Session II Discussion Notes

Scribe: Abhishek Sharma, University of South California

Paper title: Long Range Mutual Information (LRMI)

Presenter: Nahur Fonseca
Discussant: Walter Willinger

Q. What is "new" about LRMI? (Walter Willinger)

Walter's comments:
Long Range Dependence (LRD) in Internet flows has been known for a long time.
Is LRMI just another manifestation of this LRD? LRMI looks at information inside a packet instead of just the "aggregate" statistics like "flow sizes" but what more information does it convey beyond LRD.
Since LRMI is (or can be) defined on a per-flow basis, it might convey more information about the merging and splitting of flows inside the network.

A related question is "How do we use LRMI to do anomaly detection?"

Nahur's response/comments:

The root cause for both LRD and LRMI is the heavy-tailed distribution of flow sizes. However, neither LRD implies LRMI nor does LRMI imply LRD. Even with LRD, the presence or absence of LRMI depends on how we define symbols (I forget the example that Nahur gave but his presentation slide # 17 contains the example). Similarly, an access control scheme like leaky-bucket might destroy LRD in flows but LRMI may still be present.

Q. How can we detect anomalies using LRMI?

We can define our "normal" based on the rate-of-decay of LRMI observed in our training data. One possible characterization of "normal" is the following: how far is the observed rate from the rate expected using Sanov's theorem. One would expect anomalies to show significant deviations from "normal" and hence, can detect them by comparing this deviation against a "user-defined" threshold.

Q. What kind of anomalies can we detect?

The authors have some preliminary evaluation showing that LRMI is good at detecting things like port scan and web crawlers.

Q. Can we detect anomalies even if we have very few samples/ observations for that anomaly? (Jia Wang's question)

Depending on the threshold, LRMI can detect anomalies with small flow sizes.

Q. Is the extent of LRMI completely determined by the flow size distributions? (Jun Xu's question)

LRMI results from heavy-tailed distribution of flow sizes but the extent to LRMI also depends on the defined symbols.

Zhen Liu's comment: It is not clear what LRMI can be used for and how?

Paper title: Parameterize Models with bursty workloads

Presenter: Ningfang Mi
Discussant: Adam Wierman

Q. Can this Markov Arrival Process (MAP) based service time model be applied to other problems like engineering for long range dependence (LRD).

It's possible!

Q. Index of variation captures the second moment. How do we capture information about higher moments (or other information) needed for long range dependence?

MAPs with more than 2 states might be needed. But solving (determining the parameters) MAP models with more than 2 states is very complex.

Q. How we model an open or partial-open queueing model with burst workloads? (The paper only looks at a closed queueing model which is more appropriate for TPC-W like workloads. A partially-open queueing model captures real system workloads better).

An open queueing model with finite buffers is equivalent to a closed queueing model. The authors did not consider partially-open queueing models but one might require more fine-grained measurements than the measurements used in the paper to parameterize a partially-open queueing model.

Q. Why is the index of dispersion needed? Will knowing the squared coefficient of variation (SCV) and the autocorrelations suffice?

Yes. But it is difficult to determine the autocorrelations exactly using coarse-grained samples.

Q. Does the index of dispersion depend on the sampling interval?

Yes. But the method for determining index of dispersion is more robust to errors to aggregate samples collected using large sampling intervals than the technique to determine the autocorrelations.

Zhen Liu's comments: We should aim to strike a balance between accuracy and intuition. Mean Value Analysis (MVA) might be less accurate but provides more intuition than MAPs.

Paper title: DRAM is Plenty Fast for Wirespeed Statistics Counting

Presenter: Bill Lin
Discussant: Jia Wang

Q. Is the technique proposed in this paper similar to having a "small cache and interleaving of requests"? (Arif Merchant)

There are significant differences.

1. The queue sizes used in the paper are much smaller than the x86 cache.
2. Requests are queueing according to the statistics of the arrival processes (as opposed to some "optimization algorithm" used to interleave requests.

However, the idea of interleaving memory requests is borrowed from the research in disk arrays.

The paper motivates the usefulness of "not" treating the DRAM as a black box (especially for the new DRAMs that have been developed to satisfy the demand from gaming industry). But thinking of the proposed scheme in terms of the abstraction of a "cache+interleaving" might make thinking of generalizations of this technique easier.

Q. Can we count symbols (as defined by LRMI) efficiently using the ideas proposed in the paper?

Even though the level of flow aggregation at core routers is high, this aggregate will be distributed over a large number of memory banks and so the scheme/architecture proposed in the paper can count LRMI symbols efficiently.

In the paper, the authors considered the worst-case traffic possible (from the point of view of statistics counting) in order to eliminate any dependence of the results on the pattern of arrivals.

Q. How does the complexity of the approach proposed in the paper compare against schemes like "Counter-Braids" (presented at SIGMETRICS 2008)?

The two techniques proposed in this paper are much simpler.

Q. Will the proposed scheme work for determining sketches?

The scheme is general enough to be usable for computing sketches efficiently.

The read/write output of the proposed scheme is fast but the delay will still be the same as the delay for DRAMs. However, most sketch requests are independent and so the memory access delay will not hurt.

However, if the proposed scheme were to be used for CPU intensive applications with high temporal correlation then the memory access delay might hurt.