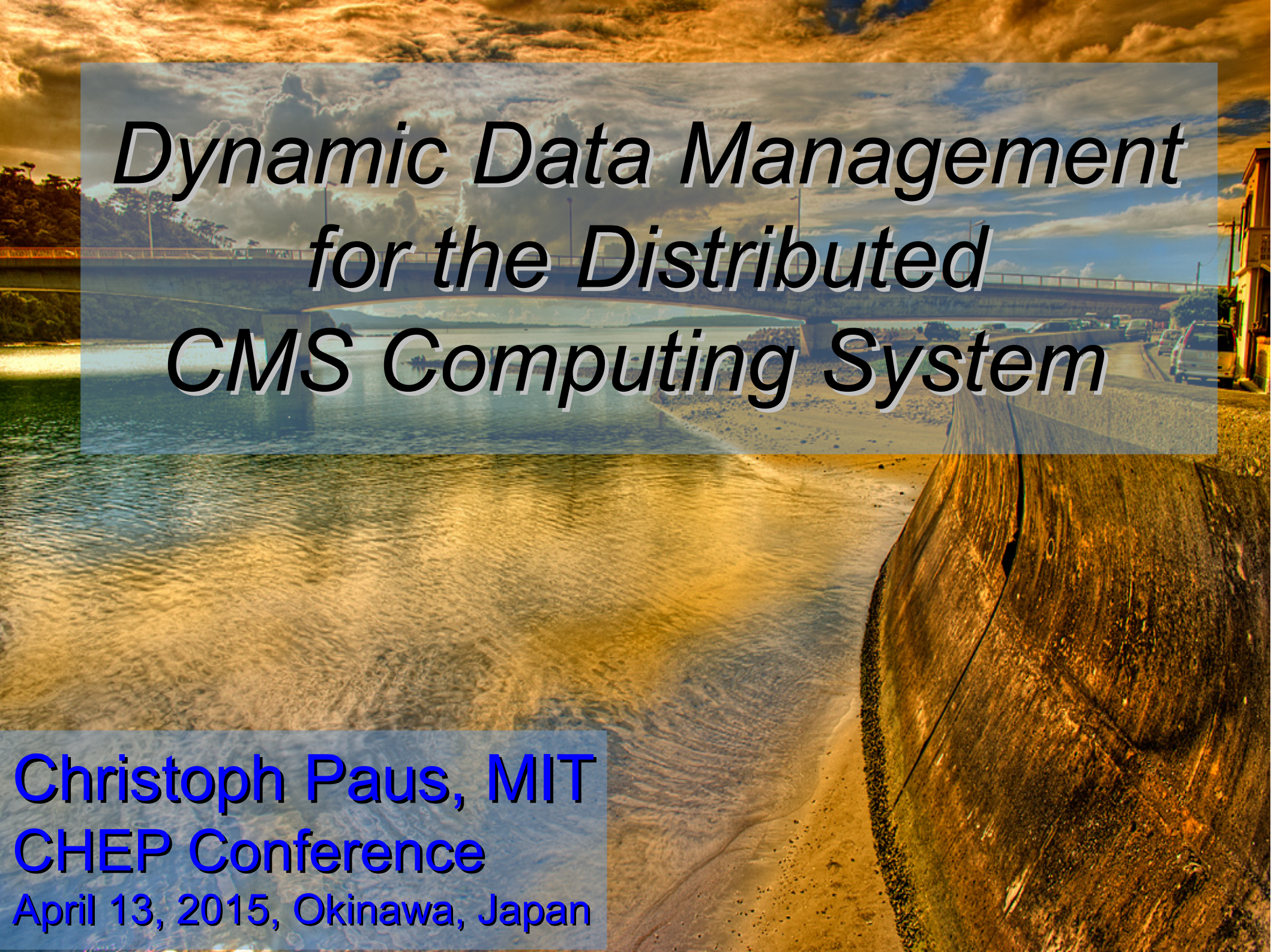


Dynamic Data Management for the Distributed CMS Computing System

Christoph Paus, MIT

CHEP Conference

April 13, 2015, Okinawa, Japan

Data Management in Distributed Systems

The Problem

- Disk storage is expensive and has to be optimally used
- Data has significantly different and time dependent popularity
- Conflicting and overlapping interests between different communities in CMS
- Manual management is labor intensive and sub-optimal

The Solution

- Create common cache pool distributed over all sites (T1+T2)
- Maintain at least one copy of all relevant data and replicate according to global popularity to optimize data access
- Replica creation and deletion based on ranking algorithms

CMS Disk Usage during Run 1

Tier-1 centers

- disk storage dominated by ‘official data samples’ (phedex)
- disks automatically integrated with tape

Tier-2 centers – typical

- official samples maintained manually via phedex
  - CompOps space ~ 30 TB temporary MC sample space
  - central space ~ 250 TB (general samples, centrally managed)
  - physics space ~ 3x250 TB (physics specific, data managers)
- user samples maintained via some flavor of srm tools
  - unmanaged space (2-4 TB per user recommended)

Disk Usage Issues during Run 1

Tier-1 centers

- did not allow user jobs – *fear of tape staging*
- as a result disks were not available for analysis

Tier-2 centers

- users seemed to need more space
- physics group space difficult to maintain:
  - manpower intensive to manage
  - storage of unpopular samples and/or potentially unused space
  - inefficient usage of disk space
  - work intensive deletion campaigns

We addressed these issues!

Disk Usage Model for Run 2

Tier-1 centers

- logically separated Tier-1 center disk and tape system
- use Tier-1 in a *Tier-2-usage* mode: no automatic staging

Tier-2 centers

- extended user managed space: ~ 40%
- centrally managed space: ~ 60%
- joined all official samples in centrally managed operation
  - no more need for physics group data managers
  - remove partitioning of storage space
  - disk space usage as efficient as model for distribution we implement
- central management – automatized based on popularity metric
  - creation of additional sample replicas of popular samples
  - removal of existing sample replicas of unpopular samples

Status – IntelROCCS Package

Included sites

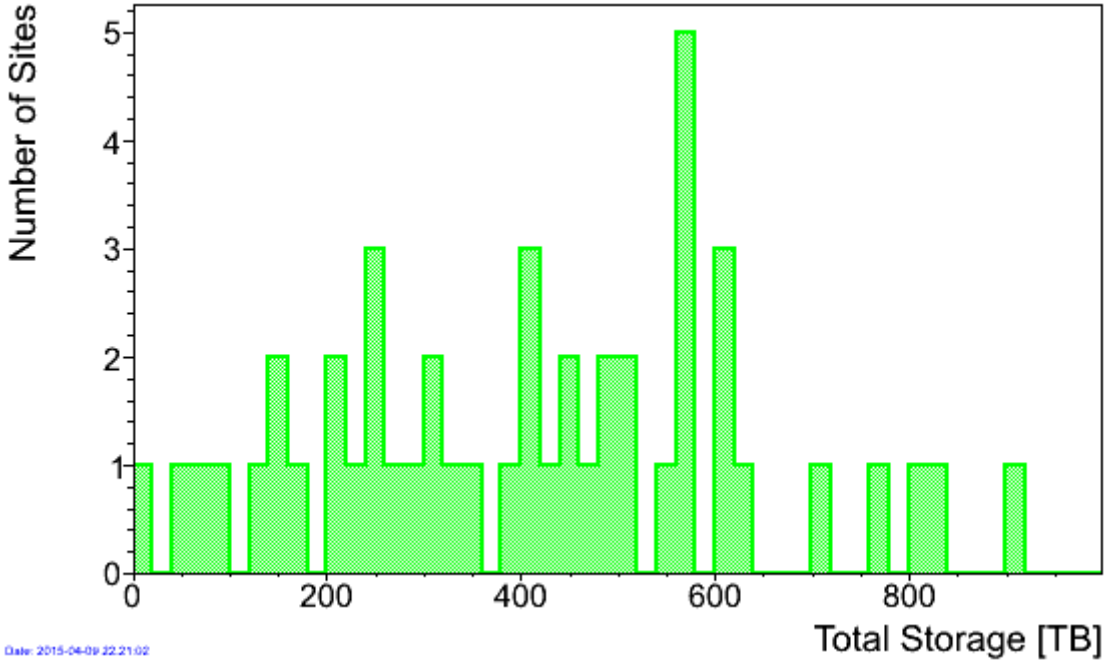
- since June 2014: all Tier-2s
- since March 2015: all Tier-1s
- sites are full to at least 80%
- last copy space ~ one third

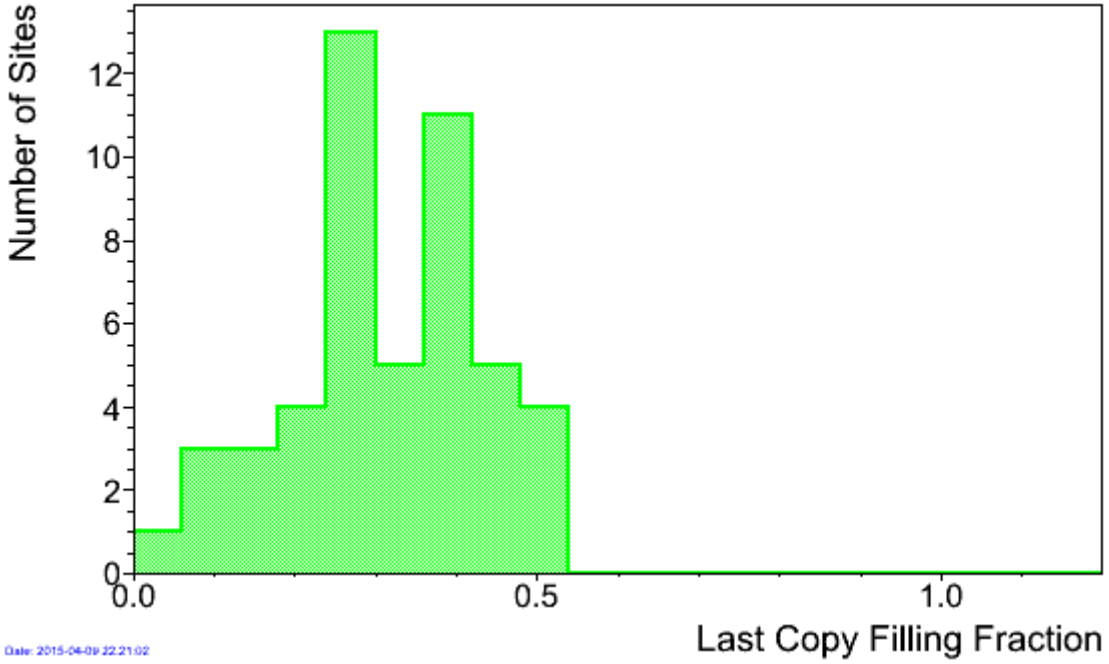
Core Components

- cache release - *Detox*
- data placement - *DataDealer*
- popularity infrastructure
- data injection and cleaning
- site caretaker - *UnderTaker*

Software/Services

- first release Mar 2014
- stable operation June 2014
- first monitor added Sep 2014
- site caretaker Sep 2014





Cache Release – Detox

Goal

- keep sites always available to receive new replicas
- disks remain full at least at set: *minimum disk quota*
- General idea: consider all samples for cache release

Special case

- maintain a set number of last dataset copies (for now 1 last copy)
- location of last copy is dynamically allocated in each cycle

Process cycle (each ~4 hrs)

- global view is created (list of datasets at each site)
- location of last copies determined and datasets removed from lists
  - last copy algorithm aims for equal last copy fraction at all sites
- ranking algorithm applied to datasets at each site (local ranking)
- global ranking created from average of local rankings
- if trigger met at site (storage passed *maximum disk quota*): release least valuable datasets until *minimum disk quota* is reached

Local Ranking Algorithm

Goal

- release samples that have not been used in a while
- keep samples actively used – more longer term

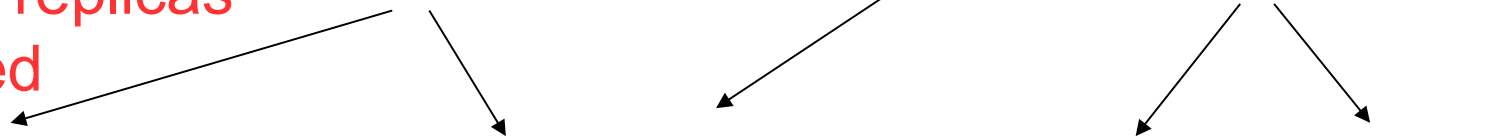
Algorithm

highest ranking replicas are first removed

protect fresh replicas

low ranking for usage

protect larger replicas



```
rank = (1 - l_Used) * (t_Now - t_Created) + l_Used * (t_Now - t_LA - n_Access/size) - size/100
```

```
t_Created   - date/time the dataset appeared on the site
t_LA        - last access date/time
t_Now       - current date/time
l_Used      - logical whether sample was used at all (0 - not used, 1 - used)
n_Access    - number of times sample was used
size        - sample size in GB
```

Global ranking: average of local rankings per dataset

Data Replication – DataDealer

Issue

- many datasets with varying time dependent popularity
- large multi-site computing system (CPU/storage)

Solution

- replicate popular datasets until similar in popularity with others
- each dataset: on tape and at least one disk copy (last/custodial)
- launch dataset replication to one or more sites based on dataset usage pattern (popularity)
- cache release (Detox) will keep sites available
- presently CMS computing system rather static: difficult to test
- pressure is starting to develop and system seems to react well

Plan

- ideally: optimize use of disk space according to a given metric
- what is the metric? suggestion: most overall CPU time used
  - There are many others one can think of!

DataDealer: Ranking Algorithm

Goal

- identify and predict if possible 'hot' (popular) datasets
- release strain on the system by creating addional replicas
- pro-active and generally speaking short term view

Algorithm (Rocker Board)

- dataset popularity normalized for each site
  based on usage during last week
  normalized to amount of free space at site
- average dataset popularity is obtained by averaging all sites
- samples with popularity beyond average are replicated (which reduces popularity)
- data volume per day is capped to avoid overreaction

Operations

Services

- Detox, DataDealer, Undertaker and Monitor – sysV services
- logging and status information published on web server
- worst failures – investigated and failsafe mechanisms implemented
  - Detox – complete deletion of data
  - stage back from tape
  - potential loss of days – inefficieny
- potential realistic failures – warning mechanisms implemented:
  - Detox – not running, sites fill up
  - DataDealer – not running, suboptimal usage
  - Popularity services – not running, suboptimal usage
  - typical time constants ~ several hours to a day

Experience so far

- no major failure as of yet – a number of small issues

What is our Performance?

System works

- users are happy, Data Managers were let go

Test – CSA14 period

- July and August when there was some planned activity

Data popularity [hits per file]

- 7.5 - CSA14 samples, 1.7 - Other samples

Data replication [replication factor]

- 1.83 - CSA14 samples, 1.43 - Other samples

How much did this help?

- system did the right thing
- maybe not enough? we have no metric to measure this so far!
- .... but we should have all information to develop it!
- part of monitoring/optimization project that has started

Conclusion

Dynamic data management – IntelROCCS package

- essential to make optimal use of storage and computing resources
- Tier-1 disk/tape separation completed and working
- entire CMS computing system included (T1s and T2s)
- cache release and data replication in stable production
- site caretaker running in production
- initial data assignment system in production
- detailed logging and debugging, basic version of monitoring
- major reduction in manual work – data manager are in retirement!

Plans

- extend data to describe system and include in monitor
- develop independent metric to measure global system performance
- optimize algorithm for replication and cache release
- machine learning techniques could be used
- computing system simulation to better understand performance