


# “Blue Protein” system at CBRC, AIST - System Management -



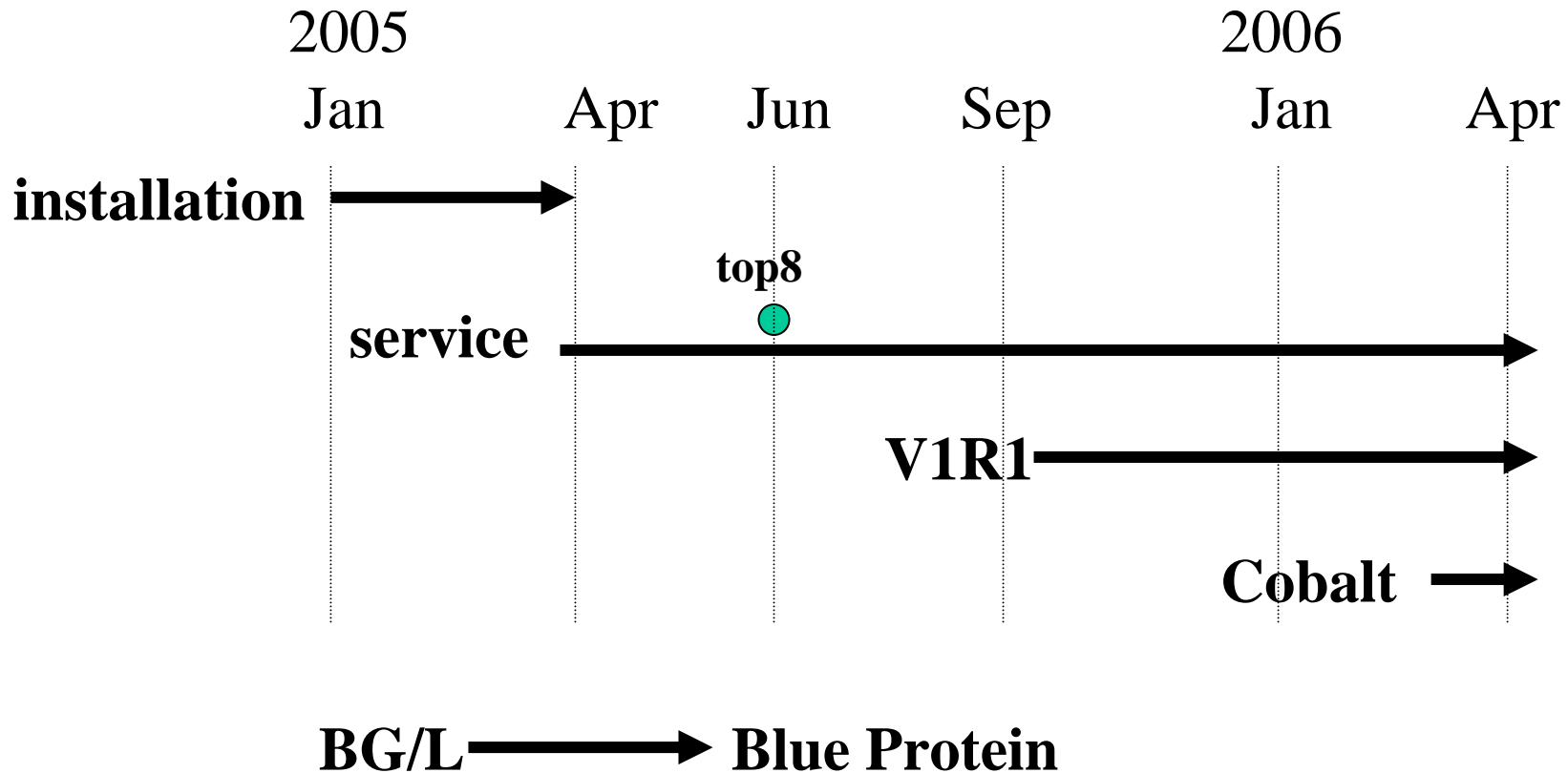
Yasuyuki Yamaguchi / System Engineer  
Computational Biology Research Center, AIST 

# Topics

- **Overview**
- Customizations
- Filesystem
- Software upgrade / maintenance
- Porting Cobalt

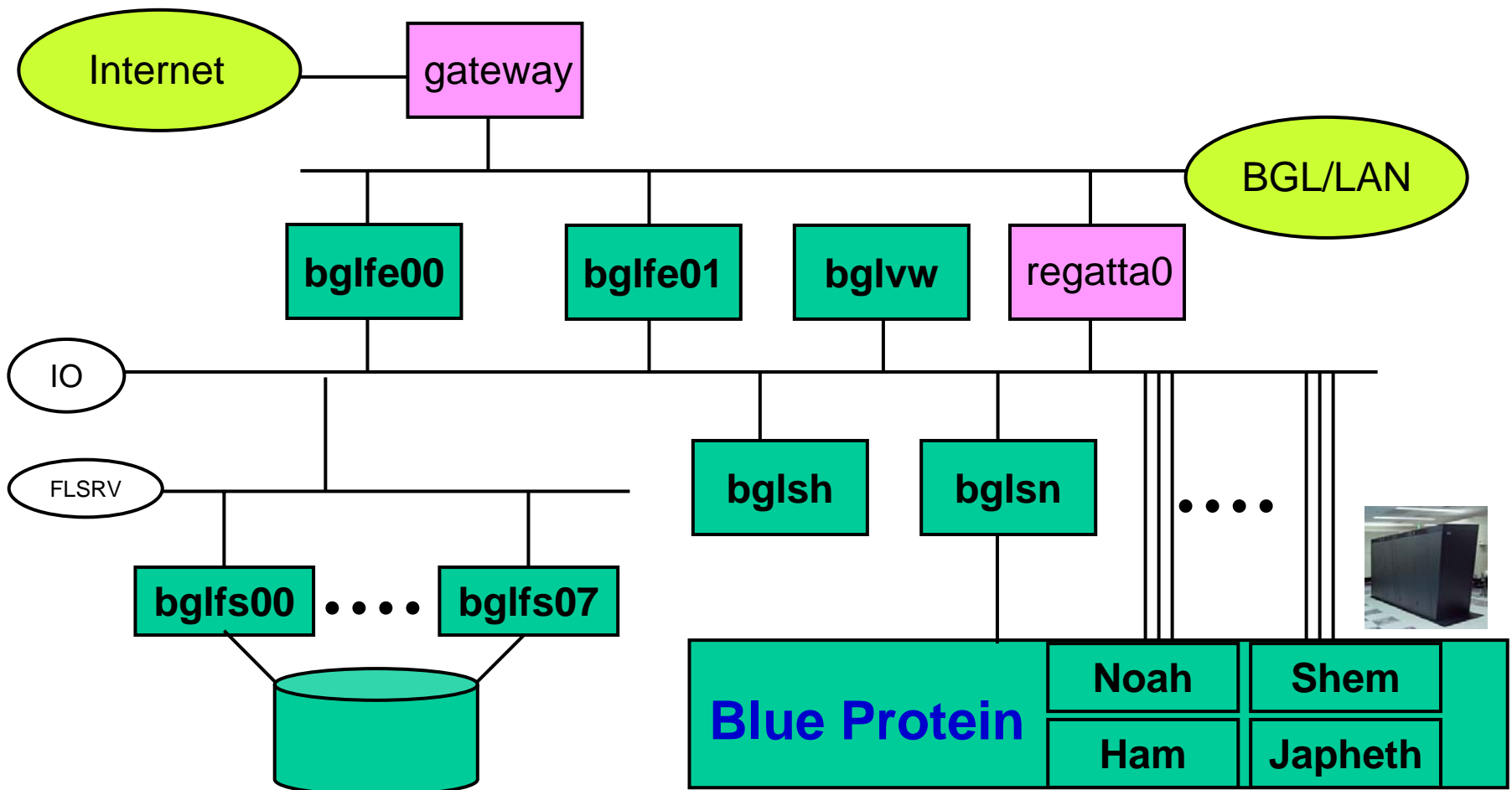
# Blue Protein history

- **history**



# System Overview

- “Blue Protein” system at CBRC

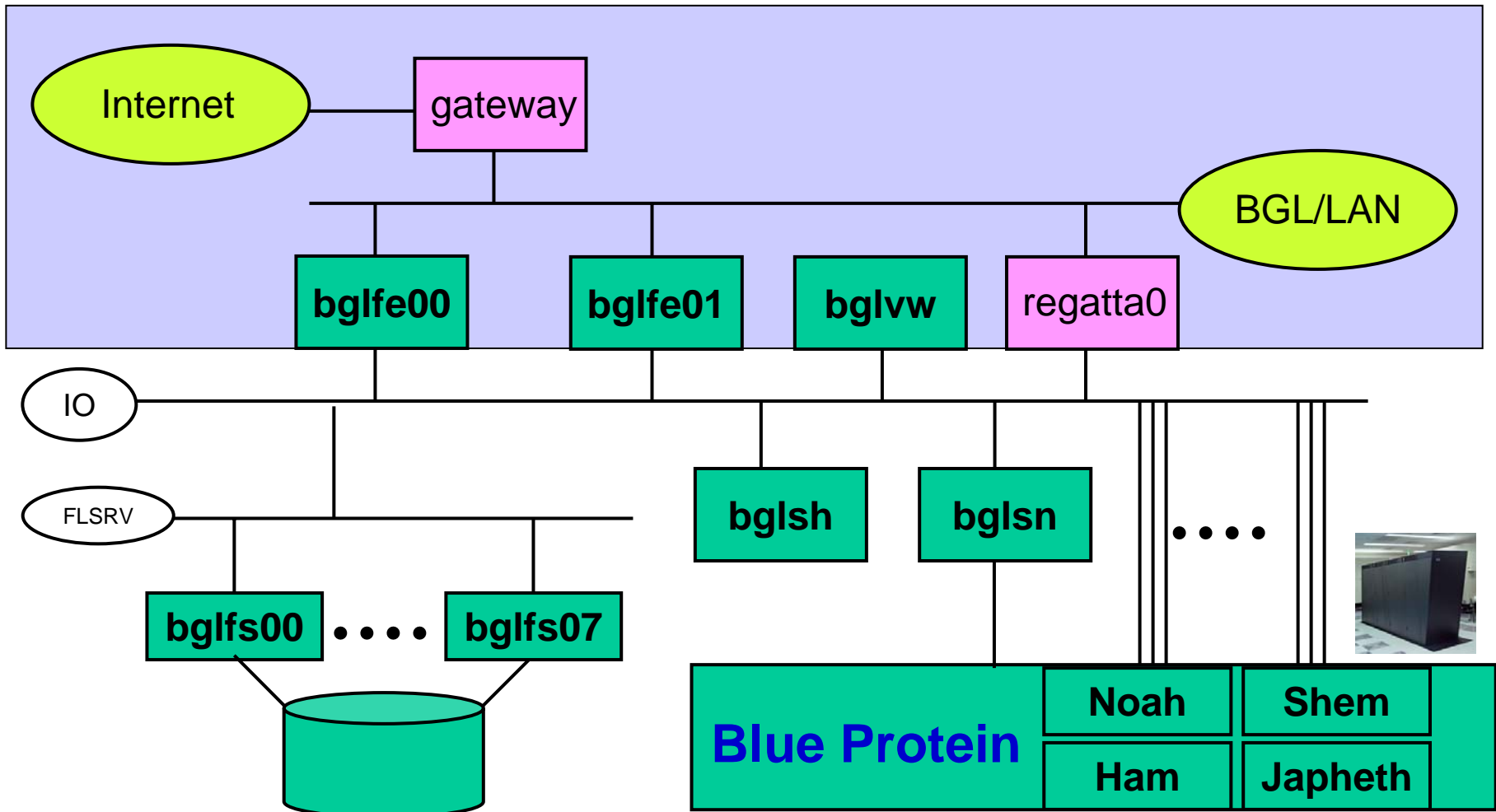


# “Blue Protein” Overview

- “Blue Protein” (Blue Gene) 4 racks
- Dual Front-End nodes [fe]
- Service node [sn]
- Eight File Servers with GPFS [fs]
- Science Host node [sh]

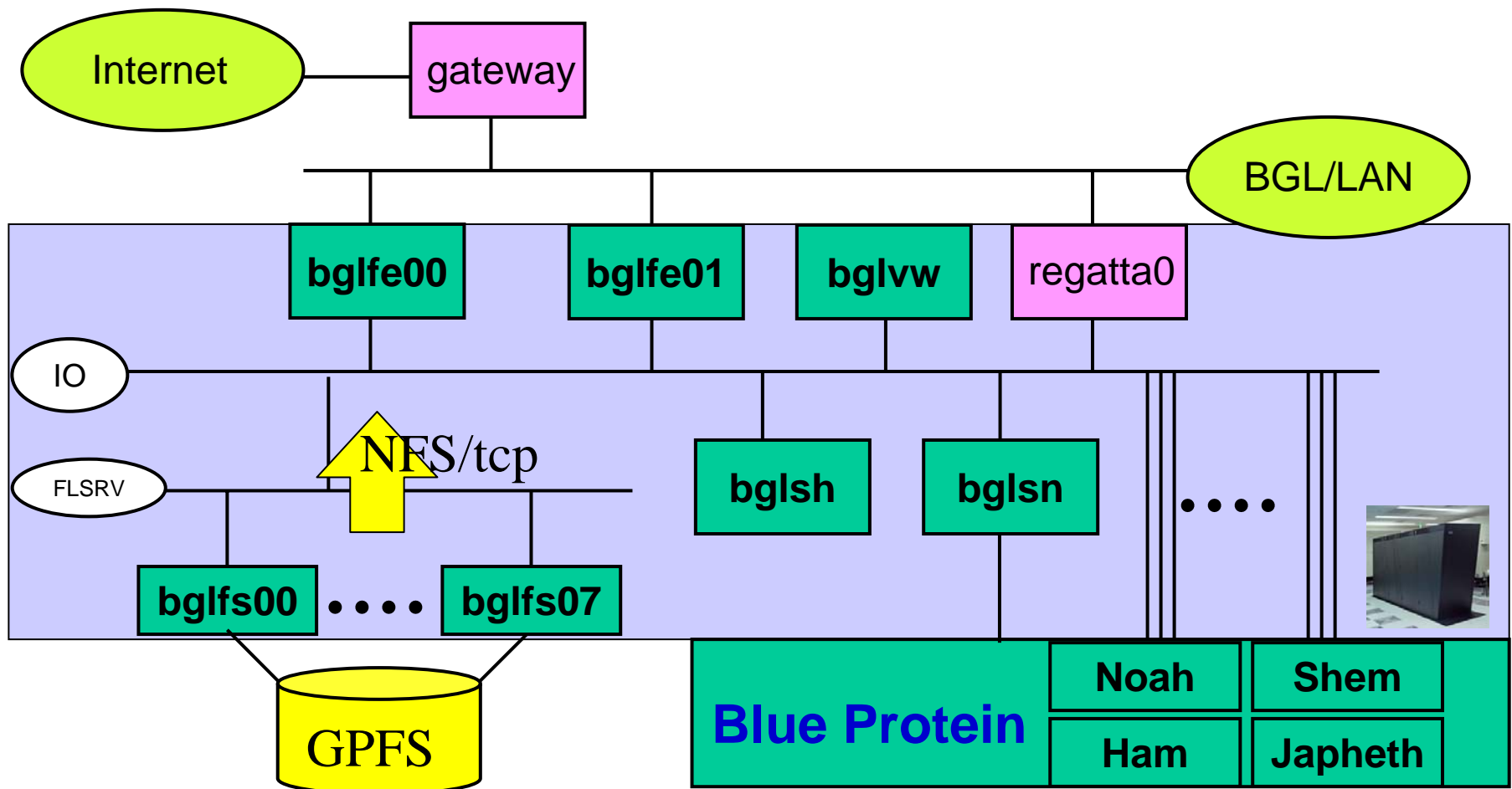
# Network topology (1)

- Internet and user access



# Network topology (2)

- File service and control network



# Topics

- Overview
- **Customizations**
- Filesystem
- Software upgrade / maintenance
- Porting Cobalt

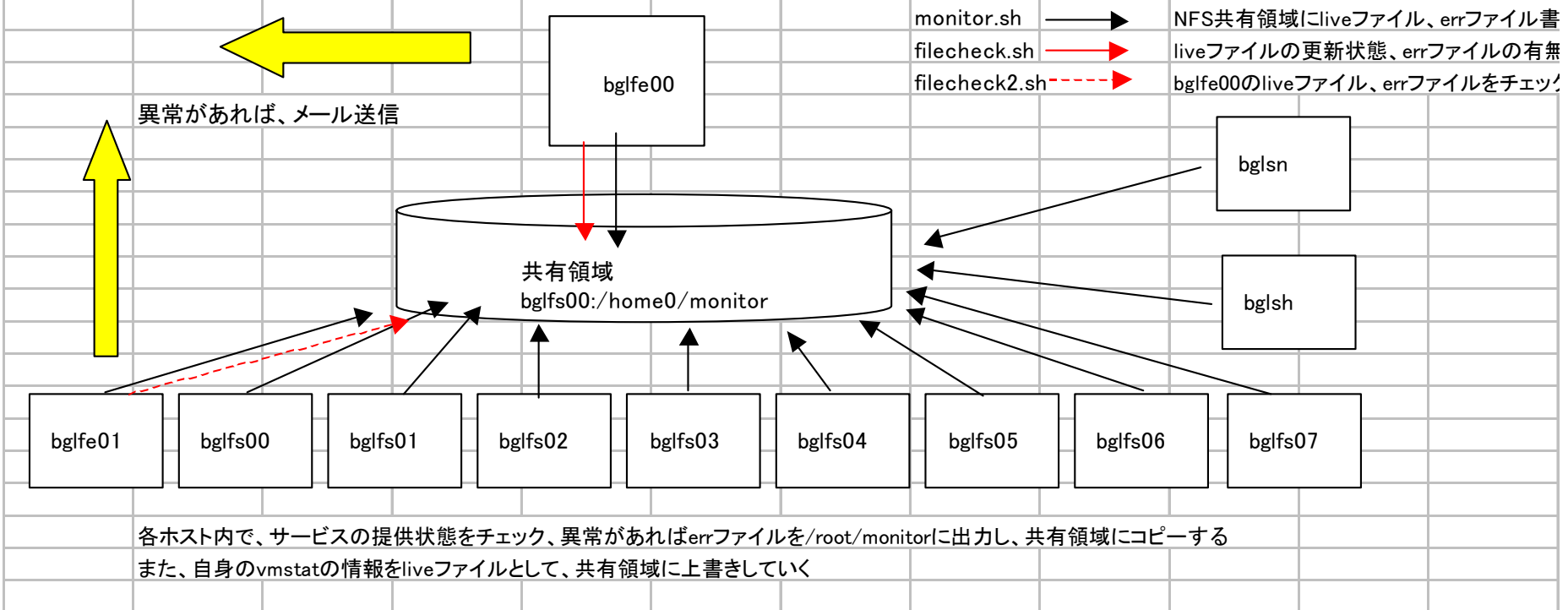


# Failure monitoring

- **We are monitoring the following subjects:**
  - **System down**
    - “Blue Protein”
    - other servers
  - **Service availability**
    - Network,GPFS,DB2,Disk,RAID,Apache

# Monitoring flow

概要



All nodes write their “alive” files on shared disk space (NFS).

Two front-end nodes monitor each other.

If there is any failure, a front-end sends e-mail to the specified address.

# Monitoring script (example)

- Below is a script to find any fatal errors from DB2 RAS event log.
- We can also check it on 'bglweb', however it's necessary to automatically find them.

```
#!/bin/sh
```

```
./bgl/BlueLight/ppcfloor/bglsys/bin/db2profile
```

```
db2 connect to xxxxx user xxxxxx using xxxxx
```

```
db2 "select * from tbgleventlog where Severity='FATAL'" |awk  
    '{print $1,$2,$4,$5,$6,$7,$10,$11,$12,$13,$14,$15}' |grep -v  
APP
```

```
db2 terminate
```

- Sorry, there are some information we cannot disclose fully...
- We frequently compare output from this script with old one, and they should be same unless new fatal error occurs.

# Manual scheduling policy

- **A single rack per research group (1024 nodes)**
  - It's our default, but sometimes flexibly changed.
  - Now 90% usage is for protein MD by AMBER8.
  - Others are for porting applications and libraries.
- **No queuing, but interactive submissions**
  - using mpirun.

# Supporting job submission

- **To run jobs interactively, a user wants to know**
  - **Where he/she can submit.**
  - **Which nodes are vacant, accessible, or occupied**
  - **with intuitive user interface**
- **Our ‘Bglweb’ provides each block status information.**

# Web-based status monitor (1)

- All **block (= 32 nodes)** status.
  - running a job
  - allocated
  - free
  - under maintenance
- Status updates every **5 minutes** by cron.
  - block information
  - status provided by DB2



# Web-based status monitor (2)

- **Displaying mode**
  - show all racks
  - show one rack
- **Running blocks**
  - username
  - job id
- **Allocated blocks**
  - username
- Each block has a link for detailed information provided by 'bglweb'.

Blue Gene Node Status - Mozilla Firefox

ファイル(E) 編集(E) 表示(V) 移動(Q) ブックマーク(B) ツール(T) ヘルプ(H)

http://10.128.128.11/bglweb/cbrc2.cgi

Blue Gene Node Status

--- Job Running  
--- Allocate  
--- Free  
--- In Maintenance

Update time Mon Sep 5 16:15:06 2005

All Rack0 Rack1 Rack2 Rack3

n0032m0n0 hourai	n0032m0n1	n0032m0n2	n0032m0n3	n0032m0n4 18598 masakazu	n0032m0n5 18661 masakazu	n0032m0n6 18658 masakazu	n0032m0n7
n0032m0n8 18620 masakazu	n0032m0n9 18609 masakazu	n0032m0nA 18668 masakazu	n0032m0nB 18631 masakazu	n0032m0nC 18646 masakazu	n0032m0nD 18665 masakazu	n0032m0nE 18610 masakazu	n0032m0nF 18611 masakazu
n0032m1n0 18570 masakazu	n0032m1n1 18569 masakazu	n0032m1n2 18633 masakazu	n0032m1n3 18637 masakazu	n0032m1n4 18586 masakazu	n0032m1n5 18588 masakazu	n0032m1n6 18594 masakazu	n0032m1n7 18571 masakazu
n0032m1n8 18596 masakazu	n0032m1n9 18605 masakazu	n0032m1nA 18604 masakazu	n0032m1nB 18641 masakazu	n0032m1nC 18582 masakazu	n0032m1nD 18578 masakazu	n0032m1nE 18643 masakazu	n0032m1nF 18653 masakazu

# Accounting

- **Simple accounting report from system logs.**
- **Accounting information**
  - **When**
  - **Where (How big)**
  - **Who**
  - **How long**
- **Source**
  - **/bgl/BlueLight/logs/BGL/\*-ciodb-\***
  - **DB2 bgljob\_history**



# Accounting results

- job-id,username,block,start date,end date

*17493,userB,n0032m0n4,Aug 31 17:23:39,Sep 1 00:59:44*

*17486,userB,n0032m4nC,Aug 31 12:56:58,Sep 1 02:47:55*

*17511,userB,n0032m0n5,Aug 31 18:36:16,Sep 1 02:50:50*

*17487,userB,n0032m5n8,Aug 31 13:30:56,Sep 1 03:23:20*

*17488,userB,n0032m5nA,Aug 31 14:37:13,Sep 1 04:31:30*

*17490,userB,n0032m7n8,Aug 31 17:12:31,Sep 1 05:23:24*

*17491,userB,n0032m7n1,Aug 31 17:15:27,Sep 1 05:28:07*

*17492,userB,n0032m7nC,Aug 31 17:20:57,Sep 1 05:31:03*

*17494,userB,n0032m6nC,Aug 31 17:27:10,Sep 1 05:39:01*

*17495,userB,n0032m6nD,Aug 31 17:29:02,Sep 1 05:40:23*

*17496,userB,n0032m7n7,Aug 31 17:31:23,Sep 1 05:42:03*

*17497,userB,n0032m6nF,Aug 31 17:34:18,Sep 1 05:45:33*

*17498,userB,n0032m6nE,Aug 31 17:37:00,Sep 1 05:48:09*

# Monthly report

- Example (August 2005) report is shown below.
  - **‘block elapse’**: elapse hours multiplied by block numbers.  
(our local definition: 1 block = 32 nodes)
  - ex.) 2 hr. 20 min. on 512 nodes = 37.4 block hours

user	node size	number of job	block elapse	elapse(hh:mm:ss)
userA	32	339	22.1	22:06:55
userC	32	28	0.1	0:03:32
userB	32	6512	60713.4	60713:26:22
userA	128	2	0.0	0:00:19
userA	512	237	37.4	2:20:12
userC	512	1	0.0	0:00:06
userC	1024	4	0.2	0:00:20
userC	4096	3	0.7	0:00:20
合計		7126	60773.9	

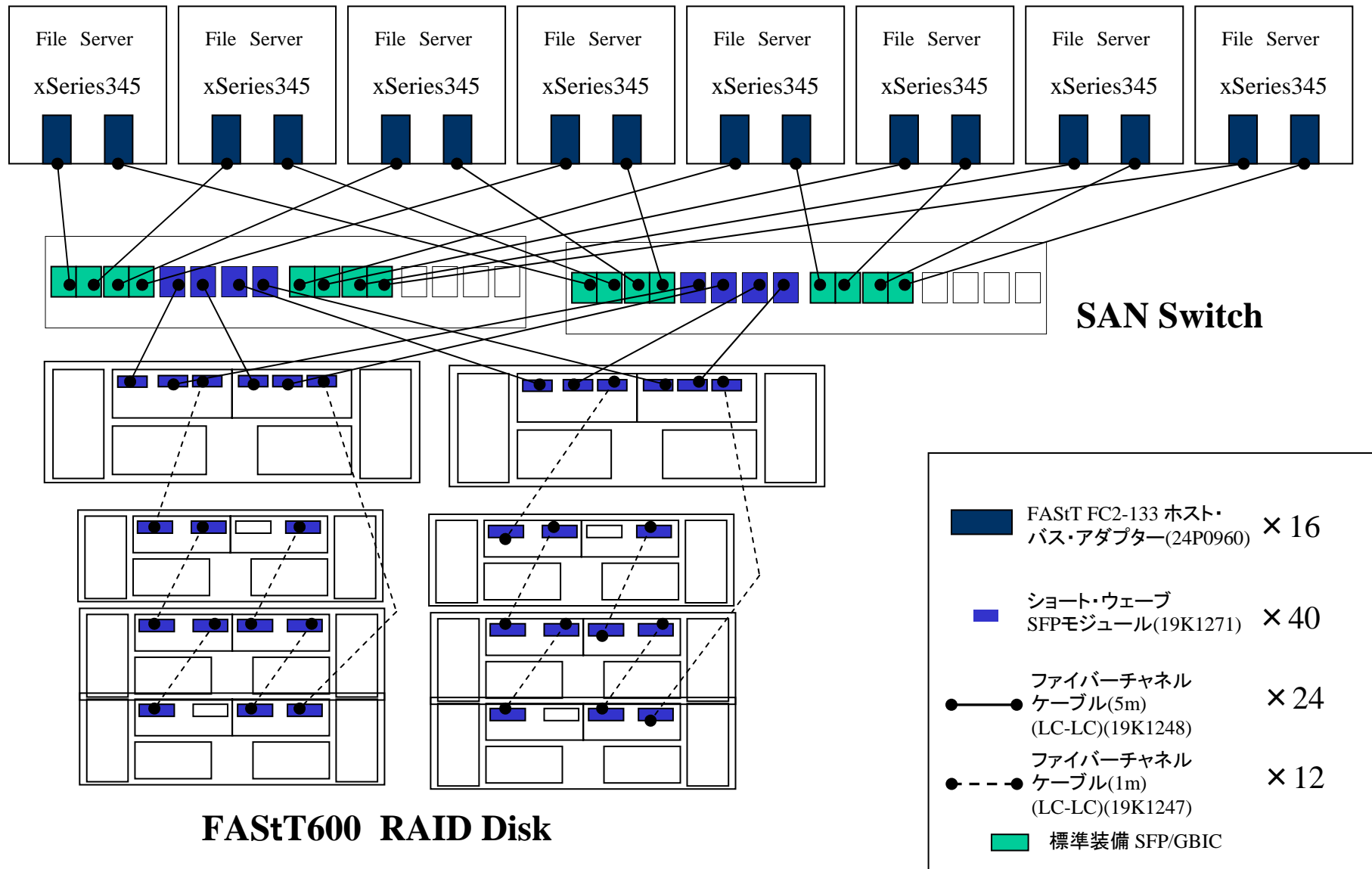
# Topics

- Overview
- Customizations
- **Filesystem**
- Software upgrade / maintenance
- Porting Cobalt

# Filesystem construction

- **system**
  - /bgl
  - service node
- **user working directory**
  - /bgfs00 ~ /bgfs07
  - GPFS shared by file servers
  - 2TB per directory
- **features of data**
  - random access (read / write )
  - not sequential, not big size

# Fibre channel connection



# Performance data

	IOZONE	diskeater	Appli like
	single thread random read/write Size: 4GB BSZ: 256kB  MB/s	parallel write  upper: 8 parallel lower 16 parallel size: 4GB sec	overwrite 1.5MB files from all nodes, and appending 0.75MB files.  32 parallel size: 6.4GB sec
FS#1:tstfs001	Seq. 126.6 Rand. 70.2	Min 31.0 Max 38.9 Min 0.5 Max 38.2	Min 137 Max 142
FS#2:tstfs002	Seq. 222.3 Rand. 98.6	Min 23.6 Max 38.1 Min 0.5 Max 36.8	Min 125 Max 130
FS#3:tstfs003	Seq. 128.3 Rand. 69.5	Min 29.1 Max 39.8 Min 0.5 Max 37.1	Min 146 Max 155
FS#4:tstfs004	Seq. 133.1 Rand. 89.6	Min 33.0 Max 47.3 Min 0.5 Max 37.6	Min 124 Max 130
FS#5:tstfs005	Seq. 111.6 Rand. 71.9	Min 23.9 Max 43.7 Min 0.5 Max 40.6	Min 128 Max 141
FS#6:tstfs006	Seq. 121.3 Rand. 90.4	Min 24.1 Max 38.3 Min 17.1 Max 36.8	Min 120 Max 127

# Topics

- Overview
- Customizations
- Filesystem
- **Software upgrade / maintenance**
- Porting Cobalt

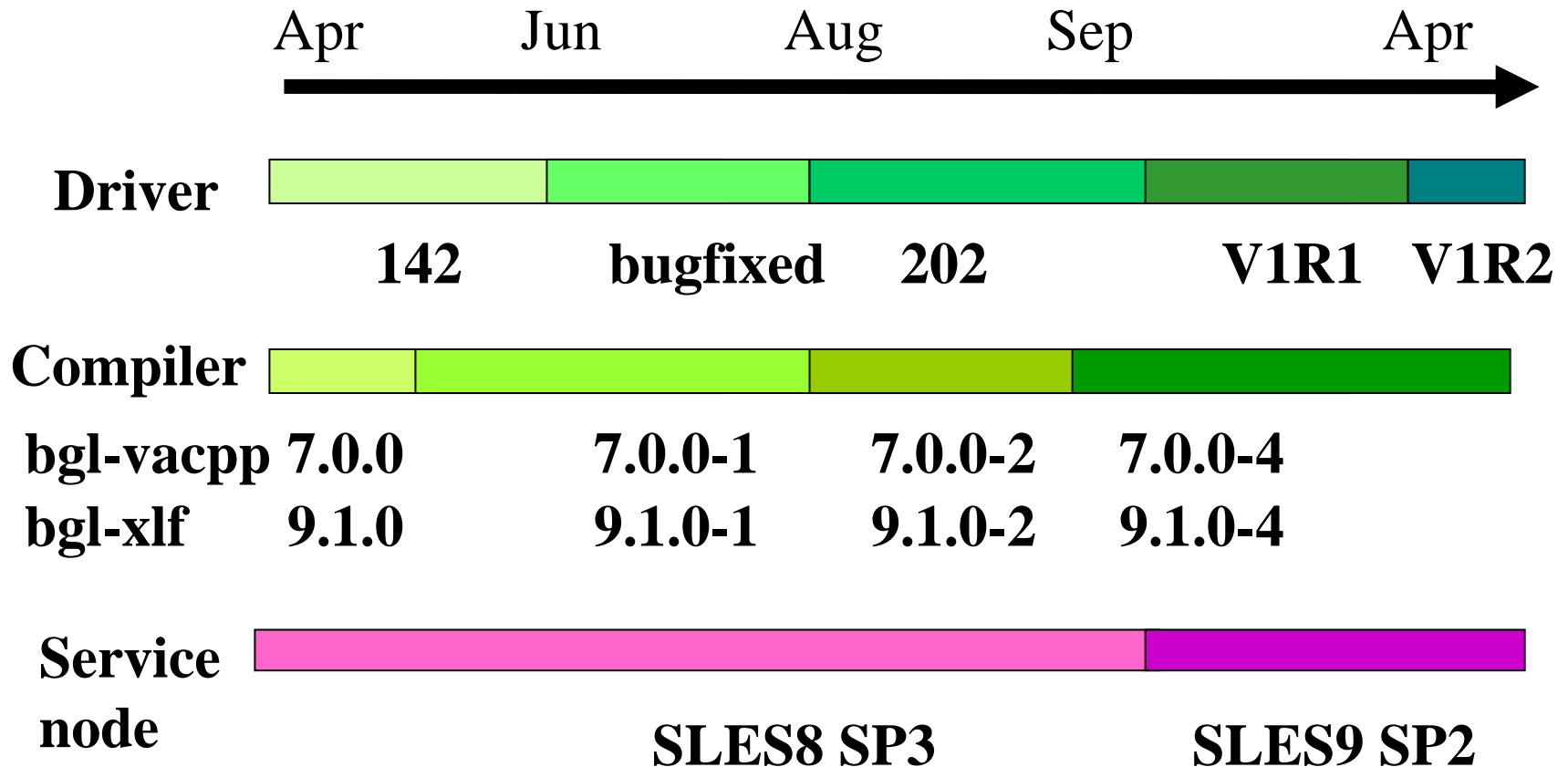
# Software update/maintenance

- **Driver upgrades**
  - 142 bugfixed
  - 202
  - V1R1
  - V1R2
- **Compiler/Library upgrades**
  - bug fix (mainly for Double Hummer option)
- **Service Node OS/package upgrades**
  - SLES9



# Software update/maintenance

- upgrades history



# V1R2

- **We upgraded to V1R2 on 11<sup>th</sup>, April**
- **Improvement**
  - **more detailed job information**
    - **elapse is available on BG/L web.**
    - **job start time is also available on mmcs\_db\_console.**

# System failure (since Apr. 2005)

- **Hardware(14)**
  - compute card (13)
  - **service card**
- **Software(3)**
  - bglmaster
  - ciodb
  - compiler
- **other nodes(2)**
- **MTTI**
  - 21 days (365 days / 17 times)

# Topics

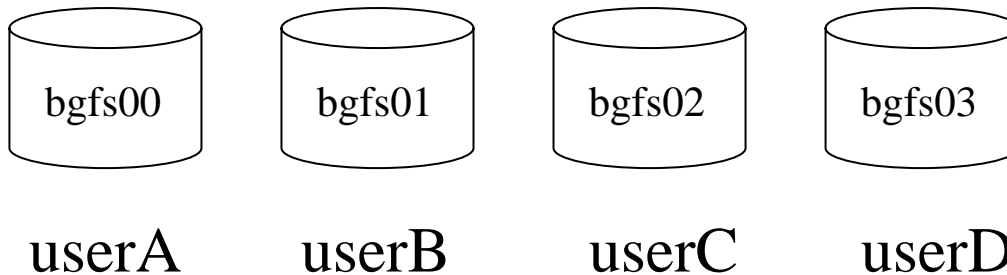
- Overview
- Customizations
- Filesystem
- Software upgrade / maintenance
- **Porting Cobalt**

# Porting Cobalt

- **target**
  - **efficient job control**
  - **easy submitting for users**
- **installation**
  - **Easy installation**
- **customizing**
  - **pre-command**

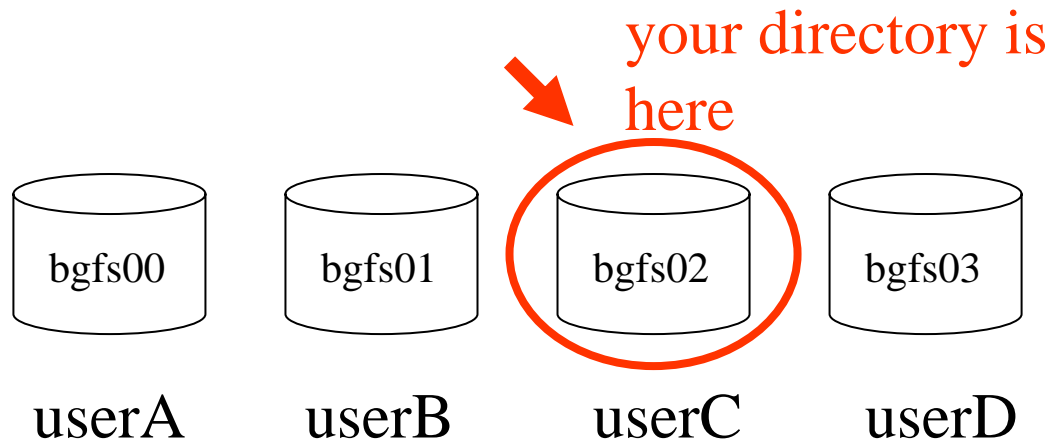
# Why Customizing?

- **Blue Protein file system policy**
  - separating working filesystems
  - that is to distribute I/O load.
  - I/O node mounts only a filesystem that is necessary for a job.



# Why Customizing?

- **Selecting filesystem**
  - Before you allocate any block, you must define which filesystem will be used.



# Why Customizing?

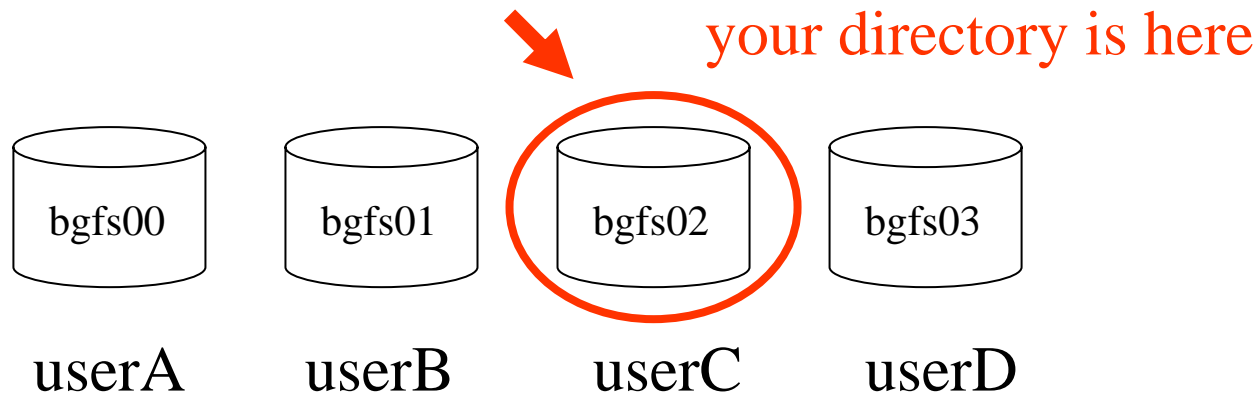
- **Defining filesystem**

- ex) In case you will run a job on n0032m0n0 ( that locates midplane0).

- you must define the following lines in /bgl/dist/var/bglfs/dirs/n0032m0n0

- export BGL\_USER\_EXPORTDIR=/exports/bgfs02

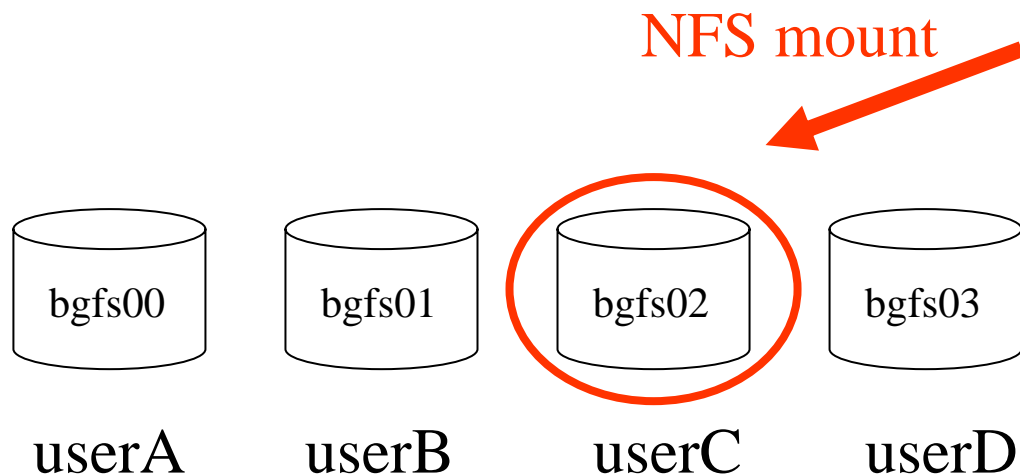
- export BGL\_USER\_MOUNTDIR=/bgfs02





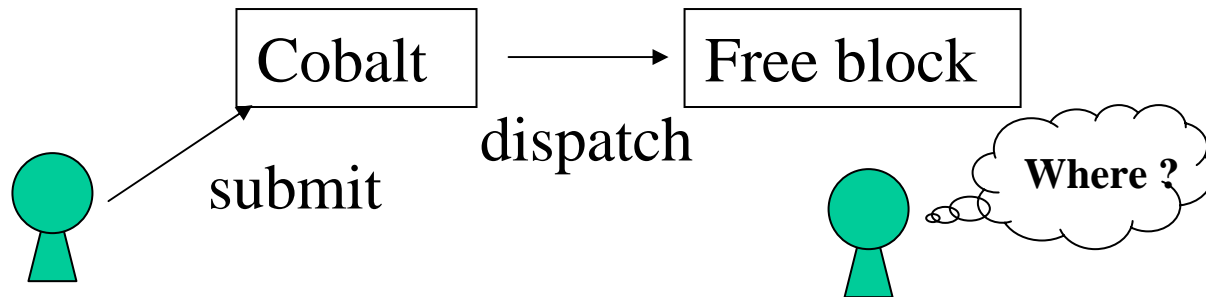
# Why Customizing?

- **Mounting filesystem**
  - I/O node of n0032m0n0 mounts /bgfs02 when it boots



# Problem on Cobalt

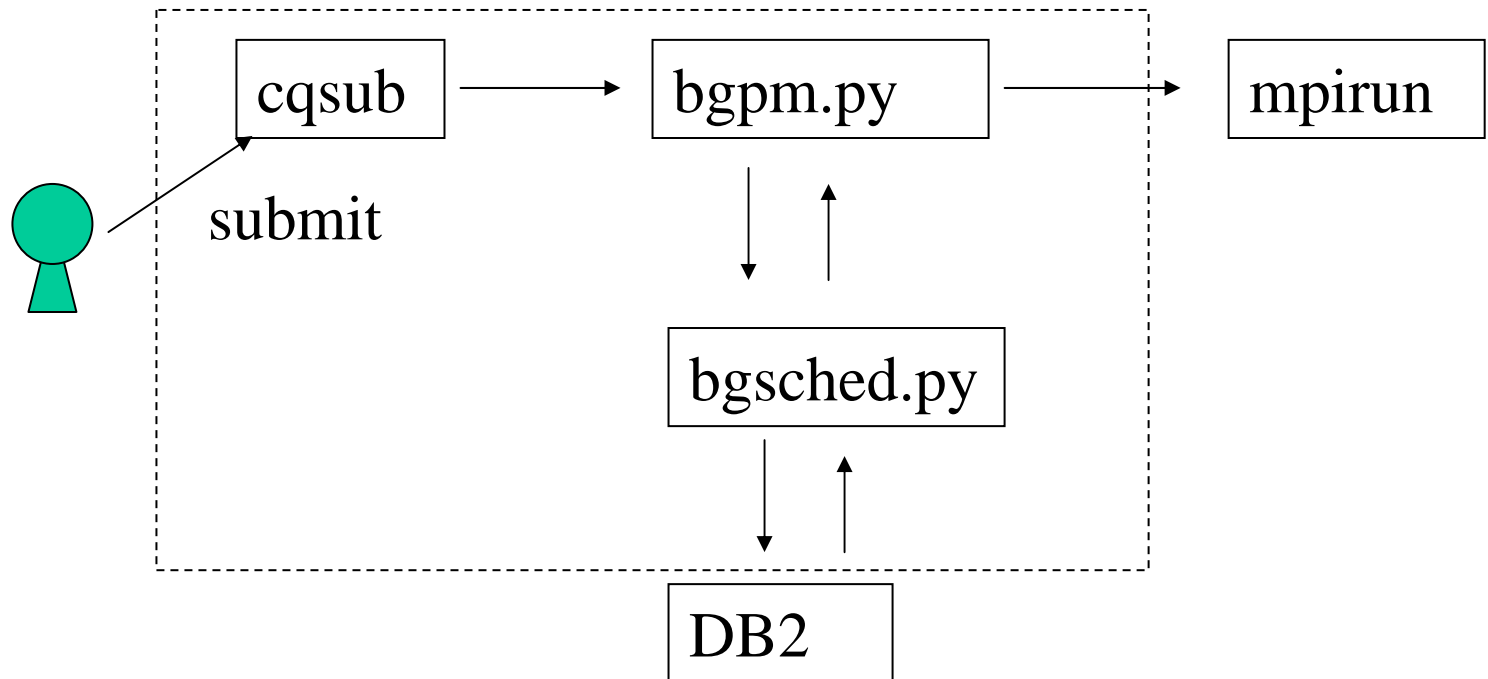
- **Job flow on Cobalt**
  - User submits a job to Cobalt without defining block.
  - Cobalt dispatches a job to free block at that time.
  - User can't define filesystems because he doesn't know which block will be used.



- **Where is job submitted? Cobalt only knows.**

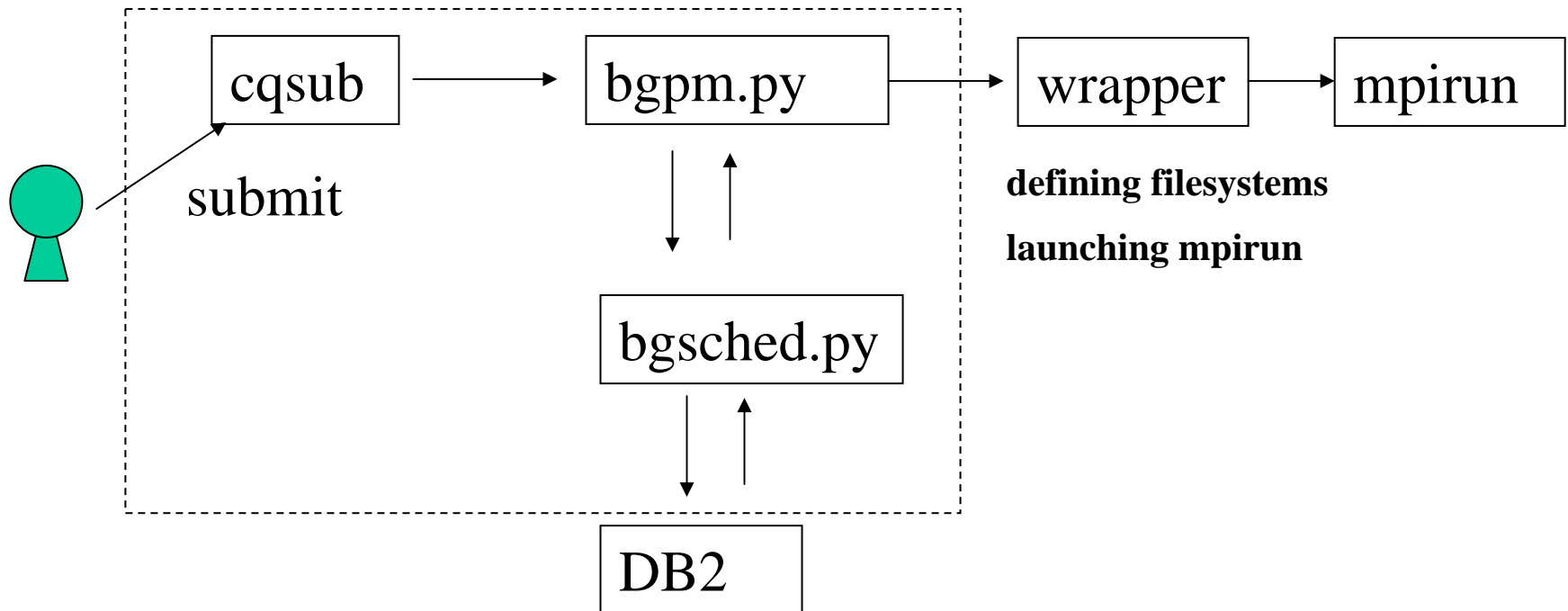
# Job flow on Cobalt (details)

- **Internal flow when dispatching**
  - Cobalt contacts with DB2 to find free block.
  - After Cobalt found a free block, it launches mpirun



# Customizing Cobalt

- **Intermediate wrapper**
  - **bgpm.py** calls wrapper instead of mpirun.
  - **Wrapper** defines filesystems for a block which is found by Cobalt, launches mpirun.



# Changing points(1)

- **changing bgpm.py**

```
bglsn:/usr/sbin # diff bgpm.py.org bgpm.py
```

```
111,112c111,112
```

```
<         cmd = (self.config['mpirun'], "mpirun", '-np', pnum, '-partition',  
partition,
```

```
<         '-mode', mode, '-cwd', cwd, '-exe', program)
```

```
---
```

```
>         cmd = (self.config['mpirun'], "mpirun_wrapper", '-np', pnum, '-  
partition', partition,
```

```
>         '-mode', mode, '-cwd', cwd, "", program)
```

```
114c114
```

```
<         cmd = cmd + ('-args', args)
```

```
---
```

```
>         cmd = cmd + (" ", args)
```

# Changing points(2)

- **/etc/cobalt.conf**

mpirun:/bgl/BlueLight/ppcfloor/bglsys/bin/mpirun\_wrapper

# Present status

- **Cobalt could bear more than thousand jobs queuing.**
  - amazing tough
  - but response was heavy....(cqstat, cqsub)
- **mixed scheduling**
  - manual submitting
  - Cobalt
  - that is because we need pre-allocation so that we often run short time jobs.

# opinion

- **We must customize it sometimes**
  - it is not so difficult because sources are small and simple.
  - I guess that it's because of Blue Gene side.
    - ex) mpirun options is changed.
  - Whenever I upgrade driver of Blue Gene, must check out some changing points.
- **Function we expect**
  - Queue priority(and attributes)
  - more detailed information with cqstat
  - pre-allocation



# **A year with Blue Gene**

- **less hardware troubles**
  - MTTI :about 3weeks
  - hot-swap in maintenance (per mid-plane)
- **less administration tools**
  - self-made tools
- **poor software development tools**
  - libraries, compilers are not matured.
  - important for application developers