Interconnection Optimization in a Multi-Nodes Storage Architecture

Aspasia Palli and Theodore Antonakopoulos
Department of Electrical and Computer Engineering
University of Patras
Patras, 26504, Greece
e-mails: apalli@upatras.gr and antonako@upatras.gr

1. Introduction

Storage systems using solid-state non-volatile memories are the mainstream technology used in today's enterprise market. These systems are built using multiple, high-capacity Solid-State Drives (SSDs), which are integrated with the main processor and the I/O interfaces using high-speed and multi-port interconnect switches. Such a switch uses multiple interfaces consisting of a number of serial links, operating in parallel, like multiple PCIe lanes. Each SSD uses a controller with a number of high-speed data channels to the non-volatile storage components. Although such systems achieve relatively high performance, in many cases a thorough analysis reveals that the maximum measured performance is less than the maximum achievable performance of the used memory technology. This is due to the architecture used for building such systems, i.e. the use of a small number of SSDs that are based on a large number of parallel memory channels and centralized control.

A different architectural approach is presented hereafter. For the same system capacity, instead of using a few high-capacity SSDs, we propose the use of a much higher number of less complex and with lower capacity storage nodes and a mesh topology that is optimized for minimum latency and maximum I/O performance. The storage nodes have to use a small number of memory channels, and multiple and operationally independent high-speed serial interfaces for data transactions between the various nodes. The memory channels of these nodes have to be based on pin-count efficient memory interfaces, thus allowing the interfacing of an adequate number of channels per controller. All these nodes have to be efficiently interconnected so that the mean node distance from the systems' external I/Os is minimized. We'll present the basic system architecture and a methodology for designing the optimum topology for a given number of storage nodes and I/O interfaces per node. We'll also present details on prototyping such a system using the well-established PCIe technology and NAND Flash technology. The presented approach can be used for building highly efficient, scalable and expandable storage systems.

2. Design Methodology

The basic components of the proposed architecture are presented in Fig. 1. The system uses a number of storage nodes, which are interconnected by a mesh topology in order to achieve minimum node distance. Each node has a number of storage channels and a number of high-speed (a few Gbps) interfaces. These high-speed interfaces can be used either for interfacing with the external world or for data exchange with other internal nodes. The number of nodes used in the whole storage system depends on the total storage system capacity, the capacity of each memory channel per node, the number of channels per node and the used non-volatile memory technology. User requests (ie. a page program request in a block device) are processed by the system nodes without central coordination and each node, by running a distributed algorithm, decides if the page will be stored locally or has to be forwarded to another node from the set of directly attached nodes. This data allocation algorithm is based on the assumption that the network architecture has minimum average distance of all internal nodes to external I/Os and the allocation algorithm results to minimum average latency per read/write command and uniform wearing of the memory cells. During page read, short messages are forwarded in the network and the node that has stored the respective page starts a forwarding process using a shortest path approach. The data rate of each link is dynamically shared between multiple data forwarding processes.

The problem of designing the optimum architecture in such a storage system is addressed in this work. Given the number of nodes, the number of I/O links per node and the number of external I/O interfaces, we propose two algorithms for determining the optimum inter-connectivity that results to minimum average distance from all external I/Os. As distance between two nodes, we define the number of links of the shortest path between these two nodes. Concerning the I/O interfaces, the distance of an internal node from an I/O interface is calculated using the node where the I/O interface is attached. As average distance we define the

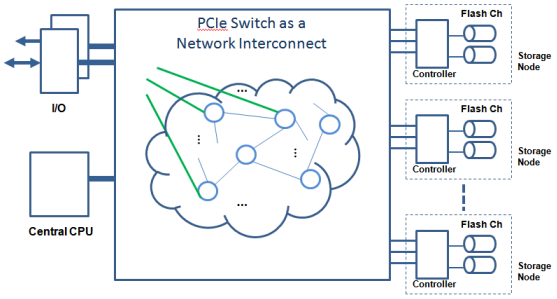

Figure 1. Storage system's architecture.

mean value of all shortest path distances between internal nodes and I/O interfaces.

For determining the optimum inter-connectivity, we developed two algorithms. These algorithms are concisely presented in Fig. 2. The first algorithm is based on the approach of designing a closed network and then the nodes, where the I/Os are attached, are determined. Therefore this algorithm uses two separate phases. In the first phase, given the total number of nodes, we design the closed network with the minimum average distance, irrespective to the number of external I/Os. This is achieved by selecting the node with the maximum number of unconnected links and a free link is used so that minimum total average distance is achieved. At the second phase, the required number of external I/Os is generated by splitting links of the closed form, so that one link is used as an external I/O, while the other site of that link is used internally to be connected with other unconnected links. Reconnection is performed by using the criterion of minimum average distance to I/O interfaces.

The second algorithm is based on a different approach. We initially select the set of I/O nodes, ie. the nodes having a link dedicated to I/O interface. Then one of the I/O nodes is selected for setting up a network where all nodes have at least a single interface connected, using minimum average distance from this single I/O node. As a next step, all other I/O nodes are connected using a single link to the existing network by satisfying the minimum average distance criterion from all available nodes. Finally all remaining links are connected using again the criterion of minimum average distance to I/O interfaces.

3. Results of a case study

Fig. 3 shows the calculated average distance of both algorithms for various system configurations when three external I/Os are used. In both cases, the average distance achieved is small compared to the total number of nodes, and that proves that designing a network with a large number of nodes and low latency (measured as number of hops) is feasible. As the number of high-speed interfaces per node increases the average distance decreases, especially when a

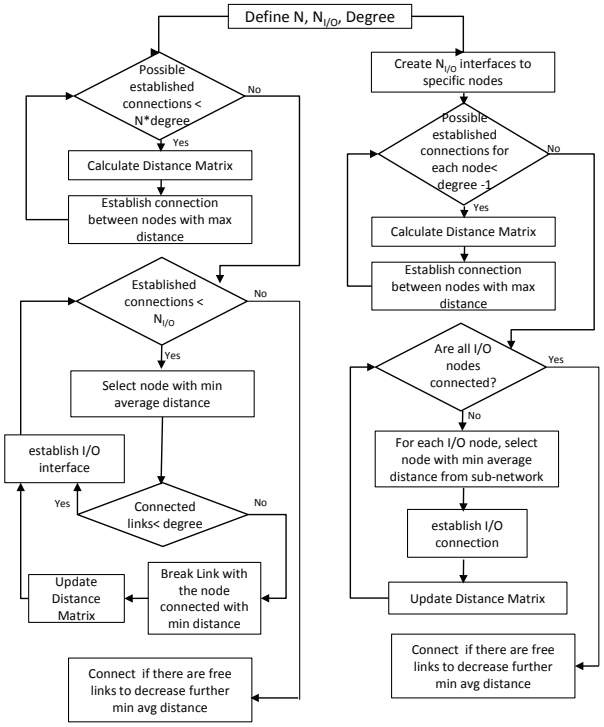

Figure 2. The interconnection algorithms.

high number of nodes is used (red is for 5, black for 4 and blue for 3 interfaces per node). In the oral presentation, we'll provide detailed analysis of the proposed algorithms and experimental results of a network with a few tens of nodes, focusing not only to the average distance, but also to maximum distance and how the response time is affected by the data allocation algorithm.

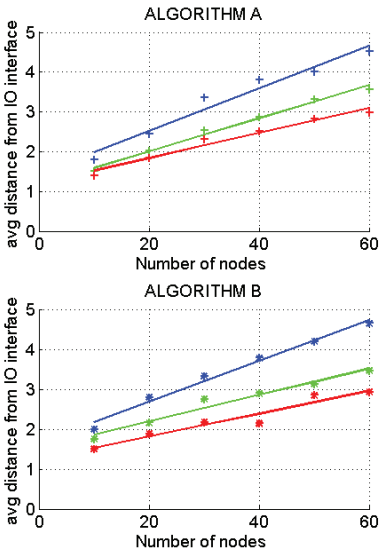

Figure 3. Average distance for various system parameters.