**Trends in Future Systems**

Presence of accelerator devices and applications using multiple accelerators have become common, such as Gesture Recognition, Middleboxes, Online Deduplication.

1. All devices are treated as disconnected hardware
2. Data management across devices must be done by applications
3. Redundant data copies could result in increased task latency

**Current model:** User-space application is the data hub and all data movement between devices passes through CPU.

---

**State of the Art**

`sendfile()`, `splice()` system calls in Linux copies data between files/sockets avoiding copy to the application space.

PTask proposes a dataflow model that simplifies data communication for tasks running on multiple devices.

GPUfs enables file abstraction for tasks running on CPU and GPU.

Nvidia GPUDirect framework enables direct communication between devices.

*Above systems provide data management support across limited devices. However, they lack a notion of global address space for all devices.*

---

**Design**

Goal: provide a **global address space** of data objects for the applications that hides the disconnect between the devices.

- **Data object abstraction** hiding the location of data
- **Decides the locality of data based on the access pattern**
- **Leverages hardware mechanisms like zero copy, PCI global address space**

---

**Applications**

**Data Abstraction layer**

**Data Management layer**

**Data Transaction layer**

**Devices**

**Data Object**

<table>
<thead>
<tr>
<th>Read file from disk</th>
<th>Process data on GPU</th>
<th>Send data over network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creation of data object</td>
<td>Object handle passed to tasks</td>
<td></td>
</tr>
</tbody>
</table>

**Storage** **Memory** **GPU** **NIC**