Despite the simplicity of the above scenario, a number of characteristics are illustrated:

Heterogeneous Processors

The DSP and GPP have different instruction sets and potentially different data representations. The accelerators are typically 16, 24 or 32 bit special purpose devices. The accelerators may have limited code and data spaces. For example, total DSP code space for the application, operating system and communication infrastructure may be under 64K instruction words. In order to save room for applications, the communication infrastructure code footprint is ideally quite small (e.g., less than 1K VLIW instruction words). The communication data footprint is of the same magnitude.

There seems to be a trend towards 32 bit processing and less constrained acceleration devices in the future, although this is not altogether clear, in particular for low-power mobile devices.

In the example, the application is executing in user mode on a general purpose operating system such as Linux. Other operating systems such as QNX Neutrino, VxWorks, WinCE, etc may be used. The DSP is running potentially another standard operating system, or a "home grown" operating system, or perhaps in a "bare metal" environment, with no operating services to speak of.

Heterogeneous Communication Infrastructure

Figure 1 shows a network on chip connection between the GPP (in this example an ARM processor) and DSP. This may range from a simple bus to a well-designed network on chip infrastructure. However, (and perhaps more typically today), this could be an ad-hoc collection of buses and bridges, with different DMA engines and other mechanisms for moving data around the system.

Heterogeneous Application Environments

Multimedia applications are very complex, and have a number of frameworks such as gstreamer, DirectShow, MDF, OpenMax, etc. The communication infrastructure should "play well" with these environments.

For example, Figure 2 shows the OpenMax multimedia framework from the OpenMax specification[3]. Host-side communication APIs may already be defined by the framework. The MCAPI communication infrastructure is more likely suited to the tunneled communication between acceleration components.

[3] See http://www.khronos.org/openmax/