**Individual Project 3: Python Key-Value Program**

**CS356- Foundations of Big Data Analytics**

**5/30/2025**

**Nickolas Marquez**

**How this Would Work in a Distributed Setup**

If we had a super large dictionary-think millions of key-value pairs- it would be too much for one computer to handle quickly or efficiently. That’s where something like a distributed setup comes in. We would break up the dictionary across a group of computers or Nodes, and each one would manage a chunk of the data. This way, the work can be done in parallel, speeding things up.

For example, let’s say we want to look up the capital of a specific state like “Texas” Instead of searching through the entire dataset on one machine, the system can direct the request to the node that’s responsible for the portion of the dictionary that contains Texas. That way, we avoid unnecessary searching a get faster If we need to update or change the capital, that request is also sent directly to the right node.

Now, if we want to do something bigger- like list all state names or all capitals- each node can process its chunk of the data and send back its part. Then the main system combines all those results into one final list. The kind of pattern is very similar t o hoe systems like Hadoops MapReduce sing a forklift to move a backpack.

Let’s be real. For something like this little script with 50 key-value pairs, using a distributed system is like using a forklift to move a backpack. It adds a lot of extra work- making sure nodes are connected properly, handling communication delays, managing backups, and so on. Its just not necessary for small jobs.  
That said, when the data gets massive, think social media apps, e-commerce platforms, or smart devices sending data 24/7- distributed system becomes incredibly useful. You can store way more data, process it faster, and stay online even if one part of the system fails.

So, while this key0value example is small, it helps us understand how these concepts apply to real-world systems. Whether it powering cloud storage, streaming services, or global databases, the basic idea stays the same: divide the data, work in parallel, and combine results efficiently. That’s the core of bog data processing.

**Python Program:**

 

