Who is using HBase:

<https://en.wikipedia.org/wiki/Apache_HBase>

Spotify:

<https://apachebigdata2015.sched.org/speaker/josh_baer.1u28lm06>

Netflix:

Airbnb:

<http://www.slideshare.net/HBaseCon/apache-hbase-at-airbnb>

Facebook:

Uses it in messaging platform: <http://www.theregister.co.uk/2010/12/17/facebook_messages_tech/>

The new Facebook messaging system uses Hbase to store the text and metadata for the messages as well as the indices needed to search these messages – an epic amount of data. Even before the new messaging system was rolled out, Facebook was juggling about 15 billion on-site messages a month (about 14 TB) and 120 billion chat messages (11 TB). It needed about three times more space to store all that data, with the average message going to multiple people, and as the new system adds good, old-fashioned email (from @facebook.com addresses), the storage requirements will only expand. HBase doesn't store everything – Haystack handles attachments and overly large messages – but it handles most of the data that's frequently updated.

Facebook also picked HBase because it's designed for automatic failover. When a server goes down, it automatically reverts to another – at least in theory. "When you're talking about a lot of data, you're obviously talking about a lot of machines, and when you're talking about a lot of machines, failure is the norm," Ranganathan says. "When failure is the norm, we want to be able to automatically serve that data without too much intervention or it taking too much time."

At the same time, Ranganathan and crew like the way HBase works to prevent failure. As he puts it, when HBase breaks data into pieces and distributes them across a cluster, it puts multiple and completely separate data "shards" on each machine. This means that if one server dies, its work can be picked up several different machines and not just one. In other words: better load balancing.

If you put your data shards on just a few machines and a machine goes down, Ranganathan says, the burden is picked up by a single server, and this may end-up toppling other servers like dominos. "[The second server] dies, and then two machines' slack has to be taken up by a third, and this thing cascades all the way down," he says. "HBase shards the data into a lot of virtual shards and puts multiple shards on a single machine. So a machine dying spreads [its data] across multiple machines and the utilization of your entire cluster goes up."

Linkedin: