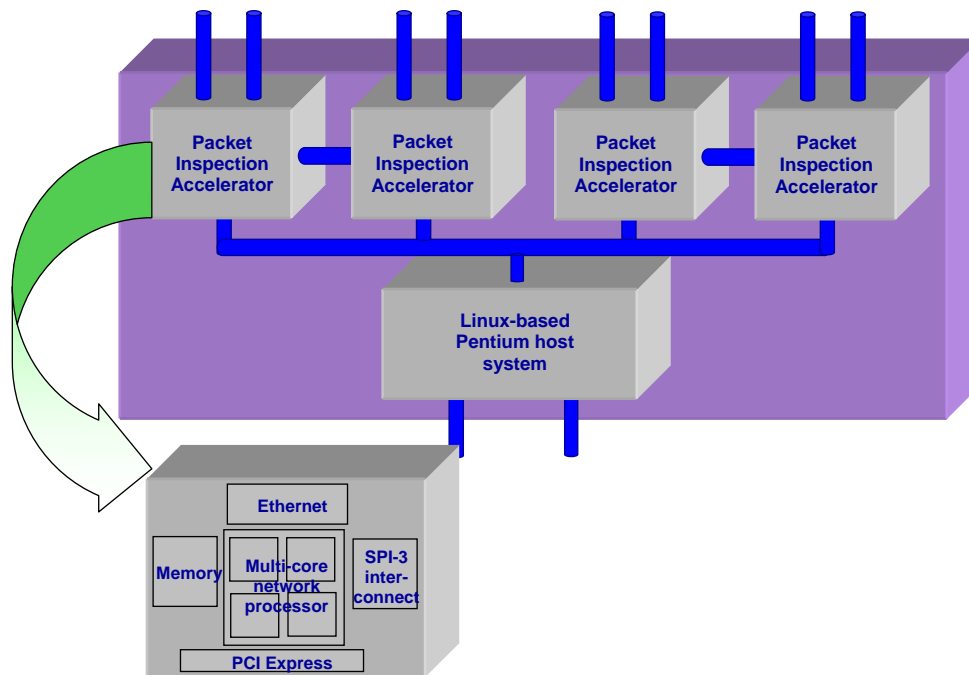


# IP Fabrics' DeepSweep™ Redefines the Playing Field in High-Performance Network Surveillance

September 18, 2006

The DeepSweep surveillance system has a wide breadth of functionality, embodied in its easy-to-use, configurable software functions called Surveillance Modules, to provide a broad range of uses in VoIP and broadband Internet intercept, content or signature-based surveillance, analysis of malicious traffic, forensic analysis of cybercrime, detection of network abuse, and others. In spite of this functional power, the area where DeepSweep really outshines other solutions is its high performance. We will explore what makes this so in this paper.

A DeepSweep system has the internal architecture in the figure below.



One can immediately see that DeepSweep deploys many concurrent processing engines, using an Intel Architecture host system on the back end and multiple, multi-core, network processors on the front end. In the DeepSweep-1 appliance, two or four Packet Inspection Accelerators, or PIXL's are used, and each contains a four-core (four independent processing engines) network processor. In the DeepSweep-10 ATCA system, each PIXL is a 16-core network processor, and there may be 2 – 8 PIXLs in the system. Adding yet more

concurrency, each core of each processor is hardware multithreaded, so each core can actually process 2-3 packets concurrently.

	Concurrent packets per core	Concurrent packets per PIXL	Concurrent packets per system
DeepSweep-1	3	12	24-48
DeepSweep-10	2	32	64-256

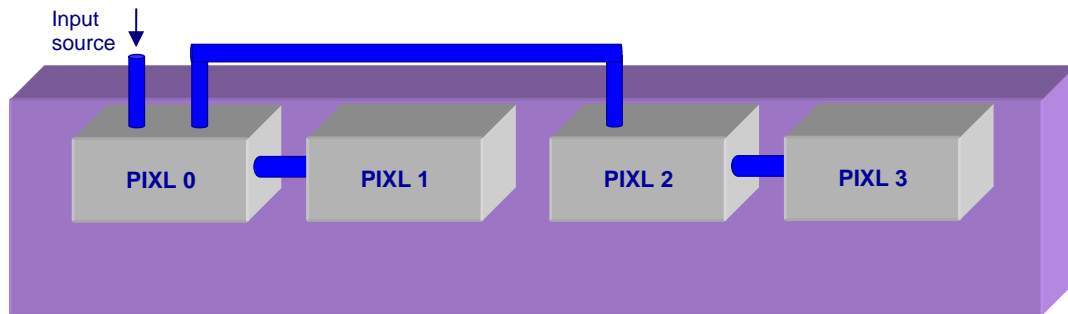
Thus one aspect of DeepSweep that leads to high performance despite the extensive functionality it provides is

1. *DeepSweep contains a substantial amount of hardware parallelism.*

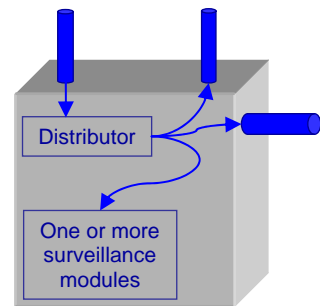
Looking at the diagram above, the benefit of the parallelism is easy to see if there are actually N streams of packet input, but what if we have a more-typical situation where we wish to do surveillance on one high-speed network link? No problem; this is where DeepSweep’s Rearrangable Accelerator architecture comes in to play. We will use a 4-PIXL DeepSweep to illustrate.

To best use the rearrangeable architecture, we need to understand the specific surveillance problem we are trying to solve, but I will show the range of possibilities here.

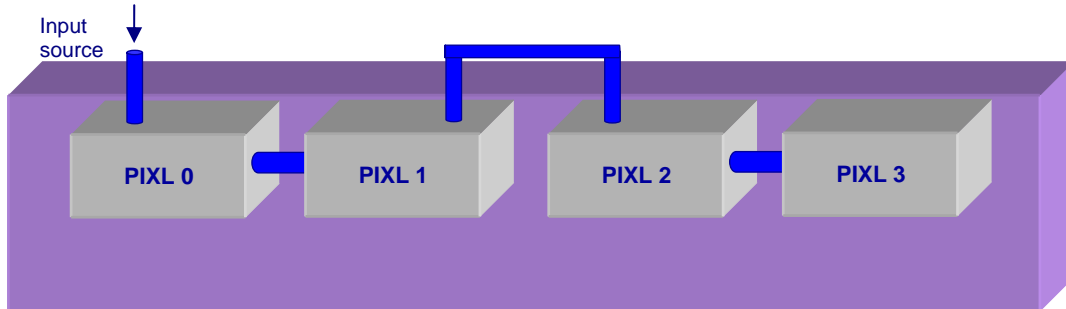
Suppose we determine that the best solution for our specific problem is to deploy all of this concurrency in parallel at the same level – i.e., each PIXL doing the same thing. Conceptually, what we want to do is split our input stream into four, and within each PIXL, of course, the stream will be automatically split in up to 12 or 32 additional streams. We do this by “wiring” the PIXLs as shown.



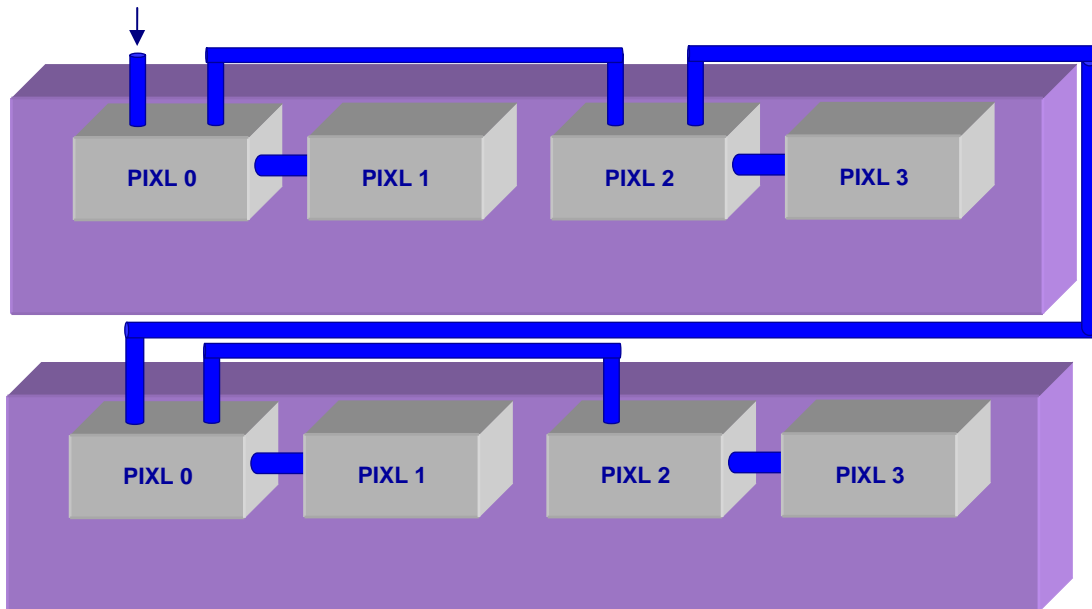
To understand how this accomplishes our goal, we have to look at a bit of detail about the software architecture of a PIXL. A PIXL consists of a distributor and one or more surveillance modules. A distributor can route the input source in up to three directions, and can do so in a sequential manner or flow-based manner. So via the browser interface we tell the distributor in PIXL 0 to divide the traffic such that some stays in PIXL 0, some goes to PIXL 1, and some goes to PIXL 2. In PIXL 2’s distributor we would specify to split the packets between PIXL 2 and PIXL 3.



At the other extreme, we might want to use a purely pipelined approach, where some surveillance modules reside in PIXL 0, their hits and/or misses flow to PIXL 1 and some other surveillance modules, and then we flow to PIXL 2, and so on. This corresponds to the diagram below, and the distributors are not used (each distributor just routes its input to its internal chain of surveillance modules).



To look at one other configuration, let's say that the processing we need DeepSweep to do is so extensive that four PIXLs are insufficient (and we aren't using the ATCA DeepSweep-10 where more PIXLs can be added). In this case, we can couple DeepSystems together. In the example below, we have decided we need pairs of PIXLs pipelined, with a total of four PIXL pairs.



So another aspect of DeepSweep that leads to high performance despite the extensive functionality it provides is

*2. DeepSweep's Rearrangable Accelerator architecture allows the processing power to be optimized to the problem at hand.*

And since it is modular,

*3. DeepSweep's Rearrangable Accelerator architecture allows processing power to be added incrementally.*

But we are not done. DeepSweep-1 is a 2U appliance with two or four PIXLs, and the network interfaces are all 1 GE. If our problem requires vastly more horsepower and perhaps 10GbE interfaces, we can move to the DeepSweep-10. First, it is important to note that the user interface and usage model of DeepSweep-1 and DeepSweep-10 are identical, so someone who has used one can readily use the other, and surveillance module definitions and databases are transferable from one to the other. In terms of performance, DeepSweep-10 gives us the following extensions:

- Network interfaces can be 1GbE or 10GbE
- Each PIXL is considerably more powerful
- Because of the blade structure of ATCA, we can add a larger number of PIXLs

*4. The DeepSweep family, all completely compatible, extends to multiple 10 gigabit inputs.*

There is one other performance advantage that might appeal to the sophisticated user. Unlike all other systems that we know of in this category, DeepSweep can be used as an open system. That is, the sophisticated user can place an existing or newly developed Linux application in the system and have the surveillance module(s) send "hits" to the application. If the user needs to extend the capabilities of DeepSweep in some way, he can do so directly within the DeepSweep system, rather than having to ship the output to some external system.

*5. The DeepSweep system can be an open system, allowing a user to add specific backend processing.*

## Numbers

The proof of the architecture is, of course, in actual results. In the table below, we give typical processing rates for DeepSweep-1 and projected rates for DeepSweep-10. We have used several single-surveillance-module examples that we feel are representative of real problems.

The results are stated in millions of packets examined per second; one million packets per second is roughly equivalent to a fully loaded 1 GE network segment. The examples assume a DeepSweep-1 with four PIXLs and a DeepSweep-10 with eight PIXLs.

	DeepSweep-1 Million packets per sec	DeepSweep-10 Million packets per sec <sup>1</sup>
CALEA type intercept (watching for a small number of IP addresses and VoIP phone numbers / user identifiers)	4.0	40.0
Email intercept (watching email headers for a small number of specific email addresses)	3.7	30.0
Watching for HTTP traffic to a particular web site, and within such for a particular cookie	4.5	40.0
Packet flow surveillance machine, searching the payload of each packet against a large signature database and then grabbing the rest of the flow (e.g., TCP flow)	4.0	35.0

The percentage of hits and what actions are specified for hits of course have a big effect on performance. For the DeepSweep-1 numbers above, we assume a 1-per-1000 hit rate. For DeepSweep-10, we assume a 1-per-10,000 hit rate (because we are searching a far bigger haystack).

---

<sup>1</sup> Projected.