Forseti: Dynamic Chunk-level Reshaping for Data Processing on Heterogeneous Clusters

Researchers at the George Washington University have invented a novel algorithm for large-scale computing frameworks running on a distributed heterogeneous system to process large amounts of data. The algorithm minimizes job execution time by balancing residual workloads in heterogeneous environments and provide significant improvement in performance. The algorithm has the ability to reshape data chunk size processed by heterogeneous machines on the fly and to dynamically balance the workload assigned to parallel tasks. Our preliminary results show that such algorithms could outperform other frameworks such as Hadoop and SkewTune by up to 68% and 50%, respectively.

Data-intensive computing frameworks typically split job workloads into fixed-size chunks, allowing them to be processed by parallel tasks on distributed machines. Ideally, when the machines are homogeneous and have identical speed, chunks of equal size would finish processing at the same time. However, such homogeneity cannot be guaranteed in practice. Computing machines in datacenters are often heterogeneous in terms of hardware configuration, software system, and network conditions, all of which result in diverging processing time for chunks belonging to the same job. Such divergence, together dynamics and uncertainty during chunk processing, can lead to significant performance degradation at job level, such as long tails in job completion time due to residual chunk workload and stragglers.

Researchers at the George Washington University have invented a novel processing scheme addressing the above mentioned shortcomings in the state of the art called Forseti. This novel scheme is able to reshape data chunk size processed by heterogeneous machines in a dynamic fashion, and as a result, mitigates residual workload and stragglers to achieve significant improvement in performance. Forseti does not require any a priori knowledge of the machine configuration nor job statistics. Instead, it infers such information on the fly and adjust data chunk sizes at runtime, making the solution robust even in environments with high volatility. In its implementation, Forseti exploits “Java Virtual Machine reuse” feature to avoid task start-up and initialization cost associated with launching new tasks. Forseti has been tested on a real-world cluster and its performance has been evaluated by using several realistic benchmarks. The results show that Forseti outperforms a number of baselines, including default Hadoop by up to 68% and SkewTune by up to 50%.

Applications:

- Data centers
- Supercomputing and managing computation over large computing facilities

```
Algorithm 3: commForseti()
1: if $\mathcal{S} = \{\emptyset\} \& \mathcal{T}_u = \{\emptyset\} \& \tau > \varepsilon$ then
2:     for $n \in \mathcal{T}_f$ do
3:         $n.commit = \text{true}$
4:     end for
5: else if $\tau \leq \varepsilon$ then
6:     $n.commit = \text{true}, \forall n \in \mathcal{T}$
7: end if
```
Advantages:

· Outperforms other frameworks such as Hadoop and SkewTune by up to 68% and 50%, respectively

· Minimizing job execution time by balancing residual workloads in heterogeneous environments and provide significant improvement in performance.

· Reshaping data chunk size processed by heterogeneous machines on the fly such as to dynamically balance the workload assigned to parallel tasks.

Inventors

Prof. Suresh Subramanian

Prof. Tian Lan

Sultan Alamro