The Facebook Stack


Figure 1: The Facebook infrastructure stack. I omit front-end serving systems about which details are unknown. Arrows indicate data exchange and dependencies between systems; simple layering does not imply a dependency or relation.

In addition, there are several papers that do not directly cover systems in the Facebook stack, but describe workloads, techniques or data centre hardware:

- Descriptions of the physical design of Facebook’s server machines as of 2011 [FHL+11] and data centre network architecture as of 2013 [FA13].

- Another paper on the HBase back-end for Facebook messages [ABC+12] and a measurement paper looking at the HDFS-level usage patterns of this HBase deployment [HBD+14].

- Papers on the use of erasure codes in HDFS at Facebook [RSG+13; SAP+13; RSG+14].

- Several papers analysing the Facebook memcached workload [AXF+12] and evaluating new sampling strategies to improve hit rates in memcached [LLD+13].

- A study of Facebook’s wide-area photo caching infrastructure [HBR+13].

- A description of how Facebook uses shared memory to persist in-memory state across restarts of Scuba server processes [GCG+14].

- The HipHop Virtual Machine (HHVM) is a JIT compiler and runtime for PHP code heavily used in front-end page generation [AEM+14]. Previously, Facebook used a source-to-source compiler (also called “HipHop”, HPHPc) to transform PHP into semantically equivalent C++ code that can be compiled into native code [ZPY+12].
Bibliography


