Status of SAM Deployment at CDF

Scope: Implement and deploy SAM as the main Data Handling tool at CDF

CD Project Status Meeting

Presented by K. Genser

03/08/2005

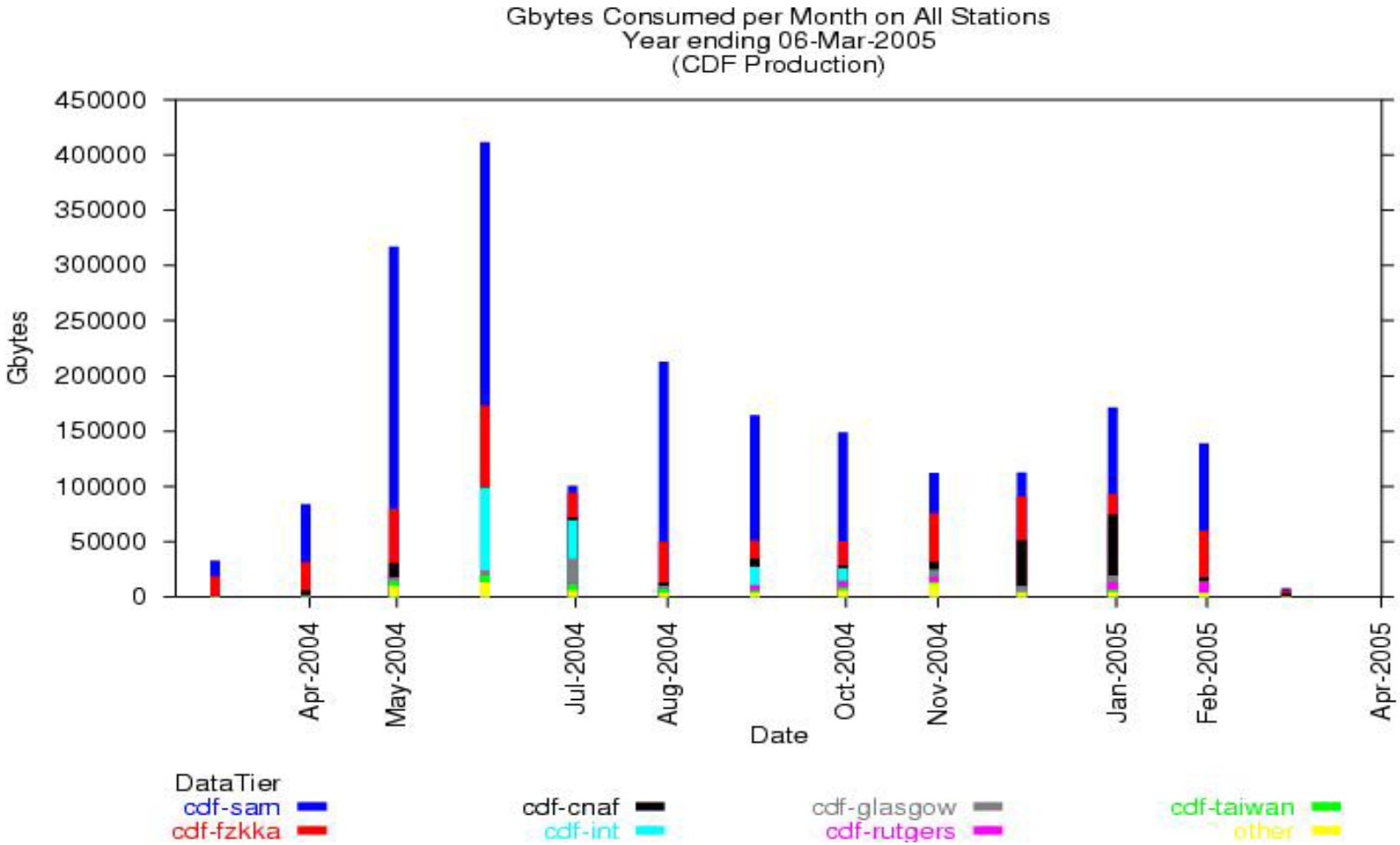
SAM Integration into CDF Analysis Infrastructure

- SAM fully integrated into AC++ analysis framework
- Local CAF usage
  - Users can submit CAF jobs specifying SAM as data handling method and define SAM datasets as input data
    - mainly integration and load tests, very light use recently, predominantly by expert users (mainly due to recent, till about a month ago, instability of SAM version v6 dbserver)
- Strong remote usage for analysis and Monte-Carlo generation of selected datasets using SAM v5 (converting to v6)
- Achieving v6 dbserver stability was a very important milestone (it also included ability to lock files at remote stations)
- Prototype SAM-farm functional – see Suen's talk
- SAM accounted for about ¼ of dCache recent input I/O (including tests)

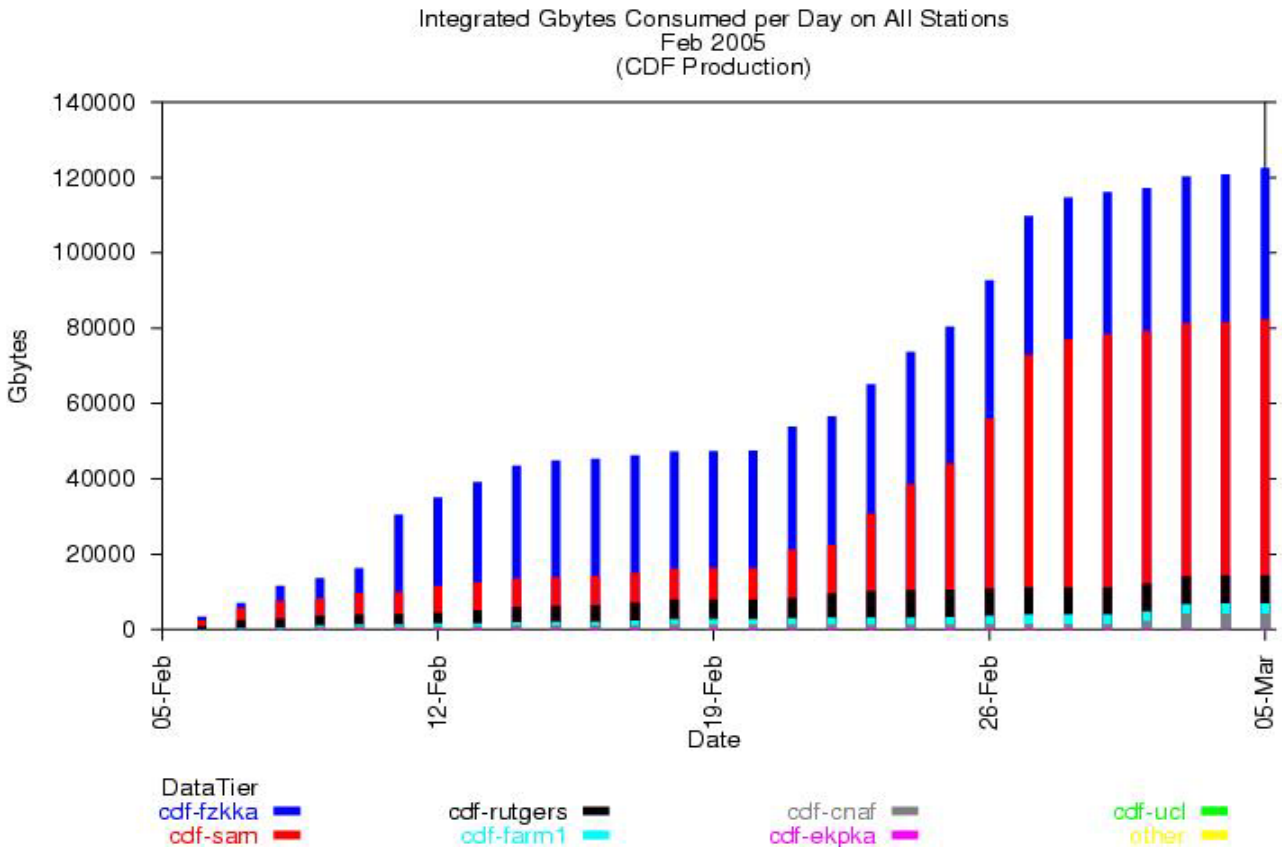
Examples of SAM usage

- Remote SAM Stations with significant amount of data cached on disks
  - cdf-cnaf - SAM v6 ~24TB
  - cdf-fzkka - SAM v5 ~23TB
  - cdf-taiwan - SAM v5 ~2.8TB
  - cdf-rutgers - SAM v6 ~2.5TB
  - Compare to permanently cached files at FNAL CDF dCache: ~65TB + 85TB of files cached temporarily
- Import of data back to FNAL through cdf-cat station, e.g.: files generated at Rutgers U ~9.3TB
- Organized skimming effort of a 13TB B physics dataset resulting in 9 datasets totaling ~4TB with the results stored at CNAF and at Fermilab

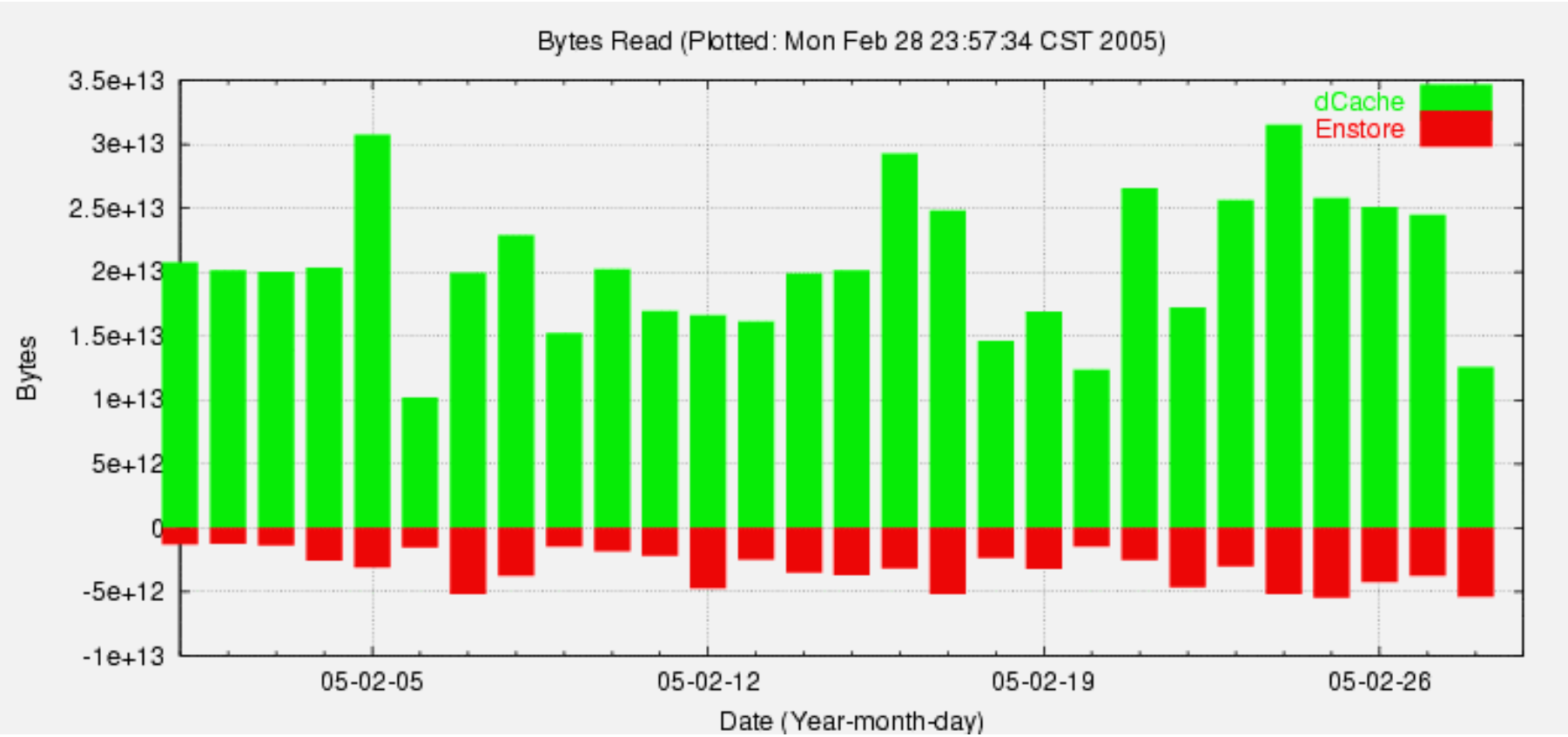
Last Year SAM Usage



Last Month SAM Usage at CDF



February 2005 dCache usage at CDF



Future Deployment Steps

- Retire Data File Catalog (DFC)
  - Fully implement direct SAM metadata entry for raw data (retirement of DFC to SAM tables metadata copying via “predator”) ~ 2 weeks
  - Provide for SAM storage of general user files (replace the “book” functionality) ~6 weeks
    - Includes hardware deployment, software infrastructure and documentation
  - Implement and/or document metadata browsing tools equivalent to the ones provided for the DFC ~4 weeks
  - Verify consistency of SAM and DFC tables ~2 weeks
  - Making DFC read only requires SAM to be deployed in production
    - Although there is an option to use SAM tables only and bypass explicit use of SAM

Future Deployment Steps cont'd

- Optimize usage on CAF type systems
  - Address potential problems related to NFS bottlenecks associated with SAM Python usage on CAF ~2 weeks
    - Using non Python SAM API (c++) in the next release of the CDF offline code practically eliminates the problem but requires additional testing
  - Mitigate issues related to SAM project start time by CAF and the actual start of the jobs
    - Together with/by the CAF Team - "imminent" ~(?) weeks

Future Deployment Steps cont'd

- Generic Deployment Items
  - Purchase and/or deploy proper hardware (servers)
    - Implement automated monitoring of hardware and software to satisfy CAF and remote usage needs ~8 weeks
  - Do full CAF scale load tests ~ a day in ~4 or 8 weeks (once dChache software upgrade is done)
  - Verify and update basic user documentation ~2 weeks
  - Verify and update administrator/shifter documentation ~2 weeks
  - Define support procedures ~1week
  - Deployment redlines should coincide with the readiness of the SAM based farm
    - Although the SAM-farm could write metadata to DFC tables as well
- Other:
  - Assess and mitigate if needed potential pre-staging inefficiency related to dCache usage as SAM cache
  - Deploy/use dCache raw data read and write pools to optimize SAM-farm I/O ~2 weeks; but after dCache software upgrade; together with/by "dCache Team"