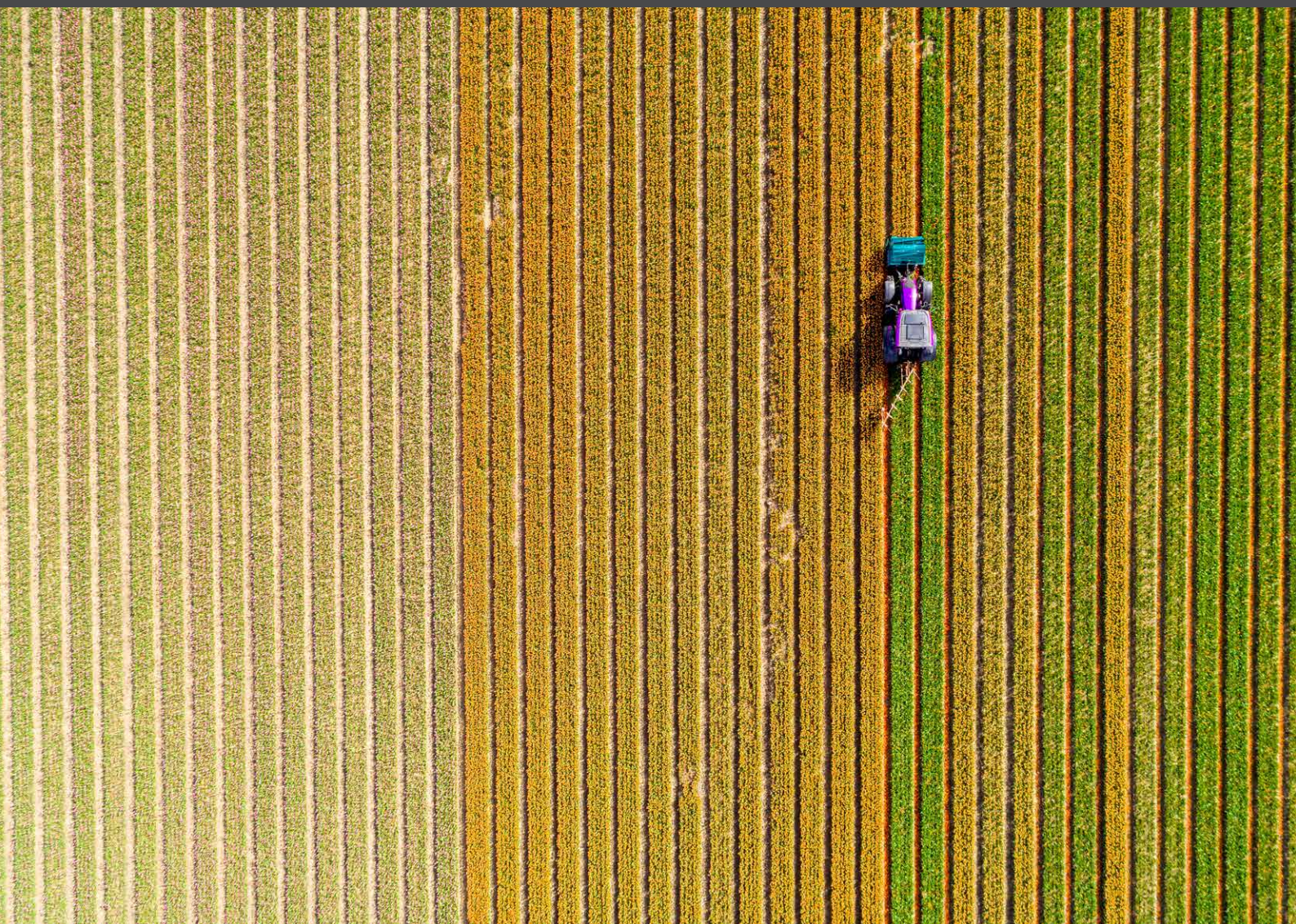
SciBite
an ELSEVIER company

Use case

# SciBite and Hadoop: Transforming big data



SciBite
an ELSEVIER company

# SciBite and Hadoop: Transforming big data

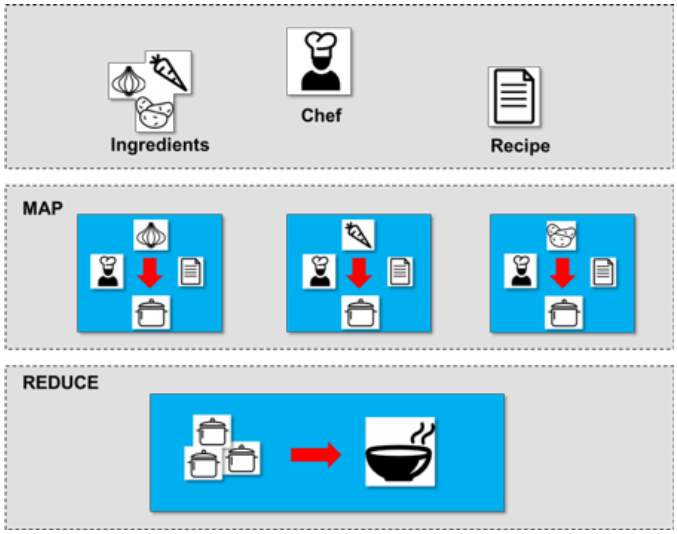
**While the hype surrounding Artificial Intelligence and Machine Learning seems to have relegated "Big Data" to yesterday's news, the challenge of processing large data collections has not gone away. Platforms such as Apache Hadoop are the backbone of both big data and machine-learning activities within many major life science enterprises. Hadoop systems can hold billions of data objects (documents, spreadsheets, binary files, etc.) but suffer from the common problem that such objects can be hard or organise due to a lack of descriptive meta-data. As an expert in scientific search, SciBite has been engaged in a number of such projects with the aim of improving the discoverability of this vast resource. Our TERMite and SciBiteSearch platforms employ a class-leading approach to embed ontology-based indexes across your data, with all of the many advantages this enrichment brings.**

As part of these efforts we are often tasked with working in a Hadoop environment and thought it might be valuable to share our experiences and describe how our platform interacts and delivers real value for our customers in this context.

## Hadoop Primer

If you're reading this it's likely you have a good idea of Hadoop and concepts such as Map/Reduce. If not, there are thousands of websites that can give you a deep introduction to the topic, starting with the Hadoop home page. Briefly, Hadoop itself is defined as a free, Java- based programming framework that supports the processing of large data sets in a distributed computing environment. The "distributed" part means that a Hadoop cluster can contain many individual compute/data nodes and any given file may be found somewhere on those nodes. You don't need to know where, the Hadoop system will manage that for you. For the "processing" part, the key concept here is that of "Map/Reduce", a technique that is devised specifically for processing large datasets in parallel on a cluster.

There are many tutorials on Map/Reduce but the basic principles can be described using all manner of analogies. Here we'll use the concept of a chef cooking their favourite meal. In a non-Hadoop situation we may have only one chef, cooking a single meal at a time, not something suited to say massive parallelisation. In a Map/Reduce environment, we have one chef (master) coordinating multiple chef's (workers) each in their own kitchen (node). The ingredients (data) are split between the nodes, with each worker receiving a specific task in the overall recipe to perform (map function – where the data is analysed). Once each node has done its task, the results are collated and optionally further combined (reduce function) into the final result.



SciBite
an ELSEVIER company

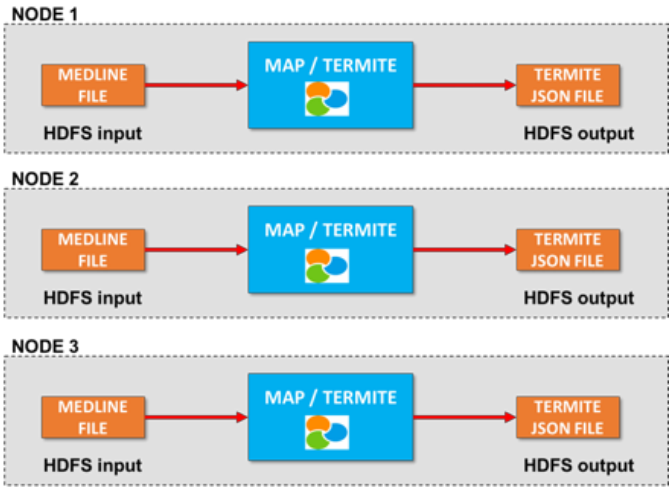
## Where SciBite fits

To bring our food analogy back to a real use-case, imagine a situation where our ingredients are perhaps millions of documents and our recipe is to perform deep semantic analysis with TERMite. By increasing the number of nodes on the cluster, we should be able to get through these documents in shorter time, the more nodes we have, the faster we can go.

However, running any kind of meaningful analysis is not straightforward, one must define just how the task will be parallelised: how should the documents be split; should TERMite be installed on each node and how should it be configured; how should the task be invoked and how should the results be handled, what does the "reduce" step look like? While there is often some bespoke set-up required due to differences in customer environments, we have established a robust operating procedure for performing this kind of analysis to speed up TERMite deployments on Hadoop.

At a very high level, the TERMite engine can be seen as an operator transforming content (e.g. Medline document to JSON file containing semantically enriched Medline articles). In Map/Reduce terms, this operator is the map function reading files from the Hadoop file system (HDFS), transforming them with semantic annotation, and writing these results back to some location on the HDFS. The Hadoop framework in turn, is in charge of executing the map functions in a co-ordinated manner across multiple nodes in parallel.

The flexible and extensible architecture of the SciBite platform, including TERMite, means that such a deployment is relatively straightforward requiring us only to create a small Java program that links TERMite into the Map calls within a Hadoop workflow. We've worked hard to optimise this, using data streaming to reduce the amount of RAM and CPU required for each map function, allowing us to run more threads in parallel. As shown in the graphic below, Hadoop orchestrates the splitting of a large set of documents (in this example the entire Medline) dataset, sending sub-sets to each node on the network where TERMite can run its analysis. If we want faster processing, we simply increase the number of compute nodes, Hadoop will automatically adjust its splitting and processing to take full advantage of the increased capacity without any input from the operator.



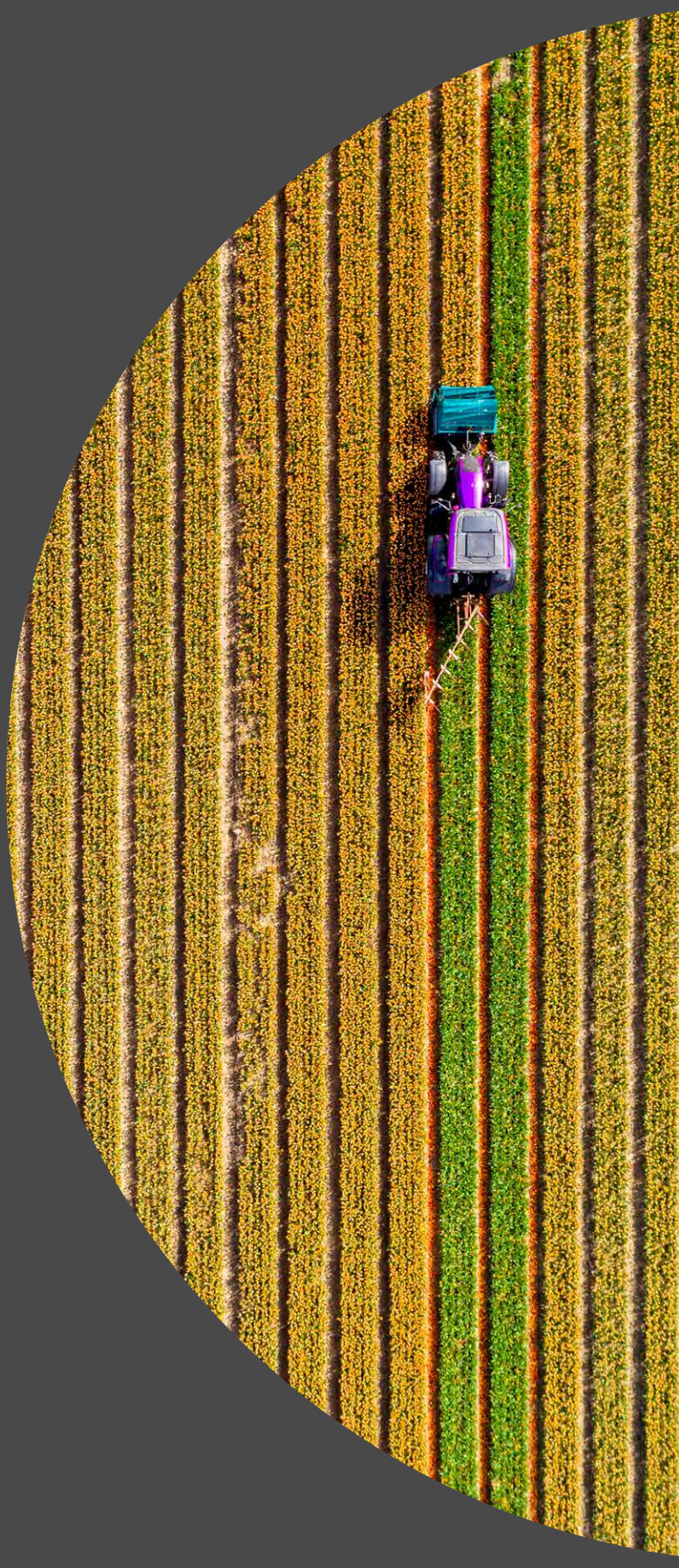

## The outcome

So, what does this give us? Perhaps this is best illustrated by some metrics. As many will know, TERMite can process at speeds around 1 million words per second on a standard CPU. A typical run on a single desktop machine can process the entire Medline database (27 million short articles) in around 5-6 hours depending on disk type, CPU load etc. Of course, Hadoop does not "speed up" TERMite on any individual node, it should run at the same sort of speed. But what Hadoop brings is a simple, extensible way to run many analyses in parallel. In our local environment we routinely use a Hadoop 2 cluster running YARN and HDFS, each consisting of 3 machines with 32GB of RAM, 16 cores and 500GB of disk.

Performing the equivalent analysis in this environment brings the time down to less than an hour. That's all 7 billion plus words of the world's biomedical knowledge processed with rich text analytics in less time than most company meetings. We're pretty proud of that! Critically, by integrating naturally into the Hadoop ecosystem, this analysed data is now ready for further processing such as statistics, network analysis or machine learning.

We hope this short summary demonstrates the value Hadoop+SciBite can bring in large-scale processing and highlights the unique position of our platform as a toolkit designed to integrate with many different systems, Hadoop being just one of many. For more information or technical discussion on this or any other integration, please get in touch via our website at **https://www.scibite.com/contact-us/**

SciBite
an ELSEVIER company



SciBite's data-first, semantic analytics software is for those who want to innovate and get more from their data. At SciBite we believe data fuels discovery and we are leading the way with our pioneering infrastructure that combines the latest in machine learning with an ontology-led approach to unlock the value of scientific content. Supporting the world's leading scientific organisations with use-cases from discovery through to development, SciBite's suite of fast, flexible, deployable API technologies empower our customers, making it a critical component in scientific, data-led strategies. Contact us to find out how we can help you get more from your data.

To learn how SciBite can unlock the value of your data, speak to one of our experts today or email us at contact@scibite.com

**Head Office:**
SciBite Limited
BioData Innovation Centre
Wellcome Genome Campus
Hinxton, Cambridge CB10 1DR
United Kingdom

www.scibite.com
contact@scibite.com
LinkedIn: SciBite
Twitter: @SciBite
+44 (0)1223 786 129