

On-board IP architecture

A new approach to computer telephony card design

Ian Colville, Product Manager, Aculab



Executive summary

As the computer telephony (CT) industry has evolved to deal with the emergence of VoIP and IP telephony, the notion of separate digital signal processing (DSP) resource cards and IP connectivity or gateway cards has evaporated. Just like the demise of traditional, single function digital network access cards, fax cards, and voice processing cards in favour of combined products, such as Aculab's Prosody.

Multi-purpose, multi-function cards are now the norm and the industry is looking to where these latter-day design concepts can be improved – yet still operate within the industry accepted form factor specifications.

One reason for this is that despite the advent of new specifications, like AdvancedTCA and associated packet switched backplane (PSB) technology, in practice the PC server, and hence the CT industry, remains heavily biased in favour of PCI (or CompactPCI) based designs.

This paper presents an alternative to the traditional PCI driven parallel bus architecture for the core, on-board design of versatile, multi-function CT resource cards. It demonstrates a 'think outside the box' approach that combines adherence to standards with optimum use of prevalent technologies.

The unique design hypothesis will illustrate a method of overcoming limitations of proprietary device drivers to present an attractive option for large scale, distributable CT applications.

The traditional approach

From single to multi-function

The traditional approach to CT card design has evolved from the days of single function ISDN, fax and voice boards – from the likes of Dialogic and Rhetorex in the early days – to the situation where multi-purpose, multi-function cards are now commonplace, through innovations like Prosody from Aculab.

Many vendors in the industry now offer combined IP gateway and media processing resource cards. And the competition amongst manufacturers of universally functional cards in PCI and CompactPCI formats is intensifying, with several new vendors entering the fray.

Most competitive offerings seem to show strengths in one area or another; either the IP domain, or in DSP resources perhaps, but typically not both. And there are some ‘tick box’ suppliers. Customers mostly want card density and configuration options, so it is desirable for vendors to offer a ‘mix and match’ set of modules, which they balance by cost-effective manufacturing variants, employing high re-use of the same core technology.

Conventional blueprints

With the majority of vendors having migrated from ISA to PCI and/or CompactPCI cards, a common factor in base card design is the PCI bus; a parallel data bus. This has proven to be good enough for first and second generation CT applications, but presents some limitations for today’s third generation products with their IP telephony cynosure.

A key design point in relation to current generation PCI and CompactPCI cards is simply that because they are proprietary creations – albeit designed to PCI-SIG/PICMG bus and form factor specifications – they require vendor specific device drivers and libraries. These will have been written to be compatible with one or more operating systems (OS), with different driver software needed for each OS, and importantly, the libraries will be able to talk only to the vendor’s drivers.

This conventional blueprint, hitherto more than adequate, has certain limitations.

Namely, the available hardware resources can only be readily controlled from within the platform they are installed in. That is unless some technology like, say MC3 (an H.100/H.110 bus extension option), is employed to scale a solution beyond a single box and some additional middleware similar to the ECTF’s – now unheralded – CT-Media is brought into use to add control of the algorithms. Or a solution provider/third party is engaged who provides an integrated platform that controls distributed client and server applications and hardware resources. Or perhaps the customer – the hard pressed application developer or systems integrator – has no option but to write their own distribution API.

All of these options are no doubt in widespread use, but they all have one thing in common – they tackle the problem from an API/application perspective, putting the onus on the customer to add something to the mix in order to achieve the end result.

From a low-level, hardware design, device driver standpoint, the challenge to the vendor community is, “why not consider an alternative?”

The use of proprietary drivers and libraries precludes the ability to make use of the distribution mechanisms built into today’s IP stacks, which come ‘for free’ with the OS. Therein lies a clue.

The need for a new approach

Does size matter?

Well, in truth, if the needs of the application are satisfied by a discrete system in a single chassis and any future scaling is not needed beyond the maximum capacity of the platform, it really doesn't matter too much.

However, if larger systems are envisaged, much larger than the initial deployment, for example, then this could be a significant factor. There are other considerations too, such as the need for the optimum utilisation – or sharing – of valuable hardware resources. This is a prime concern of application service providers (ASPs) for instance, for whom capital expenditure and overheads are key determining factors.

If the available resources can only be readily controlled from within the box they are installed in, then sharing of resources cannot be readily achieved. That is not to say it cannot be realised; rather it becomes more difficult.

If the available resources cannot be readily scaled or expanded, then the development of large-scale, distributable CT applications is less easily accomplished.

As processor power increases, components of a size offer more per mm², and prices are squeezed, customers are getting greater value for money than hitherto. And the option of greater channel density in the same form factor, or using even larger card sizes, such as AdvancedTCA, also exists for large-scale telco system deployments.

That said, addressing the software issues and/or adding bigger product variants to their portfolio isn't the panacea, and aren't the only choices available to vendors. With an investment in PCI and CompactPCI designs and an industry wide customer base yet to be convinced of the advantages of third generation PICMG 'form and fabric' specifications, the life expectancy for these more mature technologies can be summed up by one word – longevity.

Captive cards

So what's the problem that we're trying to resolve? In a nutshell, it is the scalability and distribution of system resources. Scalability means expanding the number of resources available to a system or application. Distribution means being able to readily locate those resources independently of the application i.e., non-co-located – ideal for those ASPs or service providers.

Current generation cards installed in a chassis are essentially captive; they belong to the box they are in. They are locked into the application platform they serve; not readily accessible to any other application.

Why are they locked in?

A standard PCI card is controlled by device drivers. These are by their nature proprietary i.e., value added code designed to control only the vendor's own card or card type(s). The link between these drivers and the application that needs to make use of the card's resource functions is the vendor's application programming interface (API) library. The library is then also proprietary and only controls the same card or card type(s).

The library and driver are not designed to control anything else, anywhere else but in the same box. They cannot access devices on another box. There is no method of identifying another device, even a card of the same type from the same vendor, on another box. The library and driver haven't got such a routing mechanism.

There are ways around this of course, but like middleware, the key point is that they are not highly efficient.

To'ing and fro'ing – an example

Reasonably sized IVR deployments often require the use of more than one host machine in order to make use of the scalability, reliability and greater capacity that such systems can offer. An example of this is speech recognition systems, for which the speech recognition engines, and sometimes the applications, are typically located on one or more hosts remote from the machine containing the speech resource cards (e.g., TDM telephony and DSP resource cards with record, echo cancellation, DTMF detection, etc.).

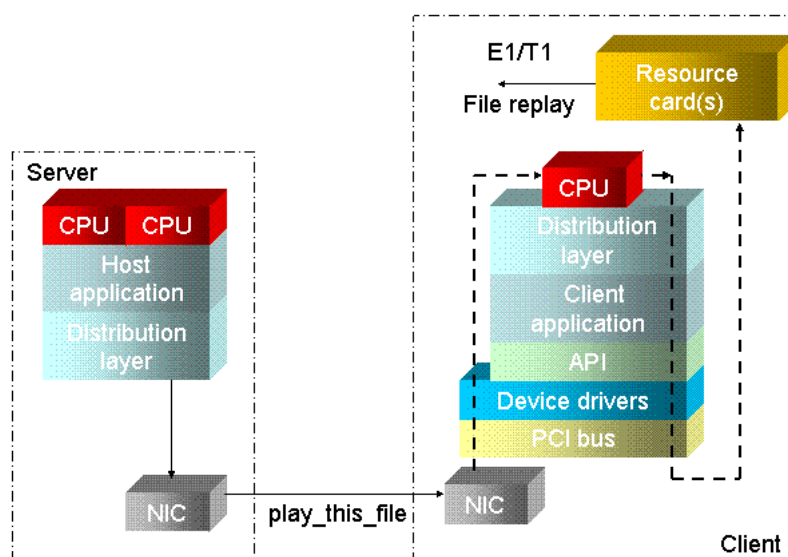
A speech recognition engine running on a server may receive recorded speech from a client machine where the speech resource cards needed to process the incoming speech signal are located. Similarly, an application running on a server may send speech to the same client machine for replay onto the line. See diagram 1.

Using current technology, this client machine will run a 'recognition client' process which interfaces between the speech resource cards and the remotely located speech recognition engines and applications. If a remotely located application wants to play a prompt, it sends a 'play_this_file' message together with the file itself, via the local area network (LAN), to the 'recognition client' process on the client machine. This process opens the file and streams it to the resource card for playback using a 'play_this_file' API call. This is inefficient.

Why? Because the file arrives at the client machine's network interface card (NIC), passes over its PCI bus and has to be processed by the client CPU before being passed via the API and device drivers over the PCI bus (again) to the speech resource card for replay. Not only does this amount to a 'double-hop' over the client machine's PCI bus, but you must also write the 'recognition client' process application as well as the main application. And you have to write your own distribution mechanism too – a distribution layer above the API – for both machines, because the API only talks to the card driver and card over the PCI bus. Even if the file is stored on the client machine, its CPU still has to get involved to send it up to the card over the PCI bus.

A better way would be to send the file direct to the resource card for replay, if only the card could accept it.

Diagram 1



The Aculab concept

Ubiquitous IP

To achieve scalability and distribution of combined media, VoIP and PSTN (TDM) system resources, this design concept uniquely employs an on-board and inter-system, all-IP control architecture. The result of this is that multiple hardware resources (cards) can be shared by multiple, independent (non-co-located) application servers.

IP has been deployed on a global scale, and huge investment has made it ubiquitous; it is a stable, proven, reliable technology. For various reasons, controlling a card over IP is considerably lighter on the available host resource than the host performing the equivalent amount of IP telephony. And as IP telephony is achievable now, so IP control of cards is easily achievable.

How does it work?

As has been stated, proprietary drivers and libraries do not have an integral routing function. On the other hand, using a common media access control (MAC) layer driver and the OS IP stack inherently provides the Ethernet connectivity and desired routing capability. So why not make use of these distribution mechanisms? After all, they are built-in to the IP stacks and so effectively come 'for free' with the OS.

Using a MAC component on a resource card in place of a more usual PCI interface device means that when the card is plugged in to the host it is recognised as a NIC and gains an IP identity of its own.

As with current generation cards, a vendor will continue to provide device drivers and libraries for these 'resource NICs', enabling applications to communicate with the card(s). The driver software will still have been written to be compatible with one or more OS, but the essential driver technology is available and well proven due to the many deployments of the chosen MAC devices. But the key point is that these drivers and libraries have standard, OS-defined interfaces at the IP stack layer.

This means that the IP stack can readily route data between a library and such a 'resource NIC' driver or, indeed, any regular NIC driver. Therefore, it uses the same mechanism whether routing data between the library and a local card or between the library and a remotely located card (via a regular NIC). As each resource card has its own unique IP/MAC address and its own Ethernet connection it is 'findable' on the network by any application in any server. In this way cards can be directly controlled via Ethernet connections on a private LAN.

The efficiency gained through this design feature comes from the removal of the 'double-hop' referred to above, which frees up the local CPU to do a whole lot more work.

Apply this technique to the entire control path anatomy of a card and the net result is an all-IP structure. Resources can be shared across both the on-board and external IP networks, enabling not just the card, but the resource functions of the card, to be controlled remotely.

This architecture makes it ideal for cards to be controlled either by a local application on the host CPU, or from a separate host PC. What is more, the API does not require the addition of a higher layer to achieve distribution, because the IP stack and NIC drivers take care of the routing of messages, whether via the LAN or locally.

And there's more

Also employing a modular, IP-interconnected design on the card means scaling from low to high densities is readily achievable in two ways;

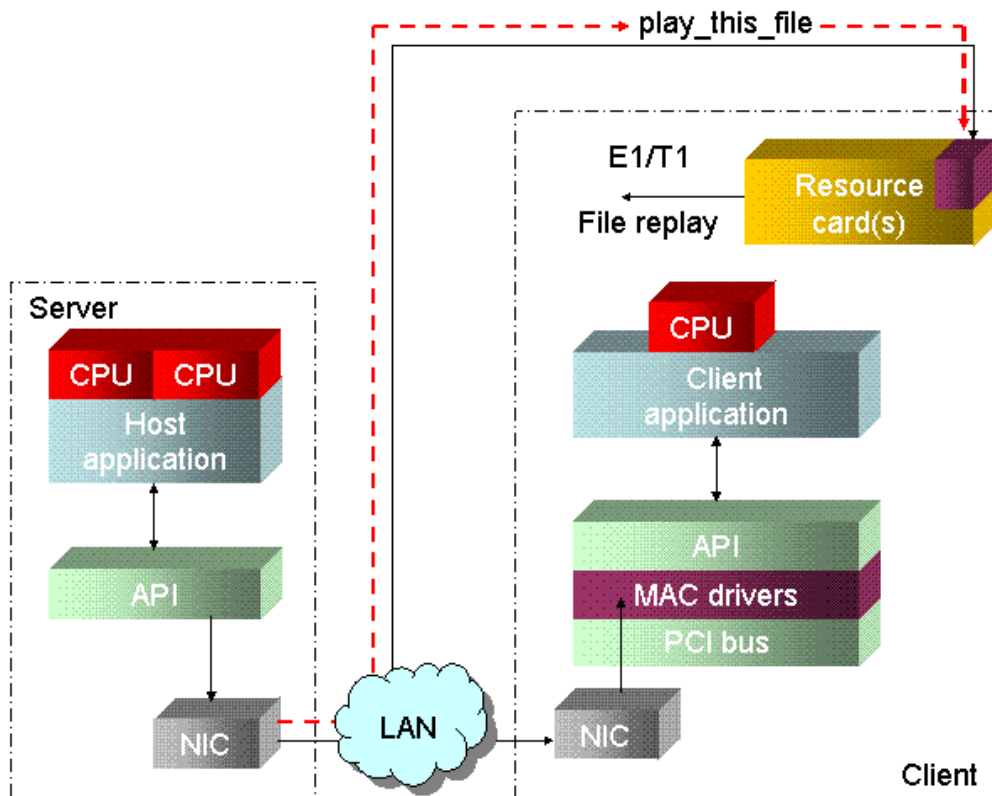
1. By providing on-board module options for DSP resources and PSTN network connectivity
2. And through adding cards in one or more chassis, to give intra- or inter-system options

Moreover, when adding cards in an additional chassis, as has been shown above, the control data path needn't go via the traditional server-to-client route. The communication data path is directly with the card via its own Ethernet connection – see diagram 2. Taken further, this idea means that the use of a 'passive' chassis, simply for system expansion, is feasible, with no client CPU requirement at all.

Monitoring of distributed chassis and cards is also readily achievable via LAN connectivity using industry standard network management procedures like SNMP to control the remote equipment.

The principal of modularity also enables customers to enjoy the flexibility of selecting a given function or not e.g., keeping the TDM trunk connectivity on a daughter module. Plus from a regulatory perspective this separation allows easier achievement and maintenance of type approvals.

Diagram 2



Conclusions

This concept is ideal for products designed to fit into the complex, converged technical space where support for circuit switched digital network access, media processing resources and IP telephony are all required in a multi-platform, distributed system. And because the interconnection is via Ethernet, the channel count of the distributed, shared resource pool is restricted only by the available bandwidth and the capabilities of the host machine's driving it.

The idea of adding modules and then cards to expand a system provides excellent, all round granularities for customers, from small-scale, low-density single card systems up to large-scale systems of virtually unlimited size. Using passive expansion chassis, filled with resource cards, also contributes to the cost-effectiveness of the end solution. Very high channel counts are possible, with greater value per channel leading to improved margins. Music to the ears of the Financial Director!

The principle of shared resources also leads to optimum efficiency, in contrast to a traditional system where under utilised capacity will be inaccessible to other applications.

A crucial factor for customers seeking to migrate is consistency with previous driver APIs. This approach allows customers to re-use existing applications, with a very small number of changes, because the interaction between the API and the driver should be essentially unchanged from an application perspective.

Furthermore, this creative approach can allow open standards platforms and solutions to more readily compete with proprietary enterprise and telco switching systems, enabling bids for projects that were previously out of reach.

About Aculab

Aculab enables developers and systems integrators to produce a variety of high performance communications solutions. Aculab's portfolio offers an exceptional mix of capabilities that are easy to integrate and bring real value – reduced costs, increased customer satisfaction and competitive advantage. A complete range of open standards building block technologies for use within telco or enterprise environments, as well as essential support services, are offered. Products include media resources, digital network access, VoIP, fax, speech processing and conferencing. Support is available to help developers through each stage of their product's life cycle including pre-sales consultancy, technical support, training and marketing.

Copyright and other notices

© 2005 Aculab plc. All rights reserved. Aculab, The Groomer, Prosody, "Connected With Aculab", Aculab Worldwide Digital Connectivity and Aculab Your Connection To The Future are registered trademarks of Aculab plc. Prosody S, TiNG and GroomerII are trademarks of Aculab plc. All other product or company references or registered and unregistered trademarks are the sole property of their respective owners.

For more information, visit Aculab's website at www.aculab.com

Aculab plc, Lakeside, Bramley Road, Mount Farm, Milton Keynes. Buckinghamshire. MK1 1PT. United Kingdom. Tel: +44 (0) 1908 273 800

The information in this white paper is provided by Aculab plc, a computer telephony company, and is for informational purposes only. Nothing in this publication forms any part of any contract.

The information contained herein is based on material, which Aculab, based on its best efforts, believes to be reliable, but no representation is made as to its completeness or accuracy. Aculab make no warranties, express or implied, in this document. E&OE