Sept. 2014; www.wsj.com/articles/n-j-mandates-cameras-for-its-new-police-cars-1410488158.

3. L. Westphal, "The In-Car Camera: Value and Impact," *The Police Chief*, vol. 71, no. 8, 2004; www.policechiefmagazine.org/magazine/index.cfm?fuseaction=display&article_id=358.

4. A. Davies, "Here's Why So Many Crazy Russian Car Crashes Are Caught on Camera," *Business Insider*, Dec. 2012; www.businessinsider.com/why-russian-drivers-have-dash-cams-2012-12.

5. *Implementing a Body-Worn Camera Program: Recommendations and Lessons Learned*, Office of Community Oriented Policing Services, 2014; www.justice.gov/iso/opa/resources/472014912134715246869.pdf.

6. S. Banerjee and D.O. Wu, "Final Report from the NSF Workshop on Future Directions in Wireless Networking," National Science Foundation, Nov. 2013.

and time. As the density of high-data-rate sensors in the IoT grows, it will become increasingly difficult to sustain the practice of shipping all captured data to the cloud for processing. The decentralized and federated architecture of GigaSight, using VMs on cloudlets for flexibility and isolation, offers a scalable approach to data collection. Sampling and denaturing data immediately after capture enables owner-specific lowering of fidelity to preserve privacy. Performing edge analytics (such as indexing) in near real time on freshly denatured data greatly improves the time-to-value metric of this data. The raw data at full fidelity is still available (during the finite storage retention period) for on-demand denaturing and big data processing. Finally, GigaSight supports interactive, content-based time-sensitive searches that the indexer didn't anticipate.

Also, although we've focused on video cameras as sensors here, any data type in the IoT that has a high data rate can potentially benefit from such a global repository. For example, each GE jet aircraft engine generates 1 Tbyte of sensor data every 24 hours.[16] The airframe of the Boeing 787 generates half a Tbyte of sensor data on every flight.[17] Modern automobiles—especially the emerging self-driving family of cars—generate comparable amounts of sensor data.

Mahadev Satyanarayanan is the Carnegie Group Professor of Computer Science at Carnegie Mellon University. He is an experimental computer scientist whose many pioneering contributions to distributed systems and mobile computing are in widespread use today. Satyanarayanan received his PhD in computer science from Carnegie Mellon University. He is a Fellow of the ACM and IEEE. Contact him at satya@cs.cmu.edu.

Pieter Simoens is an assistant professor at Ghent University, Belgium. His research interests include distributed software systems, mobile cloud computing, and edge computing. Simoens received his PhD in computer science from Ghent University. Contact him at pieter.simoens@intec.ugent.be.

Yu Xiao is currently a postdoc researcher at the department of computer science and engineering, Aalto University. Her current research interests include energy-efficient wireless networking, crowd-sensing, and mobile cloud computing. Xiao received her PhD in computer science from Aalto University. Contact her at yuxiaoaalto@gmail.com.

Padmanabhan Pillai is a senior research scientist at Intel Labs. His current research investigates the use of cloud infrastructure close to the end user to support rich, immersive, latency-sensitive applications on mobile devices. He is more broadly interested in mobile and low-power systems and interactive applications of computer vision. Pillai received his PhD in computer science from University of Michigan. He is a member of IEEE and the ACM. Contact him at padmanabhan.s.pillai@intel.com.

Zhuo Chen is a PhD candidate in the Computer Science Department at Carnegie Mellon University, working with Mahadev Satyanarayanan. His main research interest lies in distributed systems, mobile computing, and the application of computer vision in such contexts. Specifically, he explores how the effortless video capture of smart glasses, such as Google Glass, can benefit people with the cloudlet infrastructure. Chen received his BE electronic engineering from Tsinghua University. Contact him at zhuoc@cs.cmu.edu.

Kiryong Ha is a PhD student at Carnegie Mellon University. His research interests include mobile computing, virtualization, and cloud computing. Ha received his MS from KAIST. He is currently working on a cloudlet project about the convergence of mobile and cloud computing. Contact him at krha@cmu.edu.

Wenlu Hu is a PhD student in the Computer Science Department at Carnegie Mellon University. She is advised by Mahadev Satyanarayanan, and her research interests are in distributed and mobile systems. Hu received her BS in electronic engineering from Tsinghua University. Contact her at wenlu@cmu.edu.

Brandon Amos is a computer science PhD student at Carnegie Mellon University. His research interests include distributed and mobile systems. Amos received his BS in computer science from Virginia Tech. Contact him at bamos@cmu.edu.

7. M. Satyanarayanan, "Mobile Computing: The Next Decade," *SIGMOBILE Mobile Computing and Comm. Rev.*, vol. 15, no. 2, 2011, pp. 2–10; http://dl.acm.org/citation.cfm?id=2016600.

8. M. Satyanarayanan et al., "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, vol. 8, no. 4, 2009, pp. 14–23; http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5280678.

9. P. Simoens et al., "Scalable Crowd-Sourcing of Video from Mobile Devices," *Proc. 11th Int'l Conf. Mobile Systems, Applications, and Services* (MobiSys), 2013, pp. 139–152; http://dl.acm.org/citation.cfm?id=2464440.

10. J. Fan et al., "A Novel Approach for Privacy-Preserving Video Sharing," *Proc. 14th ACM Int'l Conf. Information and Knowledge Management*, 2005, pp. 609–616; http://dl.acm.org/citation.cfm?doid=1099554.1099711.

11. P. Samarati and L. Sweeney, *Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression*, tech. report SRI-CSL-98-04, SRI Inst., 1998.

12. J. Shotton, M. Johnson, and R. Cipolla, "Semantic Texton Forests for Image Categorization and Segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

13. L. Huston et al., "Diamond: A Storage Architecture for Early Discard in Interactive Search," *Proc. 3rd USENIX Conf. File and Storage Technologies*, 2004; www.usenix.org/legacy/publications/library/proceedings/fast04/tech/full_papers/huston/huston_html/diamond-html.html.

14. S. Saroiu and A. Wolman, "I Am a Sensor, and I Approve This Message," *Proc. 11th Workshop on Mobile Computing Systems & Applications*, 2010, pp. 37–42; http://doi.acm.org/10.1145/1734583.1734593.

15. P. Gilbert et al., "Toward Trustworthy Mobile Sensing," *Proc. 11th Workshop on Mobile Computing Systems & Applications*, 2010, pp. 31–36; http://doi.acm.org/10.1145/1734583.1734592.

16. J. Gertner, "Behind GE's Vision for the Industrial Internet of Things," *Fast Company*, July/Aug. 2014; www.fastcompany.com/3031272/can-jeff-immelt-really-make-the-world-1-better.

17. M. Finnegan, "Boeing 787s to Create Half a Terabyte of Data per Flight Says Virgin Atlantic," *ComputerWorld*, Mar. 2013; www.computerworlduk.com/news/infrastructure/3433595/boeing-787s-to-create-half-a-terabyte-of-data-per-flight-says-virgin-atlantic.

cn Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.