

Functional Caching in Erasure Coded Storage

Track Code: 2016-AGGA-67579

Categories:

- Computer Technology

Keywords:

- Algorithm
- Big Data
- Cache
- Computer Technology
- Data Storage

Large corporations, such as Google and Facebook, rely on erasure coding to reduce the storage cost for a given reliability. In modern storage systems, the rapid growth of data traffic, such as streaming, e-commerce, etc., has stressed underlying data storage systems. Historically, a key solution to relieve this traffic burden is caching. Chunks of popular data can be stored separately closer to end-users, reducing congestion in the network and improving delay times. Among stored data, 20 percent of the content may be accessed 80 percent of the time, so establishing a priority on this data improves functionality significantly.

Researchers at Purdue University, George Washington University, and AT&T Research have developed a novel caching framework with erasure code called functional caching, which uses erasure coded chunks in the cache. An algorithm optimizes caching and quantifies an upper bound on the mean service latency in closed-form for storage systems. Arbitrary chunk placement and service time distributions are formulated via the efficient heuristic algorithm. A simulation prototype, operated in an open-source cloud storage system, validates significant service latency reduction utilizing this caching method.

Advantages:

- Monitors placement and distribution
- Reduces latency
- Optimizes caching

Potential Applications:

- Data storage system
- Erasure coded caching

People:

- Aggarwal, Vaneet (Project leader)

- Chen, Robin
- Lan, Tian
- Xiang, Yu

Intellectual Property:

Application Date: June 30, 2016

Type: Utility Patent

Country of Filing: United States

Patent Number: 10,547,681

Issue Date: January 28, 2020

Contact OTC:

Purdue Office of Technology Commercialization

The Convergence Center

101 Foundry Drive, Suite 2500

West Lafayette, IN 47906

Phone: (765) 588-3475

Fax: (765) 463-3486

Email: otcip@prf.org