Engineering, Operations & Technology
Information Technology

The Dynamic PBS Scheduler

Jim Glidewell
High Performance Computing
Enterprise Storage and Servers
May 8, 2008

Computing Environment

- **Cray X1**
  - **2 Chassis, 128 MSPs, 1TB memory**

- **Linux Clusters**
  - **Over 1700 compute nodes**

- **Panasas Storage**
  - **Primary storage for cluster, secondary for Cray X1**

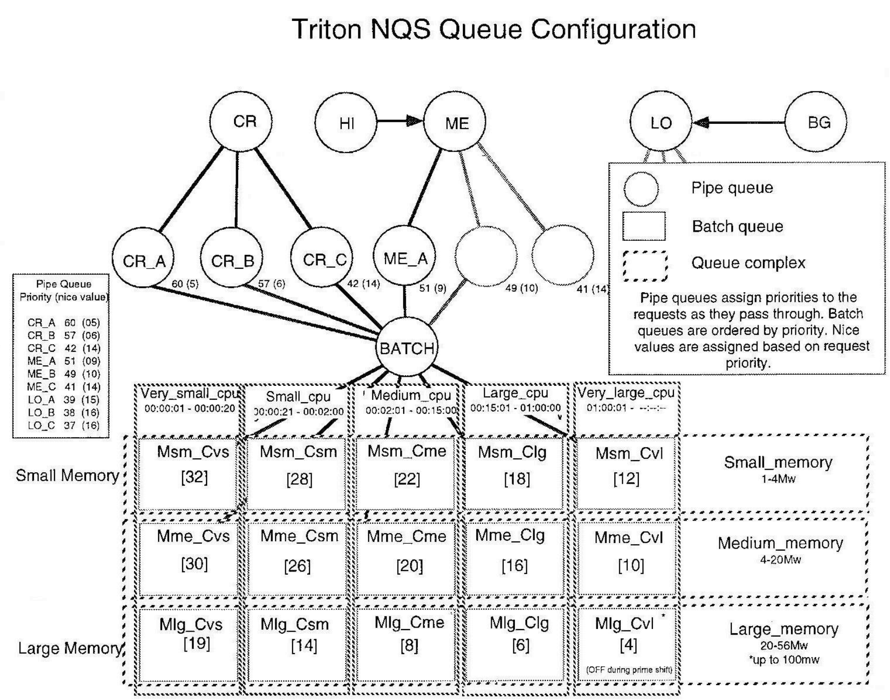
Why use a Dynamic PBS Scheduler?

- **Didn't want to write a complete scheduler from scratch**
- **Default PBS scheduler has a lot of functionality**
  - **Ordering by priority**
  - **User limits, queue limits, server limits**
  - **Starving jobs logic**
  - **Interaction with Cray's psched**
- **Multiple queues are useful…**
  - **Separate queues allow finer control of usage**
  - **Primary method for separating different "kinds" of jobs**
    - **Short versus long**
    - **Varying memory or CPU requirements**
- **…but problematic**
  - **User turnaround can suffer when a restricted queue gets swamped**
  - **Fixed queue limits can lead to underutilization…**
- **Many tuning options are not available with a single queue**
- **We needed a mechanism to balance queue limits based on a changing workload**

History

Engineering, Operations & Technology | **Information Technology**

- **First dynamic scheduler at Boeing was written in "C" and targeted to controlling NQS on our Cray T90's**
- **A more limited dynamic scheduler was used on our SGI Origin 3800, written in Perl**
- **Current dynamic PBS scheduler (dyn_pbs) is written in Perl**
- **dyn_pbs is conceptually and functionally similar to T90 version**



Goals

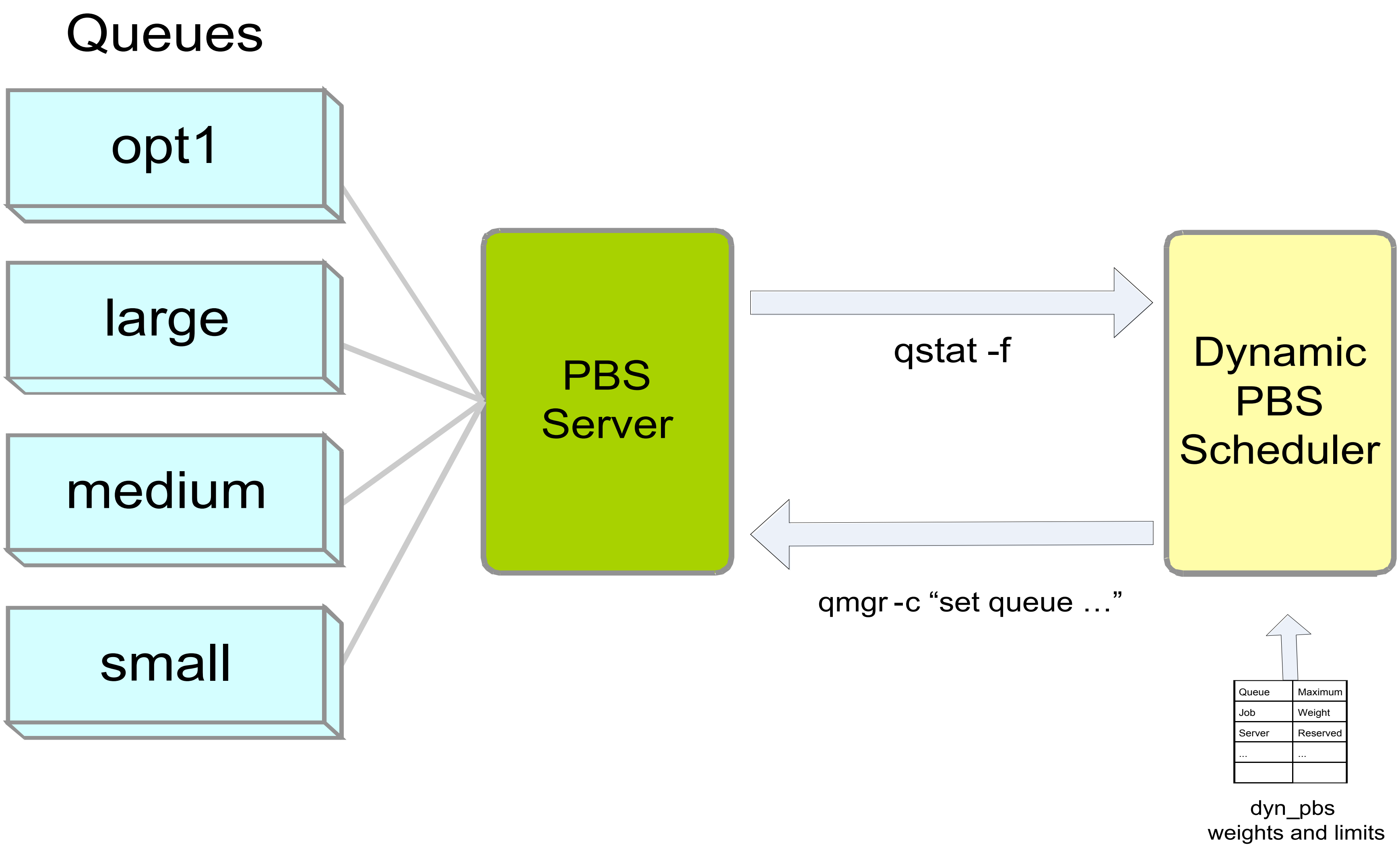
- **Functionality**
  - **Provide reasonable turnaround for all job classes**
  - **Honor user priorities**
  - **Minimize whitespace**
  - **Meet broad throughput targets**
- **Simplicity**
  - **Modest development effort**
  - **Easy to adjust and tune**
- **Safety**
  - **"Safe"**
    - **Does the minimum necessary**
    - **Can be run as an ordinary user (with PBS operator privileges)**
    - **Won't put the system in a bad state**
  - **"Fail-safe"**
    - **Won't leave the system unusable in case of daemon failure**
    - **Simple auto-restart via cron**

Workload Characteristics

- **Mix of very large and very small jobs**
  - **Large jobs are long-running and consume significant percentage of MSPs (10%-20% each)**
  - **Smallest jobs are very short running and require a single MSP or SSP**
  - **Small jobs very sensitive to turnaround**
- **Job Types:**
  - **Optimization jobs using Overflow**
    - **Usually 16 MSPs, multiple cycles**
  - **Individual Overflow analysis case**
    - **8-16 MSPs, single cases**
  - **TRANAIR analysis cases**
    - **Single MSP, usually multiple runs**
  - **ATLAS cases**
    - **Single MSP or SSP, short running**

Queue Structure and dyn_pbs Daemon

Engineering, Operations & Technology | **Information Technology**



Overview of dyn_pbs Weights and Limits

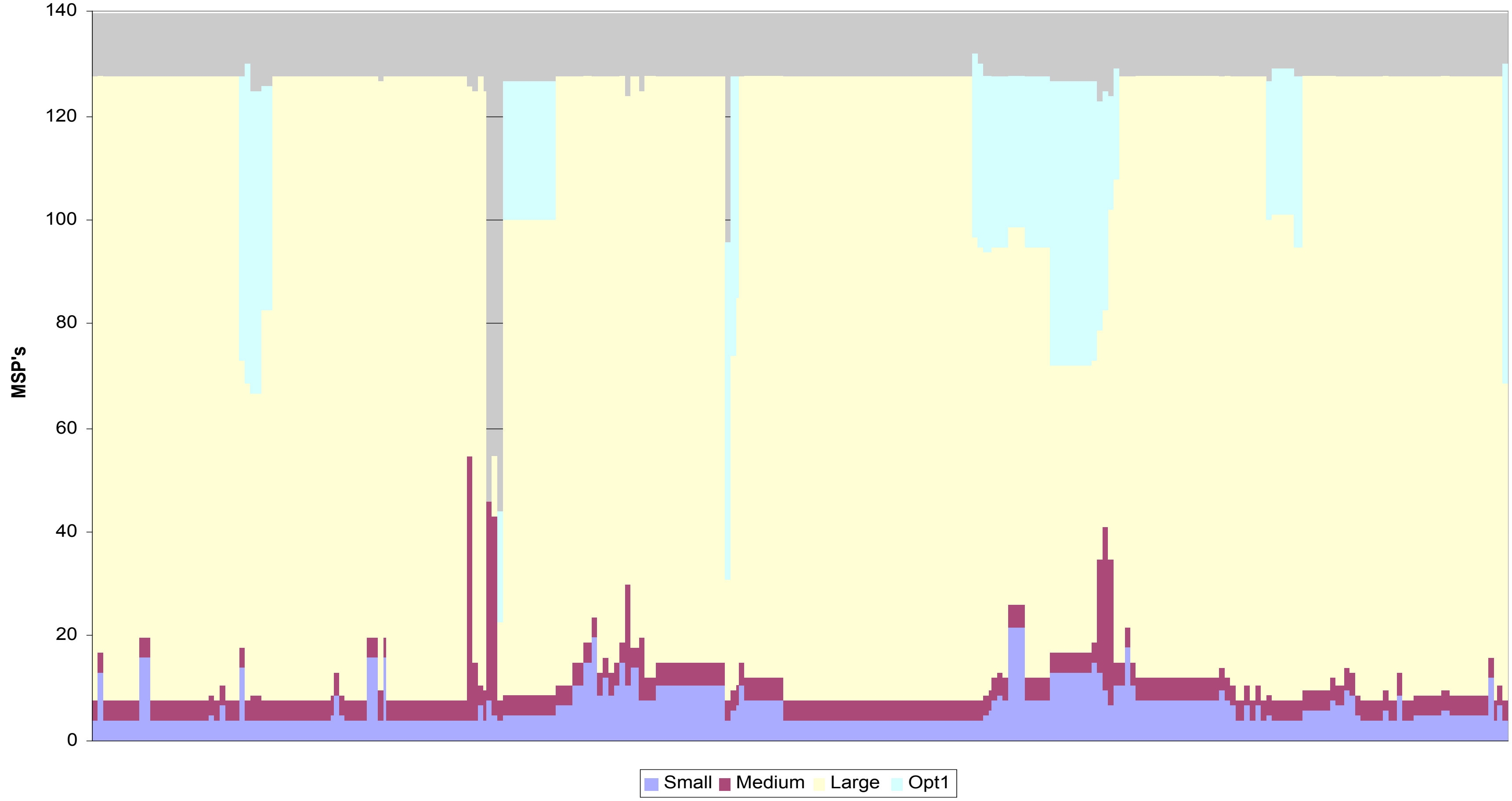
- **Server-related weights and limits**
  - **Server oversubscription factor**
  - **Reserved MSPs**
- **Queue-related weights and limits**
  - **Minimum, default, maximum number of MSPs**
  - **Queue weighting factor (for queued jobs)**
  - **Queue additional MSP weight per running job**
  - **Queue maximum weight**
  - **Queue oversubscription factor**
  - **Queue oversubscription order**
- **Job-priority-related weights**
  - **Priority-based MSP weighting factor**
  - **Priority-based job weight**

Overview of dyn_pbs Process

- Determine # of MSPs
- Gather data from PBS (qstat -f)
- Compute each job’s weight
- Compute initial queue MSP distribution
- Reclaim MSPs from any queues above their maximum
- Allocate reclaimed MSPs to other queues
  - Proportional to their computed limit
- Deal with oversubscribed queues
  - Subtract MSPs from all queues, based on queue oversubscription factors
- Issue “qmgr” directives to adjust queue’s MSP limits
  - Only for queues who’s limits have changed
- Delay, then repeat…

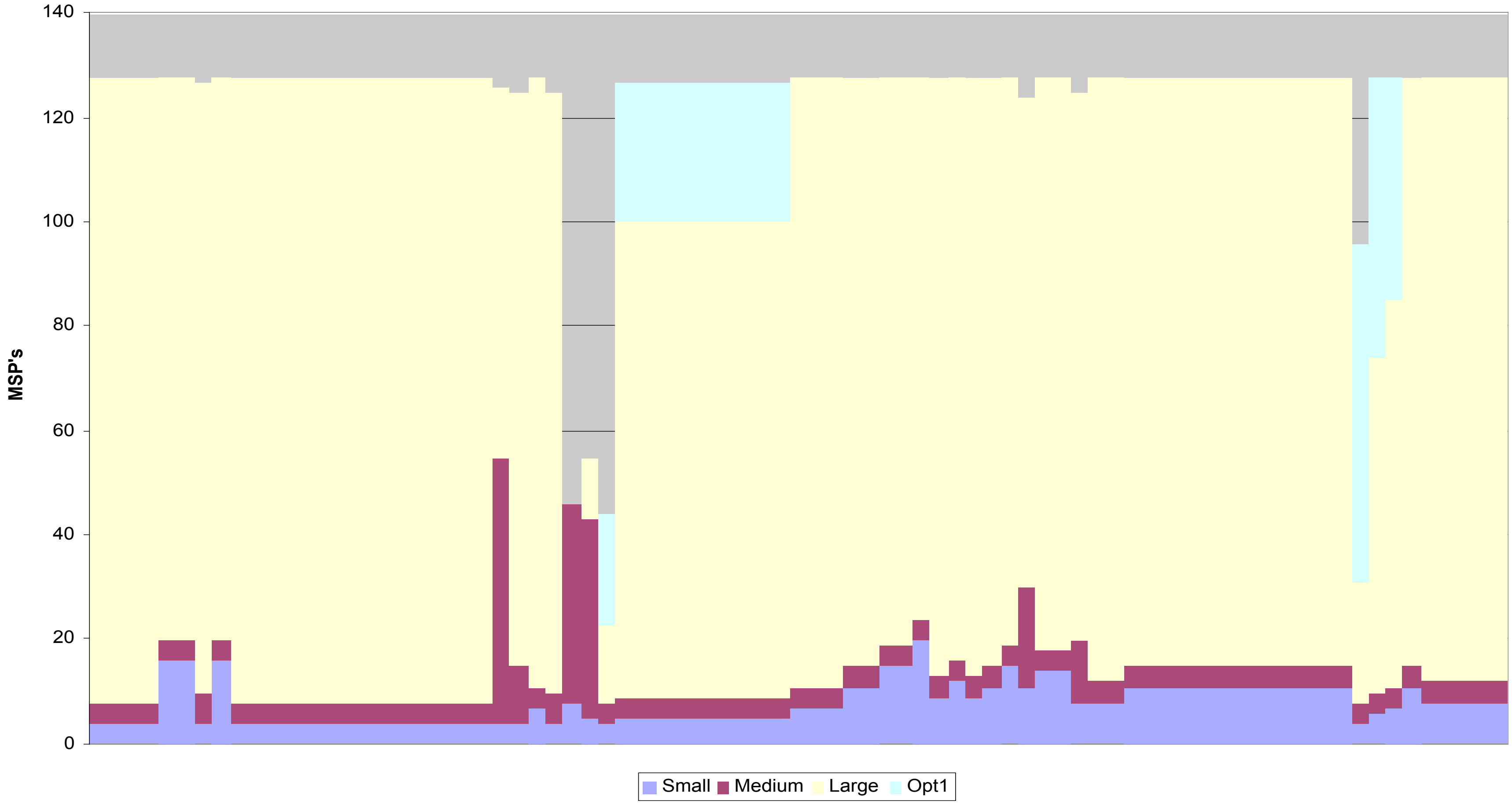
Dynamic Queue Limit Adjustments

Engineering, Operations & Technology | **Information Technology**



Dynamic Queue Limit Adjustments - Detail

Engineering, Operations & Technology | **Information Technology**



Results

Engineering, Operations & Technology | **Information Technology**

- **"No phone calls…"**

Understanding Turnaround

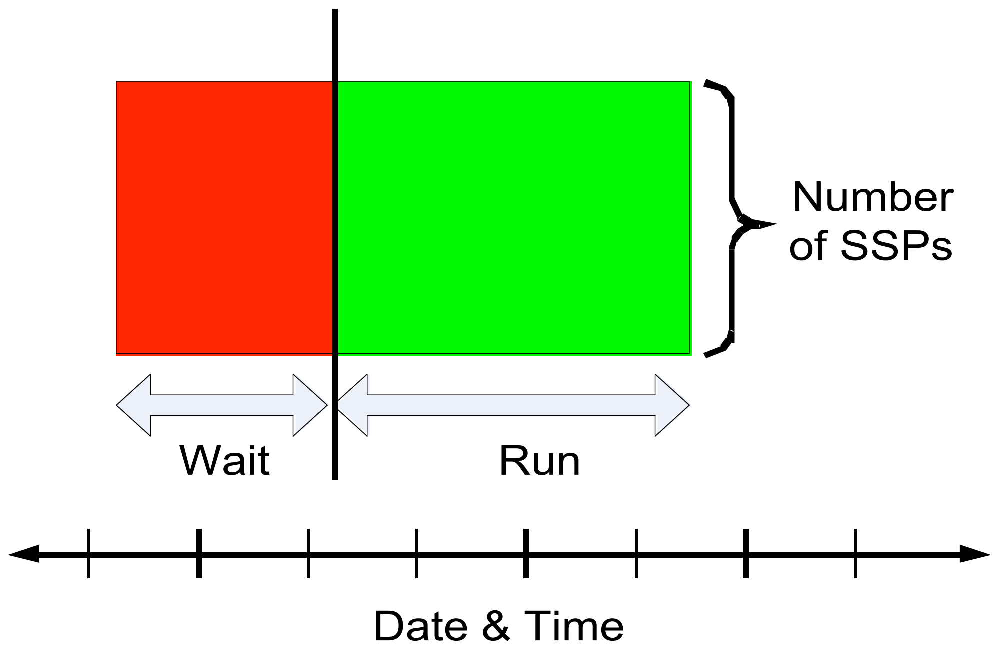
Engineering, Operations & Technology | **Information Technology**

- **The two primary goals of dyn_pbs were maximizing utilization and minimizing turnaround time**
  - **Measuring overall system utilization is easy**
  - **Turnaround measurements are more problematic**
- **Tried several metrics**
  - **Average wait time, average ratio of wait to run times**
  - **Percentiles for wait time or ratios**
  - **Same metrics bucketed or weighted by job size**
- **No single summary metric really told us what was going on**
- **In measuring turnaround, outliers are often important**
  - **But summary statistics couldn't answer *why* these jobs were outliers**
- **"Wouldn't it be nice" if we could get a birds-eye view of the details…**

The "Red/Green" Turnaround Charts

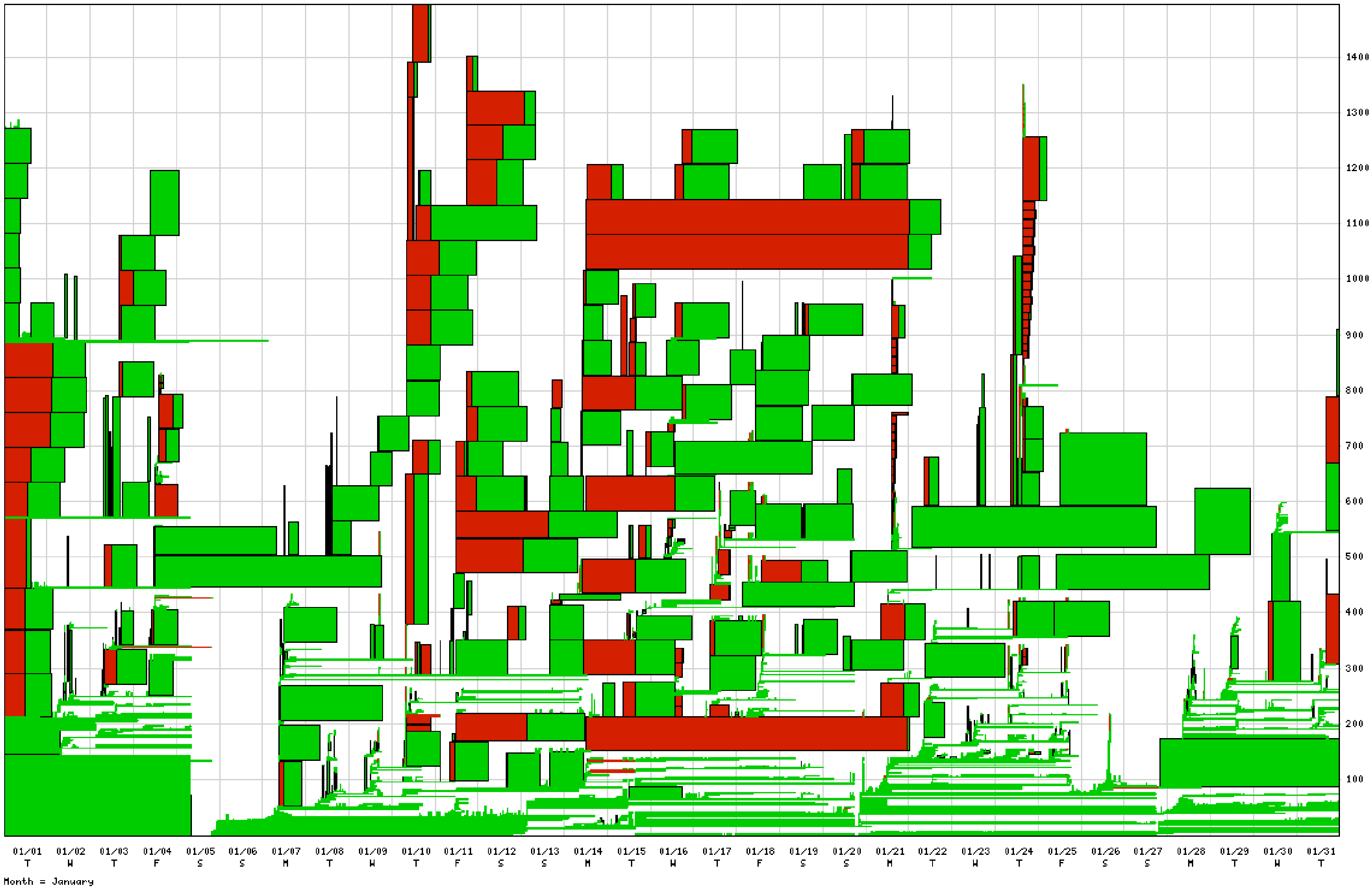
Engineering, Operations & Technology | **Information Technology**

- **Each job is represented by a two-color rectangle**
  - **Red indicates wait time (queued)**
  - **Green indicates execution time**
- **Height of rectangles represents number of SSPs (or nodes)**
- **Horizontal axis is a timeline**
- **Vertical position of jobs is not significant (first-fit)**



Turnaround - Wait versus Run

Engineering, Operations & Technology | **Information Technology**



Summary and Conclusions

Engineering, Operations & Technology | **Information Technology**

- **Varying workloads are common on many systems**
- **The default PBS scheduler has a large number of useful features, but does not adjust limits dynamically**
- **Multiple queues with fixed limits can result in**
  - **Failure to meet turnaround expectations**
  - **Increased whitespace (idle capacity)**
- **Adding a modest dynamic component to a fixed queue structure can improve utilization and still meet user expectations**
- **The dynamic PBS scheduler has helped us better utilize our Cray X1**