In order to understand (i) how applicable is general-purpose data compression for real applications (i.e., outside of previously analyzed SPEC [19] and small GP-GPU applications [7, 11]), and how severe is the observed problem – the increased number of bit-toggles, we analyze a large (231 total) group of application traces from major GPU vendor (both general-purpose and mobile applications) with six previously proposed compression algorithms. Our analysis shows that even though data compression offers a significant increase in compression ratio (e.g., more than 47% average increase in effective bandwidth with C-Pack compression algorithm [8] for mobile applications), it can also significantly increase the total number of toggles (e.g., more than 2.2X average increase with C-Pack for mobile applications). This, in turn, can significantly increase the energy of on-chip/off-chip interconnects which constitute a significant portion of the memory subsystem energy.¹

3. Our Approach: Key Idea

In this work, we aim to build a new set of mechanisms to make bandwidth compression more energy-efficient by reducing the number of toggles that significantly increases due to compression. First, we propose a new Energy Control (EC) mechanism that monitors the benefits and overheads of data compression, and decides whether it is better to send data in compressed or uncompressed form (based on both compression ratio and relative change in bit-toggle rate). The key insight behind EC is that the decision can be made locally (e.g., for every cache line) based on the cost-benefit model derived from the commonly used Energy × Delay and Energy × Delay² metrics. In this model, Energy is directly proportional to the bit-toggle number, and Delay is inversely proportional to compression ratio. Second, we define a new Local (Flit) and Global (Packet) reordering mechanisms that can be used in existing on-chip interconnects to further reduce the number of bit-toggles. Our proposed solutions are especially effective when the number of flits is small (e.g., due to data compression). Third, we propose a new Metadata Consolidation optimization for existing data compression algorithms to reduce the negative effects of inserting per word metadata into the cache line data after compression.

Our proposed mechanisms are applicable to different compression algorithms (e.g., FPC [2] and BDI [14] compression), to different communication channels (e.g., on-chip and off-chip buses), and potentially to different architectures (e.g., both GPUs and CPUs). Some of our proposed techniques (Local and Global reordering) can be efficiently applied to data transfers even without compression. We also demonstrate that our mechanisms are largely orthogonal to different encoding schemes (e.g., Data Bus Inversion (DBI) [20]) also used to minimize the number of bit-toggles, and hence can be efficiently used together to obtain the benefits of both types of techniques.

4. Novelty and Contributions

In summary, this work makes following contributions:

- We make the new observation that hardware-based data compression applied to on-chip/off-chip buses poses a new challenge for system designers – a significant increase in the number of bit-toggles after compression. Without proper care, this increase can lead to significant energy usage.

¹For example, up to 80% energy of the LLC caches is H-tree capacitance interconnects [5].
overheads when transferring compressed data that was not accounted for in prior works.

- We propose a set of new mechanisms to address this new challenge: Energy Control, Local/Global Reordering, and Metadata Consolidation.

- We provide a detailed analysis and evaluation of a large spectrum of GP-GPU applications that justify both the usefulness of data compression for bandwidth compression in many real applications, as well as the existence of the bit-toggle problem for bandwidth compression. Our proposed solutions can deliver most of benefits of bandwidth compression with only minor increase in energy consumption (in contrast to 2.2X increase in the energy consumption with the baseline compressed design).

References


