Towards smart file-based data stores

Reginald Cushing a, Spiros Koulouzisa, Adam Bellouma, Marian Bubakb,a

a University of Amsterdam, Institute for Informatics, Amsterdam, The Netherlands,

b Department of Computer Science, AGH University of Science and Technology, Krakow, Poland

Nowadays, due to the data deluge and the need for the high availability of data, online file-based data stores have gained an unprecedented role in facilitating data storage, backup and sharing [1]. Up to date, the role of these file storage systems has been, largely, passive i.e. they host files and serve files to clients upon request.

The simplistic approach of these file data stores means that they are easily deployed and integrate into other applications but may have limitations when hosted files arepart of larger distributed data-oriented computations. First, data locality plays a crucial role on the performance of a data-oriented application, file servers too far from computation or having unreliable network between computation and data will introduce bottlenecks and overhead in the running application. Second, larger computations tend to produce many intermediate result files whichcan easily inundate a file data store either from capacity or network limitations.

Our proposed approach tackles these two points by proposing a hybrid data- compute store where data stores can have a limited role in computing thus bringing together computation and data; this is extension of the concept presented in[2] and [3].

The main concept of our solution is based on the assumption that some files are not standalone ones but are related to each other e.g. two same image files with different resolution. By capturing this information at the data store as part of the file metadata we can introduce some optimization routines. From our image example we know there is a function reduce Resolution() that can map fileA to file B, thus, file A and B are linked and B is redundant so the data store can safely remove the file and re-generate it on request. As one can imagine this concept can be extended to larger computations such as work where one file is subsequently transformed into many other files which are all linked together through workflow functions.

References

[1] S. Koulouzis, A. Belloum, M. Bubak, P. Lamata, D. Nolte, D. Vasyunin, C. de Laat, [Distributed Data Management Service for VPH Applications](https://scholar.google.nl/scholar?oi=bibs&cluster=14813238901603353789&btnI=1&hl=en), IEEE Internet Computing 20 (2), 34-4, 2016

[2] R. Cushing, M. Bubak, A. Belloum, C. de Laat, [Beyond scientific workflows: Networked open processes](https://scholar.google.nl/citations?view_op=view_citation&hl=en&user=yQeKtGMAAAAJ&sortby=pubdate&citation_for_view=yQeKtGMAAAAJ:WF5omc3nYNoC), IEEE 9th International Conference on eScience, 357-364, 2013

[3] R Cushing, A Belloum, M Bubak, C de Laat, [Towards a data processing plane: An automata-based distributed dynamic data processing model](https://scholar.google.nl/citations?view_op=view_citation&hl=en&user=yQeKtGMAAAAJ&sortby=pubdate&citation_for_view=yQeKtGMAAAAJ:hqOjcs7Dif8C), Future Generation Computer Systems 59, 21-32, 2016