Going Back to the Future with Camelot

Bronson Bouchard
Stewart Grant

Amanda Carbonari
Fabian Ruffy

1 INTRODUCTION

- processing data at scale is hard
- scientists, not just computer scientists process data
- python, and C are more understood / natural than spark/hadoop/-naiad
- the state of current racks allow for comprable performance with DSM
- (small contribution) These system admit failures, but they require expertise to reason about. Applications should be reasonably fault tolerant to prevent heartache, and confusion.

2 BACKGROUND

- treadmarks
- piccilo
- ramcloud
- distributed data processing (Hadoop/pregal/spark/dryad/-naiad)

3 SYSTEM

- prior system
- threading
- page eviction
- faut tolerance

4 EVALUATION

We measured various performance aspects of Camelot to measure its scalability and competitiveness with existing data processing platforms. We developed a set of benchmark programs with diverse memory access patterns, to measure the quality of our page eviction strategies, threading performance, and raid overhead.

4.1 Experemental setup

4.2 Threading Performance

4.3 Paging Performance

4.4 RAID Overhead

5 FUTURE

6 CONCLUSION