

# White Paper - The Next-Generation Media Server

Digital-media telephony systems implement applications such as media gateways, media servers, pre-paid calling, and unified-messaging systems by directing media-processing and stream-switching resources to perform various functions on call streams. In addition to the application, these systems are comprised of computing platforms, system software, media-switching and call-control resources, and media-stream processing resources. Media streams are made available to the system through network-termination devices. Calls are set up using call-control protocols associated with the network termination. These systems are quite complex and expensive when compared with off-the-shelf open-architecture computers. But within the past 10 months, technologies have been brought to market that, if effectively synthesized, have the potential of creating a beneficial discontinuity in the price, size, scalability, manageability, and power consumption of such systems.

## Current-Generation Media Servers

For circuit-switched (based on time-division multiplexed technology—TDM) connectivity, specialized network-interface hardware is required. Typically, the computing resources that process the call-stream data are co-located with the network interface or are on adjacent resource boards interconnected through PCM-highway connections. For several reasons, DSPs, rather than the host processor, are typically used to process the stream data.

Just a few years ago the processing requirements of many telephony stream-processing algorithms, when multiplied by the number of channels in a system, demanded processing resources that either exceeded, or were a significant fraction of, the MIPS available on the system's primary computer. This meant additional compute resources were required to meet the algorithmic processing requirements of the system. Even if the host computer could handle the load, digital-media system-level software (telephony middleware) has not been available to take advantage of host MIPS as a media-processing resource. Moreover, partitioning the call-stream computing resource onto add-in boards, which could be increased in number as system expansion required, reduced the possibility of post-installation resource-limitation problems. Also, those interface boards require a chassis, fans, and power supplies, a major expense, making co-locating the DSPs on the network interfaces a relatively small incremental cost.

If host-based MIPS were used and more MIPS than were available on one computer were required, system expansion became extremely difficult and expensive. Commercially available telephony middleware (the application-level APIs and associated software) was not well suited to handle system scaling in the host-computer dimension. Extensible communications protocols were not used between the application and resource, forcing the application and media resource to be co-located. This required that the application run on each computer, again complicating system scaling. And compared with a DSP board, a PC, for example, was a large piece of equipment that required significant power.

If a PCI bus and associated I/O drivers were used to interface the add-in boards, something of a timing abyss existed between host-based software and software entities on the PCI boards. Delays of 10-40 milliseconds are common, depending on the operating system being used on the host computer. This complicated echo canceller placement if the host was to be used for vocoders or speech recognition. Moreover, some media technologies, such as fax, required tighter timing between controller and DSP resource than could be effected across the PCI bus, forcing some manufacturers of fax resource boards to dedicate microprocessors to each fax channel on the add-in boards.

Blade-based PC servers, IP-based telephony transport, and next-generation telephony middleware and media-stream frameworks are now challenging these long-standing assumptions. A new generation of media servers is now possible. These next-generation digital-media systems offer the user an unprecedented level of system flexibility, scalability, and affordability. Other than a highly scalable industry-standard computing platform based on the so-called blade server, they are all software. No specialized hardware is required.



IBM's eServer

## The Blade-Based Media Server

The recent advent of blade-based PC servers has made the scaling of host-based MIPS practicable in terms of space, power, cost, and system management. "Blades" refers to the circuit boards, implementing one or two complete PCs, that are housed in a card-file enclosure in a manner similar to telephony resource boards. Nexcom is shipping and IBM, Dell, and Compaq, among others, have announced blade server products. Depending on the vendor, these systems can include hot-swap blades, dual-redundant power supplies, status monitoring and system-management hardware and software, and other features that make them suitable for high-density high-availability server environments, which can scale to 280 servers in a single rack. With a blade server, adding a 1.33-GHz PC to a system is no more involved than adding a hot-swap DSP-resource board to a compactPCI chassis.

And it costs an order-of-magnitude less.

But what about the network interface? Obviously, there must be one. But with an IP transport, Ethernet is the usual WAN connection. The typical TDM telephony interface: thousands of dollars; the typical Ethernet interface: hundreds of dollars. Another order-of-magnitude difference.

Once the costly TDM interfaces and circuit switches disappear, the use of DSPs to process call streams must be justified without the benefit of common-equipment sharing with those interfaces. Today, PC-based MIPS are \$0.5-2; DSP-based MIPS are \$2-5.

## The Software

But low-cost, scalable MIPS are of no benefit unless system software is there to harness those MIPS to the task of processing call streams and make scaling the system in the client and server dimensions transparent to application software. This requires a new generation of telephony system software.

Every telephony system that requires call-stream processing has a software framework that is responsible for binding a media-processing resource (hardware and software) to a call stream. If the system supports multiple calls and multiple media, the necessary software framework can represent a significant portion of total system complexity. Although complexity can be reduced by making this binding static, this is a poor tradeoff when the resulting reduction in resource utilization and system scalability are considered. This means a software framework is required that dynamically binds a call-stream processing resource to a stream source and a stream sink under the control of the application-level software. Some systems use the so-called stream graph, as shown above, to model the task.

The stream graph is built under the direction of a media-specific software entity that acts at the behest of a client application. The media controller will accept commands, build the stream graph, send events to the client, and tear down the stream graph when the call is complete, thereby freeing the call-specific resources.

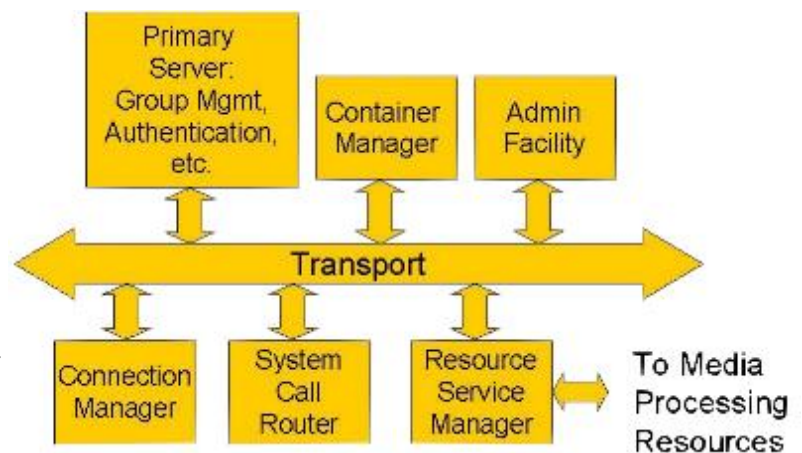
Such an environment is just as suitable to execution on a host computer as on a DSP or distributed across multiple processors.

Partitioning a digital-media system to expose the media-streams software environment is necessary for it to become the subject of specification. Over the last few years, the MSP Consortium ([www.msp.org](http://www.msp.org)) has developed and published such a specification, M.100 Media Stream Processing Environment (MSP). M.100 specifies a streams environment by specifying the APIs that entities within the environment use to perform stream-processing tasks. As long as software within the environment uses the M.100-specified APIs, the server developer can choose technologies provided by multiple vendors. The evidence that quality will go up and prices will go down as a result of such a multi-vendor environment is too abundant to be ignored.

OpenMedia is Commetrex' implementation of the MSP Consortium M.100 streams environment.

But having an open multi-vendor streams environment that executes on the multiple PCs (or other types of server blades) may be necessary but is not sufficient to make system expansion through the addition of server blades a viable system architecture. A client-server telephony-middleware software framework is required. Again, there is an open standard available. The Enterprise Computer Telephony Forum (ECTF) ([www.ectf.org](http://www.ectf.org)) has published S.100, a specification of a client-server telephony framework that supports implementations with the necessary features.

To date, most implementations of S.100 (there are four) have been vendor and resource specific, and they have not been integrated with a streams environment that supports host-based stream processing. Commetrex' OTF® Kernel is an S.100-conforming telephony middleware software product that allows the system developer to implement blade-based servers which offer seamless expansion in both the client and server dimensions. OTF Kernel offers the system developer

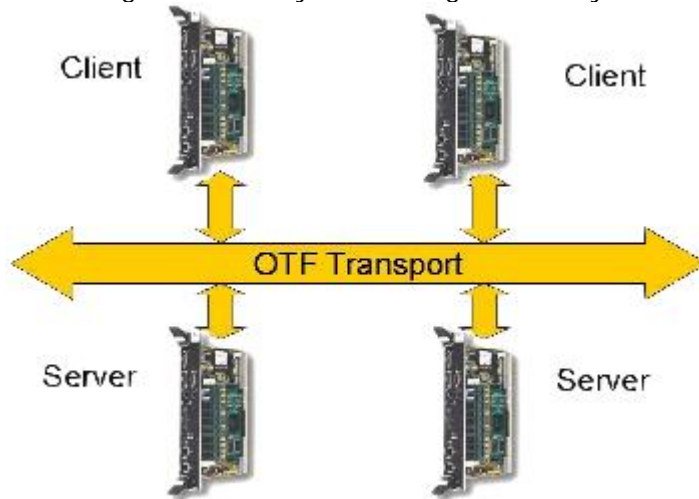


- Control of strategic platform
- Resource independence
- Seamless scalability
- Configurability/Extensibility
- Application portability

Other than OTF Kernel, all telephony middleware products are closed-architecture products bundled with the vendor's resource boards. As such, the system developer loses control of the system platform. An open, modular telephony middleware product puts the system developer back in control of the system platform.

Resource independence is achieved by abstracting the media-processing resource behind a resource service manager combined with an open environment at the client API.

Seamless scalability is achieved through a distributed client-server architecture that moves command routing, resource configuration and allocation, container management, connection management, and system management to system-service entities, as shown above. Client



applications are written without regard for the location of resources. System expansion, then, is achieved by adding a processor and configuring it through the system-management facility. Of course, clients and servers are software abstractions, making their assignment to a particular processor a function of system-availability and management issues.

## Conclusion

The availability today of blade servers and next-generation telephony middleware and stream-processing frameworks is ushering in a new generation of digital-media telephony servers. These servers are based on a new level value-adding modularity, which delivers significant benefit to the service provider:

- Scalability/configurability
- Lowest cost
- Vendor independence
- Media flexibility
- Future resilience



Nexcom's HiServer