Performance Comparison of Parallel Implementations of Cayley and SHA Hash Functions

By: Sourabh Kulkarni
Advisor: Prof. Alan Kaminsky

Introduction

- Most widely used hash functions are sequential algorithms and hence, don't make best use of available resources of modern computers.
- Parallel computation of hash value may greatly reduce the total running time of the algorithm.
- The aim of this project is comparison of parallel implementations of Cayley hash function and SHA-2 family of hash functions.

Cayley Hash

Cayley hash function uses two linear equations over Fp (field P) to hash every 0 and 1 bit of input bit string. The hash is calculated using composition of these individual functions. The final digest of length 512 bits is constructed using coefficients of resulting linear equation.

SHA-256 and SHA-512 are computed with 32 bit and 64-bit words respectively.

<table>
<thead>
<tr>
<th>SHA-256</th>
<th>SHA-512</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message block size</td>
<td>512</td>
</tr>
<tr>
<td>Digest size</td>
<td>256</td>
</tr>
<tr>
<td>Rounds</td>
<td>64</td>
</tr>
<tr>
<td>Security</td>
<td>128</td>
</tr>
</tbody>
</table>

Parallel \( j \)-lanes Hashing

Framework used to parallelize hash functions.

- Message is split across cores in round-robin fashion
- Each core has its own buffer memory
- One extra thread fills the buffers for rest of the threads

Conclusion

- As per our implementation and experiments, SHA-2 is faster than Cayley Hash functions when data is in memory. While data is being streamed, performances of both the hash functions are very similar.