ENTERPRISE-WIDE AUTOMATION IN A VIRTUAL ENVIRONMENT

Ian Cockett, Technical Director
Pebble Beach Systems Ltd., Weybridge, UK

Abstract - *Increased levels of abstraction in hardware and software layers are providing new opportunities and challenges within the Broadcast industry; particularly in broadcast automation. New software architectures provide more flexible ways of building and operating playout systems and similarly, hardware virtualization promises unheard of flexibility when deploying and maintaining it. With the general trend now being for video and audio processing to move from hardware to software, new technologies such as channel-in-a-box or integrated channel devices also stand to benefit. This paper will explore how this abstraction and virtualization can transform the utilization of the heterogeneous systems across today's broadcast facility. We will demonstrate how an intelligent over-arching software control layer can provide the link between these disparate systems and deliver real operational and technical benefits. With the relentless increase in channel counts, and constant pressure to do more with less, automation technologies are more relevant than ever.*

How we got here - A short history of automation

Since the mid-to-late 1980s broadcasters have utilized software control systems to improve the productivity and efficiency of their playout facilities. Traditionally referred to as 'playout' automation, these systems sought to increase the efficiency and accuracy of broadcast operations by automating what were previously very labor-intensive tasks. Or, they sought to increase the utilization of expensive hardware such as robotic tape libraries or CART machines in order to broadcast more channels without requiring additional expensive hardware. If two channels could be broadcast from one CART machine instead of one, it was a simple business case to argue the benefits for an automation system.

With the arrival of video server technology by the mid-1990s, automation systems were able deliver further business benefits for broadcasters by enabling high-revenue earning material to be played out from disk-based devices with greater reliability than could be achieved from tape. With the automation system managing the video server as a cache or buffer device, content could be recorded from tape in advance and any quality problems or failures, such as head-clogs, identified long before transmission. It also brought the advantage that multiple video servers could be used in parallel in order to gain even further levels of redundancy and security. The specialized broadcast video server was so revolutionary that by the end of the 1990s almost all major broadcasters were using them to playout short-form content such as commercials, trailers and other interstitial content.

As more and more use was made of video server technology, broadcasters demanded integrated content capture, ingest tools and applications to help manage the process of getting content into the disk-based systems and management tools to help them manage the quality control (QC) processes and movement of content between servers. Significant advances were also made in large capacity digital tape storage formats that made it practical for the first time to ingest once and then simply store the high-resolution files permanently, rather than re-ingesting content from the original tapes each time it was needed to be broadcast. This requirement spawned a whole new industry providing media and archive management systems for broadcasters. The long-promised convergence of broadcast and IT technologies was starting to happen.

In recent years, the focus has been on moving towards 'tapeless' production and transmission workflows. In some cases, tape has not totally been eliminated and many present-day workflows could still be described as 'less-tape' rather than 'tape-less'. However, most broadcasters now play all of their content to air from disk-based servers and most content is delivered to them as files.

Today's modern broadcast facilities are typically made up of many different IT systems: from scheduling and traffic systems, through to content management and ingest systems, archive management, playout automation and content delivery systems. It is not uncommon for broadcasters to have multiple systems from different vendors each doing a similar job. For example, multiple automation systems or archive management systems. In addition to having many different control systems and databases, present-day broadcasters also typically still have large amounts of legacy hardware that needs to be controlled.

The broadcast industry has traditionally used proprietary hardware solutions; however, the convergence of off-the-shelf IT, storage and video compression technology means that today it is perfectly possible to broadcast an entire HD channel, including complex graphics, from a single PC.

Playing HD 1080p video from low-cost hardware is no longer difficult – kids in their bedrooms can do this with their Raspberry Pis at home.

Throughout the last 20-30 years the author has seen many instances of how today's IT technology becomes tomorrow's broadcast technology. Game-changing technologies such as high performance RAID arrays using standard hard disks totally changed the way we do business today. IP networks look set to replace the proprietary serial digital interfaces (SDI) we rely on to transport video. Virtualization is another of these technologies that has become mainstream in the IT industry over the last 10 years and is set to change the way the broadcast industry is built.

DRIVING CHANGE

Before we talk more about virtualization technology, it is useful to consider what is driving the change within our industry. Rather than use new technology for technology's sake, there should be a good reason to do so.

Fundamentally, there is a need to do more with less. The growth in channel counts, and multiple methods of distributing content coupled with reduced advertising revenue puts fiscal pressure on everyone in the industry.

Reducing labor costs and operational expenditure (OPEX) and being more responsive and agile are crucial in today's competitive markets. Our industry can no longer afford to run massive systems made up of discrete single-function boxes, or racks and racks of power-hungry servers using only a fraction of their potential capacity, simply because the software was not properly designed to run more than one function in a single server, or was not designed for modern multi-core multi-threaded processors.

THE CHALLENGE

So the challenge is to provide systems that really deliver the promised business benefits; that are agile enough to cope with today's ever changing requirements. In order to maximize revenues, systems need to be flexible enough to handle many different workflows and need to be easy to change without going through time-consuming change processes.

Tomorrow's systems will still need to touch legacy hardware. Many broadcasters cannot afford to throw away their investment in traditional HD video server and 3Gbs hardware. Existing valuable resources need to be put to better use and used in smarter ways. In practice, this means that some of the 'hard' requirements such as frame-accurate real-time control of video devices cannot yet be totally ignored.

Systems need to be able to support a wide range of applications and styles of channel. From simple, jukebox-style linear channels through to complex channels that include interactivity, social media, 3D branding graphics and so on. From static channels, whose schedules are locked down weeks in advance, to live channels that change minute-by-minute. To be truly scalable and flexible we need a common platform and single user-experience that can handle this wide gamut of requirements. One of the features of many of today's channels is that they frequently change their on-screen look and offer new kinds of programming in order to capture more market share and differentiate themselves in a highly competitive market.

Virtualization provides many of the tools required to deliver these requirements.

VIRTUALIZATION IS NOT NEW.

Virtualization's origins come from 1960s mainframe computers but it is really only in the last 10 years that virtualization technologies have become mainstream.

The Gartner group defines virtualization as, "the abstraction of IT resources that masks the physical nature and boundaries of those resources from resource users. An IT resource can be a server, a client, storage, networks, applications or Operating System. Essentially, any IT building block can potentially be abstracted from resource users". [1]

But virtualization is not just about running an application on a virtual machine and gaining some benefits, it is more fundamental than that and the benefits can be more far-reaching. To understand why, it is useful to talk in more general terms about what software engineers call 'abstraction'.

ABSTRACTION

"The essence of abstraction is to extract essential properties while omitting inessential details." [2]

Software abstraction is a large subject in its own right. Abstraction underpins practically all modern software design. We often hear about certain specifics such as Object Orientated design, Client-Server Architectures, Multi-tiered design, even 'Virtual Machines'. There are many facets and benefits to all of these approaches. They all fundamentally enable software designers to build maintainable, well organized and reliable applications. By dividing an application into separate parts , software developers need not concern themselves with all of the details or complexity of all the parts. However, good software requires a full understanding of the requirements, modelling the design well and using the correct levels of abstraction.

The benefits are not purely for the writers or developers of the software. Users benefit too. Take, for example, the simple task of sending a text message. As a user, I type a message, I select an addressee and I hit send. I don't need to worry how the message is sent: as an SMS, iMessage, or over which carrier or network connection, 4G or Wifi. I simply want to send a message, so I use software that enables me to do that at a high level without worrying about the detail.

In broadcast automation we use abstraction all the time to simplify the operation of systems and make complex broadcast systems easy to use.

We can define a simple transport control that looks like a VTR control panel, for example, that enables us to control video server port or IP stream. Neither of these have anything to do with tape, but users are familiar with the concept of cueing or preparing a clip and playing a video stream, so a VTR-style user-interface works well. Another example is a software router control panel that might make a complex multi-stage multi-level video router appear as one simple device.

One of the indications of success for any piece of software is when a user remarks, "Wow, that was really easy to use". That is often the mark that the designer has done their job well.

So abstraction can help us design systems in a more flexible way and it also can simplify operations for users.

HARDWARE ABSTRACTION

The concept of hardware abstraction might be more familiar to people who maybe have already used virtualization, perhaps to run a virtual copy of the Windows operating system on their Mac computer. Or perhaps they remember the infamous "Your system is running low on virtual memory" message that plagued earlier versions of Windows. Both are forms of hardware abstraction and virtualization.

There are many types of hardware and software virtualization in use today. The technology has introduced new vocabulary such as:-

- Host – the physical machine that hosts the virtual machines
- Virtual Machine (VM) – a software container that emulates a physical machine, in which an instance of an OS is run.
- Hypervisor – configurable software that allows the virtual machine to interact with its physical host and MAY manage resource allocations and other constraints. It is, in essence, the virtualization implementation.
- I/O – Input/Output in this context is any type of input or output from a given software element

Benefits of hardware virtualization map directly onto real business benefits.

- Platform optimization – taking advantage of multi-processor, multi-core hardware platforms. Reducing the number of servers required and reducing the environmental control and power usage.

- Application agility - by removing the interdependence between OS and hardware, services can be commissioned and decommissioned much faster. VMs can also be moved between hardware platforms in a matter of seconds.

- Resilience - hardware virtualization can provide resiliency in the virtualization layer, allowing monitoring, automatic restarts, VM mirroring and a host of other options.

- Increased efficiency in hardware provisioning and utilization - virtualization layers can optimize how CPU and memory resources are shared between VMs, thereby reducing unused computing resources.

- Load balancing – Hypervisors can optimize hardware resources by managing allocation of VMs to available CPU cores. If one VM needs more resources, other VMs can be reallocated to accommodate their workloads either manually, or as a scheduled or automated process.

- Reduced CAPEX and OPEX - hardware virtualization can result in fewer, high spec servers, optimized resource utilization, reduced routine maintenance cycles, lower power consumption, less cooling, easier cable management, and more flexibility.

- Management and Disaster Recovery - VMs can be managed centrally allowing administrators to start, stop and move VMs with minimal downtime.

OTHER TYPES OF VIRTUALIZATION.

There are many types of virtualization where abstraction can be used to gain some kind of tangible benefit. This could be:-

- Network virtualization where we might separate the logical and physical network layers in order to make it more manageable.
- Desktop virtualization where we allow users to see the same desktop irrespective of which computer they are using, or
- Service virtualization where we might model an entire system or service to make it easy to simulate an entire production environment.

We can immediately see we are on our way to accessing some of the benefits of virtualization just by maybe using some virtualized hardware. But in fact, to fully access the benefits of virtualization we need to fully embrace the concept: designing our software properly to be fully distributed, to be agnostic of where it is hosted or configured and then to use best practices to host that software in a robust virtualized hardware solution.

DESIGNING SYSTEMS FOR VIRTUALIZATION

Implementing a broad-based virtualization initiative requires an understanding of what hardware virtualization provides and where there may be technology gaps.

At a basic level, hardware virtualization is designed to abstract the basic, component parts of a computer system such as the CPU cores, memory, network and storage I/O. Traditional automation systems were characterized by specifics such as large numbers of RS422 serial ports and GPIOs, in addition to broadcast-specific timecode and reference signals such as LTC, VITC and black-burst. So in order to use virtualization in our broadcast environment we still need to be able touch these 'old school' technologies from our brave, new world, so as to be able run systems that are locked to a station's video frame rate and that can touch legacy hardware.

But the power of a distributed, abstracted approach to our automation system is not just about controlling legacy hardware and RS422 devices. Modern, integrated software-based systems and sophisticated automation can do much more and can enable us to develop new kinds of workflow. Take broadcast graphics, for example, which was once one of the bastions of proprietary creation tools and bespoke hardware. From the automation perspective, managing the creation of graphics was, "somebody else's problem" and real-time control from the automation system was usually primitive at best.

Standard desktop applications such as Adobe Photoshop, After-Effects and off-the-shelf and open-source graphics renderers are already changing the way motion graphics are created and delivered for broadcast. As we are not hindered by the physical constraints of a traditional automation system, we can easily integrate these new workflows and technologies; controlling external applications such as After Effects directly from the automation layer. This enables us to manage the creation of graphics through a single integrated database within the automation system and bring back status and feedback into the Master Control and administrative user-interfaces. We can leverage our distributed architecture to control a whole farm of render boxes, if necessary, and share these resources across multiple channels. And, as all of the control layer can be virtualized, the system could even incorporate cloud resources as part of the solution. Where the control elements and controlled devices are located no longer matters.

By reducing the dependency on proprietary or bespoke hardware we will ultimately gain flexibility in our virtualized system. If we still can touch the old world of RS422 and GPIO control where this is required, we can give users flexibility and migration paths and give them the benefit of both worlds. We can focus on the business benefits – like fixing upstream workflows, whilst not needing to worry about the actual hardware that might be required for a particular channel.

Abstracting the fundamental components of the entire automation system ensures flexibility regardless of system size and type, but we must apply this abstraction to all parts of the system if we are to gain the maximum benefits. Remember, we want the freedom to host functionality when and where we choose. To do this, we must divide all of the components that make up our system into as small pieces as possible. We can then group the components into a general purpose server or 'node' as shown in Figure 1 below:-

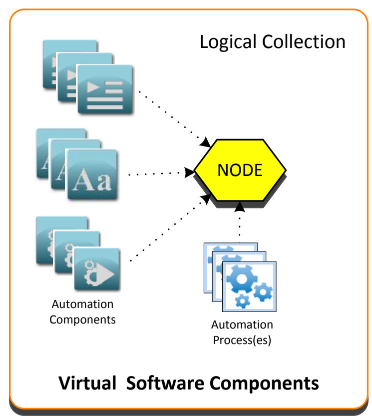

FIGURE 1. COMPONENTS ARE COMBINED TO CREATE LOGICAL NODES.

Once the logical collection of functions is designed for a particular system, they can then be combined together and installed on a physical platform whose power, memory, and physical location best suits the user, their facilities and service delivery models.

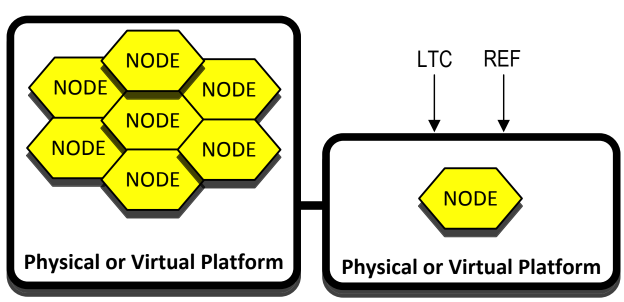

FIGURE 2. NODES MAY BE COMBINED BASED ON PHYSICAL HARDWARE OR CHANNEL GROUPS

There are real world caveats depending on the extent of Hardware Virtualization to be deployed. Platform flexibility is the key to enabling system designers to select those areas of a system that will benefit from virtualization and which, due to interfacing or facility constraints are better suited to traditional hardware hosting. These might include:-

- Real-time deterministic serial I/O devices – e.g.RS422
- Legacy time code distribution

This concept also extends to the new concept of an integrated channel playout device, or channel-in-a-box. We can design the integrated channel to be part of our abstracted architecture and simply be one of the nodes in our network.
If we can design the solution to be agnostic of the type of hardware or CPU being used, we gain flexibility and can even run our integrated channel from a fully-virtualized platform, with simply an IP stream in and IP stream out. If we have a range of graphics options that don't mandate the use of a GPU, it means that we can even deliver 2D graphics on our virtualized integrated channel as we don't need a GPU.

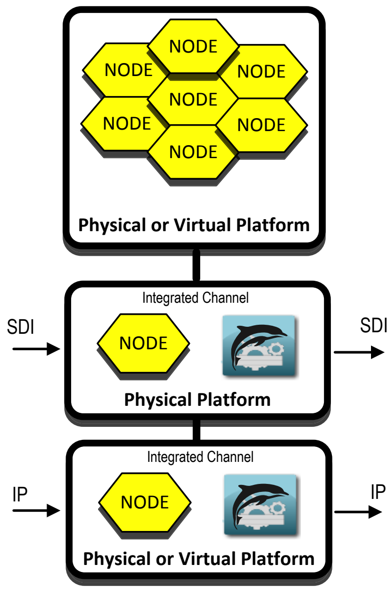

FIGURE 3. INTEGRATED CHANNELS SIMPLY BECOME NODES WITHIN THE VIRTUALIZED NETWORK

If the integrated channel must have an SDI output in order to feed downstream equipment, then our distributed approach allows that node to be hosted on a physical server containing a video card.

APPLICATION SERVER VIRTUALIZATION

The technique of Application Server Virtualization allows multiple server instances - regardless of logistical constraints - to scale services, load balance to optimize resources, and provide tiered resiliency. When correctly architected, this also improves serviceability as software updates or maintenance can be performed without system shutdown or service risk. Properly implemented, a facility is free to add, remove or re-host services behind the abstraction layer.

Using the appropriately constructed building blocks of components, nodes, and standard IT hardware, the diagram below shows a service virtualization that may minimize or eliminate bespoke devices but retain resiliency and serviceability. Users might choose to group components together based on channel groups or a service provider might build a multi-tenanted system grouped by customer.

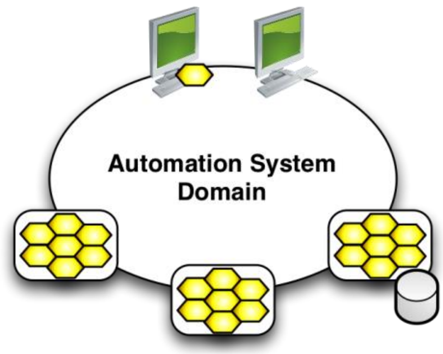

FIGURE 4. PLAYOUT AUTOMATION SYSTEM USING VIRTUALIZED SERVERS

SERVICE VIRTUALIZATION

For Playout automation, service virtualization represents the goal of complete system flexibility, multi-dimensional scalability and the ability to adapt with additional features and functions. This can present itself using traditional best-of-breed, multi-vendor solutions or integrated channel devices. [3]

GEOGRAPHIC ABSTRACTION

Once we have achieved our goal of being able to control legacy equipment, utilize the latest integrated channel technology and deploy new services at will, the next logical step is to enable users to have geographic freedom of control. Whether this is a segmented domain within the same facility, a multi-site operation, or cross-country connection, the advantages are obvious. Using our abstracted view of the world we can now move services not only between servers in our facility, but between countries or remote offices.

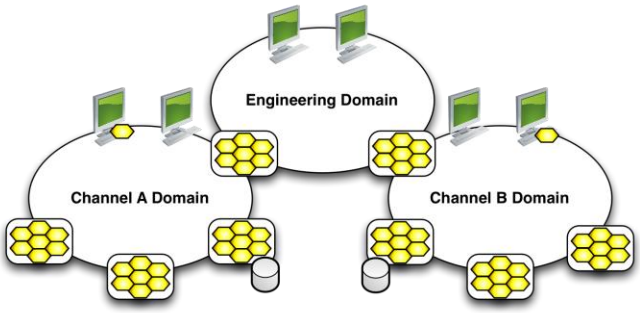

FIGURE 5. LOCAL PLAYOUT AUTOMATION VIRTUALIZED AND SEGMENTED

Controlling cost and managing resources across the television network and station enterprise are natural uses of virtualization. Figure 6 includes Disaster Recovery as one example where an additional abstraction may be the transparent use of alternative platforms such as Channel-in-a-Box (CiaB). Various applications are replicated in real time to alleviate WAN related disruptions.

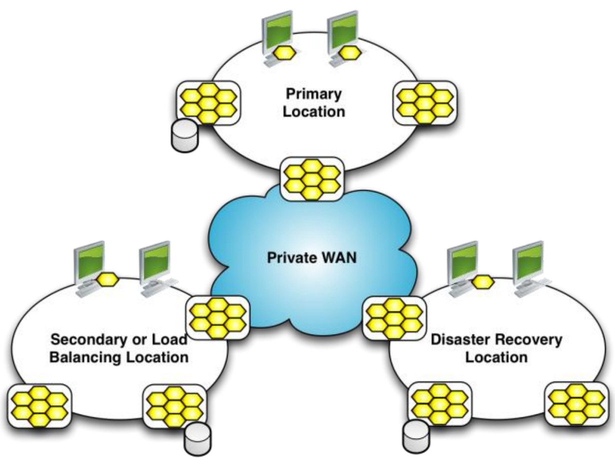

FIGURE 6. GEOGRAPHICALLY DISTRIBUTED PLAYOUT AUTOMATION

Another WAN permutation is station centralization. The example in Figure 7 includes centralized administration support, promos and graphics creation in an alternate location, with station control available locally, through sister stations or centrally.

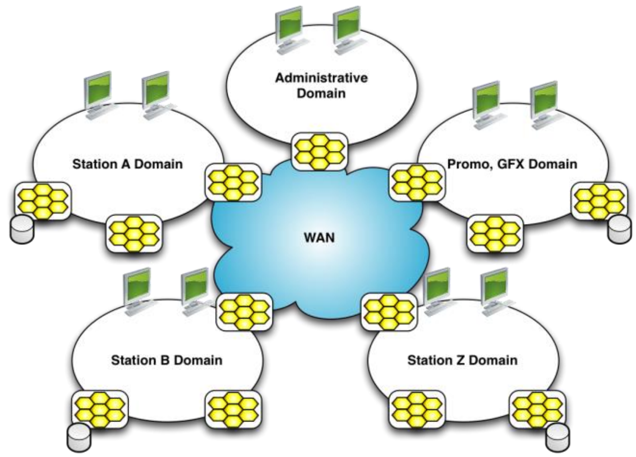

FIGURE 7. STATION CENTRALIZATION WITH GEO-ABSTRACTION

CONCLUSION

Virtualization technologies over the last 10 years have radically changed the way the IT industry now builds and deploys its systems. It is the technology on which today's ubiquitous 'cloud' services and applications are built. History has shown us repeatedly that IT technologies will continue to change our industry. Moore's law of ever increasing semiconductor densities has meant that CPU power has also continued to double roughly every 2 years. The result is that the video and audio processing on which our industry is built, is rapidly moving away from bespoke proprietary hardware to software systems. Similarly, connectivity is also moving away from SDI interfaces to standard IP networks.

The tendency in the past has to been to introduce new technology as islands within a larger infrastructure, resulting in the heterogeneous systems we see in so many facilities today. Virtualization offers a way forward to streamline and tackle this problem, but as we have seen we cannot simply take legacy systems and re-host them within a virtual environment. Software and systems must be built from the ground-up to be fully distributed to be able to maximize the benefits brought to us by this technology. Done properly, this can deliver the required business benefits and will allow broadcasters to maximize on their investments in existing and new technology.

With the relentless increase in channel counts, new types of delivery method and the constant pressure to do more with less, new automation technologies are more relevant than ever, and are the key to being able to maximize the benefits from virtualization technologies.

References

[1] http://www.gartner.com/it-glossary/virtualization/

[2] Ross, D,T, Goodenough, J,B, Irvine,C,A "Software Engineering Processes, Principles and Goals", *IEEE Computer* Vol 8, No 5.,May 1975, pp 17-27

[3] Openshaw, E, Sakata,G “Playout Automation in a Virtual Environment” *SMPTE Conf Proc;* October 2013; 2013:1-19

Author Information

Ian Cockett, Pebble Beach Systems Ltd, Weybridge, UK