ISBN 978-89-6211-278-8

가시화 시스템 사용자 가이드

(Visualization system user's guide)

구 기 범 (voxel@kisti.re.kr)

한 국 과 학 기 술 정 보 연 구 원
Korea Institute of Science & Technology Information

목차

그림 차례

소스 차례

1. 배경

본보고서는 KISTI의 주력 visualization 시스템인 Picasso(hostname: picasso.ksc.re.kr)의 전반적인 사용법에 대해 설명한다.

Picasso는 2008년 1월 말에 기본적인 설치를 완료했고, 2008년 1/4 분기 ~ 2/4분기 초반까지 LINPACK 벤치마크를 수행한 후 현재는 소수의 내/외 사용자에 대해 시범 서비스를 제공하고 있다. 슈퍼컴퓨터 4호기와 같은 계산 전용 시스템과 달리, Picasso는 별도의 물리적인 입/출력 장치를 갖추고 있을 뿐만 아니라 GPU와 같이 계산 시스템에서는 쉽게 볼 수 없는 특수목적 프로세서도 갖추고 있다. 그리고 데이터 visualization 작업은 순수한 MPI 작업과는 실행 형태가 많이 다르기 때문에 계산 시스템의 운영정책이나 스케줄러 정책을 그대로 적용하는 데에는 한계가 있다. 따라서 Picasso는 독자적인 운영정책에 따라서 운영되고 있으며, 사용자는 여기서 설명하는 사용 방법을 숙지할 필요가 있다.

Picasso의 사용 방법은 앞으로 큰 변화는 없겠지만 사용자의 증가, 내부 운영정책의 변경 등의 사유로 세부내용은 지속적으로 바뀔 수 있다. 하지만 기술보고서와 같은 문서 형태로 세세한 변경내역을 일일이 기록하는 것은 효율적이지도 않을뿐더러, 최종 사용자에게 알려지기까지의 시간이 너무 오래 소요될 것이다. 따라서 visualization 팀에서는 별도의 웹 페이지(http://sv.ksc.re.kr/svwiki/SystemManual)를 통해서 최신 매뉴얼을 언제라도 볼 수 있도록 했다.

2. 하드웨어 구성

Picasso의 하드웨어 구성은 '가시화 시스템의 최종 설치 사양'이나 웹 페이지(http://sv.ksc.re.kr/svwiki/SystemInformation)를 참고한다.

3. 로그인 방법

Picasso에 접속하는 방법은 on-site login과 remote login의 두 가지로 구분할 수 있으며, remote login은 SSH와 VNC의 두 가지로 다시 나눠진다.

가. On site login

Picasso의 콘솔에서 직접 로그인하는 것을 의미한다. 이때에는 입체영상, 트래킹 장비, 음향장비 등 Picasso를 구성하는 모든 요소를 직접 조작할 수 있다.

나. Remote login

1) SSH

Picasso는 현재 패스워드 인증을 사용하고 있지만 향후 id_rsa.pub를 이용하는 사설키/공개키 인증 방법으로 바꿔나갈 계획이다. id_rsa.pub는 리눅스나 상용 유닉스에서는 ssh-keygen 명령어를 이용하고, MS-Windows에서는 puttygen.exe를 이용해서 간단하게 만들 수 있다. 이미 사용하고 있는 id_rsa.pub가 있다면 그것을 그대로 사용해도 된다.

id_rsa.pub가 만들어지면 이 파일을 Picasso 관리자에게 e-mail로 보내면 된다.

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/foobar/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/foobar/.ssh/id_rsa.
Your public key has been saved in /home/foobar/.ssh/id_rsa.pub.
The key fingerprint is:
4b:a9:6b:d1:b3:31:8b:90:fb:e2:cc:8e:69:7b:6c:00 foobar@anyhost
```

[소스 3-1] RSA 키의 생성

2) TurboVNC

가) TurboVNC 접속 설정

Picasso는 TurboVNC를 VNC 클라이언트/서버로 이용한다. TurboVNC로 Picasso에 접속할 경우, master02번으로 연결된다(그 외의 호스트에 대해서는 방화벽으로 모두 막혀있다).

TurboVNC를 이용해서 Picasso에 접속하려면 먼저 Picasso에 해당 사용자의 계정이 만들어져 있어야 한다. TurboVNC를 위한 전용 패스워드를 만드는 방법은 다음과 같다. 이 패스워드는 보안을 위해서 일반적으로 로그인할 때 사용하는 패스워드와 다르게 지정하는 것이 바람직하다.

```
57 [johndoe:~ > /opt/TurboVNC/bin/vncpasswd
Using password file /home/johndoe/.vnc/passwd
Password:
Verify:
Would you like to enter a view-only password (y/n)? n
```

[소스 3-2] VNC 패스워드 설정

ssh를 이용해서 master02에 접속한 후 vncserver를 다음과 같이 실행한다. 특히 vncserver를 실행할 때 나타나는 출력 메시지에서 New X desktop is master02:1 부분을 유심히 봐둬야 한다.

```
73 [johndoe:~ > /opt/TurboVNC/bin/vncserver

New 'X' desktop is master02:1

Starting applications specified in /home/johndoe/.vnc/xstartup
Log file is /home/johndoe/.vnc/master02:1.log
```

[소스 3-3] VNC 서버의 실행

특별한 에러 메시지가 나타나지 않았을 경우 별다른 문제는 없겠으나,

만일을 위해 Xvnc가 다음과 같이 정상적으로 실행되고 있는지 확인한다. 특히 여러 명이 TurboVNC를 사용하고 있을 경우 Xvnc 다음에 나오는 :1(display)이 다르게 나올 수도 있으니 주의해서 확인한다.

```
76 [johndoe:~ > ps ax | grep vnc
22113 pts/14   S      0:00 /opt/TurboVNC/bin/Xvnc :1 -desktop X -h
ttpd /opt/TurboVNC/bin/../vnc/classes -auth /home/johndoe/.Xauth
ority -dontdisconnect -geometry 1240x900 -depth 24 -rfbwait 12000
0 -rfbauth /home/johndoe/.vnc/passwd -rfbport 5901 -fp unix/:7100
-deferupdate 1
```

[소스 3-4] VNC 서버의 실행 확인

Xvnc가 정상적으로 실행되고 있는 것을 확인하면 ssh 연결은 끊어도 된다. 그 다음 자신의 컴퓨터에서 TurboVNC 클라이언트를 실행한다. 클라이언트가 어떤 호스트에 접속할 지 물어올 때 'picasso.ksc.re.kr:1'과 같이 입력한다. 이 때 :1은 vncserver를 실행할 때 할당받은 번호를 사용하면 된다.

정상적으로 연결이 되면 클라이언트는 패스워드를 물어오는데, 이 때 vncpasswd를 이용해서 지정했던 패스워드를 입력하면 정상적으로 로그인이 된다.

나) OpenGL 어플리케이션의 실행

TurboVNC를 이용해서 Picasso에 접속했을 경우 OpenGL 어플리케이션을 그냥 실행시키면 다음과 같은 에러가 발생한다.

```
1 [johndoe:~ > glxgears
Xlib: extension "GLX" missing on display ":1.0".
Error: couldn't get an RGB. Double-buffered visual
```

[소스 3-5] VNC 서버를 실행한 상태에서의 OpenGL 어플리케이션 실행

이 문제는 VirtualGL로 해결할 수 있다. 이 패키지는 서버 쪽에만 설치하면 충분하기 때문에 개별 사용자가 자신의 컴퓨터에 설치할 필요는 없다(Picasso에는 이미 설치되어 있다).

```
3 [johndoe:~ > /opt/VirtualGL/bin/vglrun glxgears
8422 frames in 5.0 seconds = 1684.254 FPS
```

[소스 3-6] VirtualGL을 이용한 OpenGL 어플리케이션의 실행

다) TurboVNC 끝내기

일단 vncserver를 실행하면 클라이언트의 실행이 끝나더라도 Picasso에는 Xvnc가 계속 실행되고 있다. 따라서 클라이언트를 다음에 연결할 때에도 이전에 작업하던 환경이 그대로 복구된다는 장점이 있다. 하지만 다음과 같이 vncserver를 완전히 끝내는 것도 알아둘 필요가 있다.

```
78 [johndoe:~ > /opt/TurboVNC/bin/vncserver -kill :1
Killing Xvnc process ID 22113
```

[소스 3-7] VNC 서버의 종료

위에서 :1은 처음 vncserver를 실행할 때 할당받았던 display 번호를 그대로 사용하면 된다.

4. 시스템 사용 예약

시스템 사용 예약은 원격지 사용자에게는 해당되지 않는다. KISTI의 Visualizatoin Room에서 직접 작업을 하고자 할 경우 042) 869-0606으로 연락해서 사용 예약을 하고, 지정된 시간에 방문하면 된다.

5. 주요 소프트웨어 사용방법

가. 사용방법 일반론

1) 소프트웨어 설치 디렉토리

리눅스 배포판에 포함돼있는 소프트웨어나 RPM으로 설치되는 소프트웨어는 모두 기본 디렉토리에 설치되지만 그렇지 않은 소프트웨어는 특수한 경우(디바이스 드라이버 등)를 제외하고는 거의 대부분 /usr/local에 설치된다.

/usr/local의 모든 소프트웨어는 실제로는 /usr/local/소프트웨어-버전 형태의 디렉토리에 설치되지만 이 디렉토리를 가리키는, /usr/local/소프트웨어라는 symbolic link가 존재한다. 따라서 사용자는 /usr/local/소프트웨어 디렉토리에 소프트웨어가 존재하는 것으로 가정하고 환경을 설정하면 나중에 해당 소프트웨어의 업그레이드가 진행된다고 해도 사용자는 별도의 환경 설정을 바꾸지 않고도 최신 버전을 사용할 수 있게 된다.

```
lrwxrwxrwx  1 root root    13 Apr  7 13:17 PortAudio -> PortAudio-
v19/
drwxr-xr-x  4 root root  4096 Apr  7 13:16 PortAudio-v19/
lrwxrwxrwx  1 root root    10 Feb  4 21:09 POVRay -> POVRay-3.6/
drwxr-xr-x  6 root root  4096 Feb  4 21:07 POVRay-3.6/
lrwxrwxrwx  1 root root     8 Feb  3 20:04 Qt -> Qt-4.2.3/
drwxr-xr-x 12 root root  4096 Feb  3 20:01 Qt-4.2.3/
drwxr-xr-x 12 root root  4096 Feb  3 18:22 Qt-4.3.3/
lrwxrwxrwx  1 root root    10 Feb  4 23:11 QUANTA -> QUANTA-1.0/
drwxr-xr-x  8 root root  4096 Feb  4 23:19 QUANTA-1.0/
lrwxrwxrwx  1 root root     8 Apr  7 13:40 SAGE -> SAGE-3.0/
```

[소스 5-1] /usr/local 디렉토리의 내용(일부 발췌)

2) 64bit vs. 32bit

대부분의 경우 64-bit 라이브러리가 들어있는 디렉토리는 lib64, 32-bit 라이브러리가 들어있는 디렉토리는 lib 형태의 이름을 갖지만 /usr/local 밑에 설치되어있는 일부 소프트웨어의 경우 64-bit 라이브러리

도 lib 디렉토리에 존재하기도 한다. 따라서 /usr/local에 설치되는 소프트웨어를 사용할 때에는 64-bit 라이브러리와 32-bit 라이브러리의 설치 위치 등을 정확히 파악해서 LD_LIBRARY_PATH를 설정해야 한다(SAGE, Torque, Gelato, VTK, CR, CUDA 등).

3) LD_LIBRARY_PATH

/usr/local에 설치하는 소프트웨어는 거의 대부분 LD_LIBRARY_PATH를 별도로 지정해줘야 해당 소프트웨어의 shared object를 사용할 수 있지만, 자주 사용하는 몇몇 프로그램에 대해서는 /etc/ld.so.conf.d에 등록해서 굳이 LD_LIBRARY_PATH를 설정하지 않아도 shared object를 사용할 수 있도록 했다. 여기에는 CAVELib, Chromium, CUDA, QUANTA, VTK, MVAPICH2 등이 해당된다.

나. 배치(batch)작업의 실행

Picasso는 real-time 어플리케이션의 실행에 중점을 두지만 batch 작업도 실행할 수 있도록 노드를 구성했다. Picasso는 Torque(버전 2.3.2)를 batch scheduler로 사용한다.

1) Picasso에서의 작업(job) 구분

Picasso에서 실행하는 모든 작업(job)은 다음과 같이 구분한다.

- **Interactive job** : 주로 master01과 display01~display16을 이용하는 real-time rendering, video streaming 등의 작업이 여기에 해당한다.
- **GPU batch job** : User interaction이 없고 주 계산을 GPU로 수행하는 프로그램들은 모두 이 범주에 들어간다. 특성상 CPU는 GPU 계산을 위한 전처리/후처리 역할만을 주로 수행하기 때문에 많은 CPU를 사용하지 않는 경향이 있다.
- **CPU batch job** : User interaction이 없고 주 계산을 CPU로 수행

하는 프로그램들은 모두 이 범주에 들어간다. 일반적으로 우리가 생각하는 MPI 작업이 여기에 해당된다.

- **Rendering farm job** : Picasso를 rendering farm으로 사용할 때에만 나타나는 작업으로, Torque로 통제가 불가능한 형태의 작업이다. (아래 참고)

2) Picasso에서의 작업 제출(submit) 방법

Torque에 batch 작업을 제출하는 방법은 여타 클러스터 환경에서의 방법과 동일하다. 가장 먼저 해야 할 일은 작업 제출 스크립트를 작성하는 것이다. 아래의 스크립트는 MPI를 사용하는 작업을 제출하는 스크립트다. 여기서 주의해야 할 점은 mpirun이 아닌 mpiexec를 사용한다는 것이다.

```
$ cat pbs.script
#!/bin/bash
#PBS -l nodes=90:ppn=2
#PBS -q batch
#PBS -N cpi
#PBS -o out
#PBS -e err

cd /home/voxel/tmp
/usr/local/MPI/bin/mpiexec -n 180 ./cpi
```

[소스 5-2] PBS 스크립트

#PBS -l nodes=90:ppn=2 에서 nodes는 90을 초과할 수 없고, ppn은 6을 넘어갈 수 없다. 따라서 하나의 작업이 사용할 수 있는 최대 CPU의 수는 540으로 제한한다. Batch 작업을 실제로 호스트에 할당하는 것은 Torque가 맡아서 처리하므로 별도의 machinefile은 필요하지 않다.

한 가지 주의해야 할 사항은 Torque 스케줄러의 경우 multi-thread를 같이 사용하는 작업에 대한 고려가 전혀 없다는 점이다.

스크립트를 작성한 후의 작업 제출은 qsub 명령어를 이용한다.

```
$ qsub pbs.script
```

[소스 5-3] qsub를 이용한 작업 제출방법

제출한 작업의 상태는 qstat 명령어로 확인할 수 있다.

```
[bongju@master02 script]$ qstat -a

scheduler:
                                                            Req'd   R
eq'd   Elap
Job ID             Username Queue    Jobname    SessID NDS   TSK Me
mory Time  S Time
-------------------- -------- -------- ---------- ------ ----- --
- ------ ----- - -----
44.scheduler-ib      bongju   batch    testpbs     20129   --   --
 --    --  E 00:00
```

[소스 5-4] qstat를 이용한 작업 확인

작업이 실행되는 중간에 끝내려면 qdel 명령어를 이용한다. 간혹 qdel을 해도 실제로는 작업이 계속 돌아갈 수 있는데, 이때에는 직접 노드를 확인해서 작업을 죽여야 한다. 그게 어려울 경우 시스템 관리자에게 문의하면 된다.

```
$ qdel 44.scheduler-ib
```

[소스 5-5] 작업의 취소

다. 클러스터 환경을 위한 도구

Picasso는 클러스터 형태로 구축된 만큼 몇몇 어플리케이션을 실행할 때나 문제가 발생할 경우 클러스터 환경에 맞도록 조작해야 한다.

1) pdsh

여러 노드에서 같은 명령어를 실행할 때 pdsh를 이용하면 편리하다. Picasso에서는 주로 master 노드에서 여러 대의 display나 render 노드에 대해 명령어를 실행하고자 할 때 사용한다. 일반 사용자는 display, render에 대해서만 이 명령어를 사용한다.

각 노드에서 실행한 명령어가 출력하는 메시지는 모두 사용자의 콘솔로 redirect되기는 하지만 호스트 이름에 따라서 정렬되지는 않는다. 따라서 사용자가 별도로 sort를 실행해야 한다.

```
$ pdsh -w display[01-16] hostname | sort
display01: display01
display02: display02
display03: display03
display04: display04
display05: display05
display06: display06
display07: display07
display08: display08
display09: display09
display10: display10
display11: display11
display12: display12
display13: display13
display14: display14
display15: display15
display16: display16

$ pdsh -w display[01-16] -x display15 hostname | sort
display01: display01
display02: display02
display03: display03
display04: display04
display05: display05
display06: display06
display07: display07
display08: display08
display09: display09
display10: display10
display11: display11
display12: display12
display13: display13
display14: display14
display16: display16
```

[소스 5-6] pdsh를 이용한 명령어 수행

2) CAVERUN

CAVELib을 사용하는, 특히 프로젝터와 스크린을 사용하는 어플리케이션은 master01과 display01 ~ display16에서 모두 동일하게 실행해야 정상적인 출력이 가능하다. 하지만 매번 pdsh를 이용하는 것은 귀찮은 작업이기 때문에 별도로 제공하는 caverun이라는 스크립트를 이용한다.

```
$ ls -l /usr/local/bin/caverun
...
-rwxr-xr-x 1 root root   236 Sep  3 11:00 caverun*
...

$ caverun sample
Running sample on master node ...

...

(시간 지연 : 1초)
Running ls on client nodes ...

...

(시간 지연 : 1초)
Cleaning up IPC ...

...
```

[소스 5-7] caverun을 이용한 CAVE 어플리케이션 실행

3) cleanIPC

CAVELib으로 작성한 어플리케이션을 여러 번 실행/종료하다보면 shared memory나 semaphore를 제대로 반환하지 못해서 다음에 CAVELib 기반 어플리케이션을 실행하고자 할 때 shared memory/semaphore를 할당받지 못했다는 에러 메시지와 함께 제대로 실행되지 않는 경우가 있다. 비록 앞에서 설명한 caverun 스크립트 내에서 cleanIPC를 실행하기 때문에 이런 문제가 발생할 가능성이 줄어들기는 했지만 완전히 제거했다고 말할 수는 없다. 따라서 필요에 따라서 사용자가 직접 cleanIPC를 실행할 수 있도록 했다.

```
$ cleanIPC
```

[소스 5-8] 단일 노드에서의 cleanIPC의 실행

CAVELib 어플리케이션은 display 노드에서도 실행되므로 필요할 경우 display 노드에 대해서도 cleanIPC를 실행하는 것이 좋다.

```
$ pdsh -w display[01-16] cleanIPC
```

[소스 5-9] pdsh를 이용한 다수 노드에서의 cleanIPC 실행

앞에서도 설명했지만 caverun 스크립트 내에서도 cleanIPC를 실행하기 때문에 사용자가 cleanIPC를 직접 실행할 필요는 거의 없을 것이다.

관리자(root)가 직접 cleanIPC를 실행하면 trackd, X-Windows 등 다른 어플리케이션까지 영향을 받을 수 있으므로 절대로 관리자 계정으로는 실행하지 않도록 한다. 정 필요할 경우에는 개별 shared memory, semaphore의 사용현황을 일일이 파악해서 필요한 것만 삭제해야 한다.

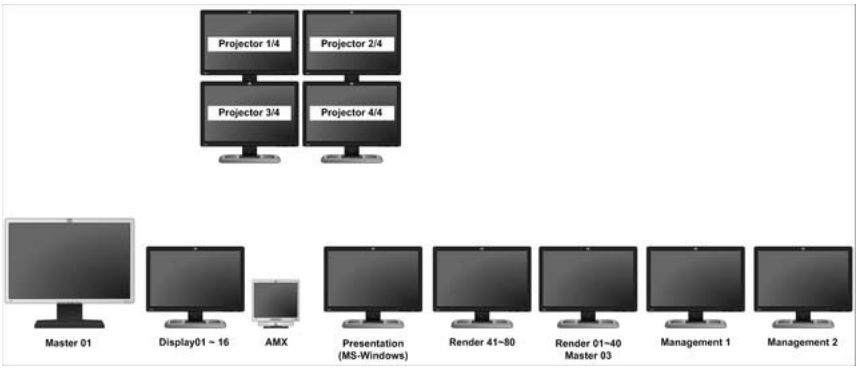
6. 주요 장비의 사용방법

가. 콘솔의 구성

Picasso의 콘솔 데스크는 [그림 6-1]과 같이 구성돼있다. Picasso의 컴퓨팅 시스템은 기본적으로 많은 노드로 구성된 클러스터이기 때문에 다양한 용도의 콘솔을 동시에 운영해야 한다.

콘솔을 구성하는 모니터의 실제 역할은 [그림 6-2]와 같다.

[그림 6-1] 콘솔 데스크



[그림 6-2] 각 콘솔 모니터의 역할

- Master 01 : 이 모니터는 master01 노드와 직접 연결돼있고, 스크린과 프로젝터를 사용하는 모든 작업의 시작점이다. 따라서 가시화 실험실을 방문하는 사용자는 모두 이 모니터를 사용한다.
- Display01~16 : 이 모니터는 master02와 모든 display 노드의 출력을 확인하는 데에 사용한다. 각 노드의 출력을 전환하려면 Master 01 키보드의 scroll lock 키를 두 번 연속으로 누르면 된다.
- AMX : 프로젝터와 오디오관련 기기를 조작한다. 일반 사용자의 직접 조작은 지양하고 있다.

- Presentation : MS-Windows가 설치되어 있다. 원래는 프리젠테이션 전용 PC로 활용하려고 했으나, 전용 프로젝터의 도입이 어려워져서 현재는 제대로 사용하지 못한다는 단점이 있다.
- Render 41~80 : master03번과 render node와 연결되어 있으며 ATEN kvm으로 연결되어 scroll lock을 두 번 연속으로 누르면 대상 노드를 바꿀 수 있다.
- Render 01~40 :
- management 1 : input node와 연결되어 있으며 trackdserv와 master01노드의 trackd가 항상 실행되어 있는 상태이다.
- management 2 : 스토리지서버와 게이트웨이 서버, 매니지먼트서버들을 모니터링 한다. HP kvm으로 연결되어 프린트스크린을 한 번 누르면 노드를 바꿀 수 있다.

나. 가상현실 입력 장치

1) 완드 / 헤드트래커

Picasso의 가상현실 입력 장치는 InterSense의 IS-900 MicroTrax와 BG Systems의 FlyBox가 준비되어 있다. 기본 장치는 IS-900 Micro Trax으로 설정돼있는데, 만약 FlyBox를 사용하고자 한다면 trackd와 trackdserver의 설정을 변경해야 한다. 이는 시스템의 관리자 권한이 있어야만 가능한 작업이므로 관리자에게 문의해야 한다.

[그림 6-3] Picasso의 헤드트래커(좌)와 완드(우)

완드의 버튼 배치는 [그림 6-4]와 같다.

- Button 1 : 중앙 왼쪽
- Button 2 : 왼쪽
- Button 3 : 중앙 오른쪽
- Button 4 : 오른쪽
- Button 5 : 가운데 (조이스틱)
- Button 6 : 아래쪽 trigger

[그림 6-4] 완드의 버튼

가장 왼쪽 버튼을 수초간 누르면 전원이 들어오고, 왼쪽 버튼과 오른쪽 버튼을 동시에 누르면 꺼진다. 헤드 트래커는 충전 배터리의 버튼을 누르면 바로 켜진다. 반대로 전원을 내릴 때에는 버튼을 수 초간 누르고 있어야 한다.

Wand와 head tracker를 사용할 때 전원을 올린 후 일정 시간(수 초 ~ 수십 초)이 경과해야 정상적인 작동을 보장할 수 있다. 두 장치 모두 사용이 끝나면 충전 스테이션에 장착해서 배터리가 방전되지 않도록 한다([그림 6-5]).


[그림 6-5] 충전 스테이션

배터리 사용시간이 1시간 정도 남으면 low battery 임을 알리는 깜빡임이 있고, 10분 동안 아무 움직임이 없으면 자동으로 트래킹을 멈추고 전원이 꺼진다.

2) FlyBox

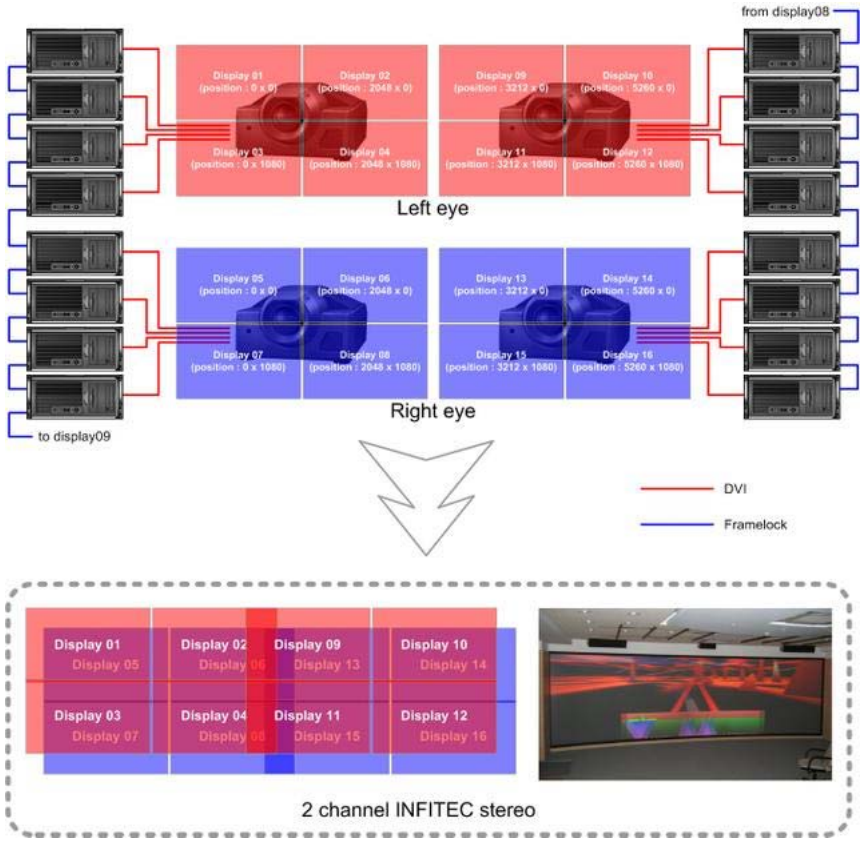
앞에서도 설명했지만 헤드트래커와 완드 대신 사용할 수 있는 입력장치로, Picasso의 콘솔데스크에 위치해있다. 하지만 이 장비를 사용하기 위해서는 trackd의 설정을 변경해야 하므로, 실제로 사


[그림 6-6] FlyBox

용하기 전에 관리자에게 문의해야 한다.

다. 프로젝터

Picasso는 4대의 SONY 4프로젝터를 갖추고 2채널 입체영상을 구현한다. 스테레오 모드로 작동할 때, top 프로젝터들은 왼쪽 눈 영상을, bottom 프로젝터들은 오른쪽 눈 영상을 출력한다.



[그림 6-7] Picasso의 프로젝터 구성

프로젝터는 램프 사용시간에 따라서 전체 화면의 밝기 등에 큰 차이를 보여주는데, Picasso의 경우 일반적인 램프 수명은 1000시간이고, 실제로는 800시간 정도 사용하면 화질저하가 뚜렷해진다. 그리고 일단 프로젝터의 전원을 올린 후 최소 6시간 이상 사용하지 않은 상태로 전원을 내리면 6시간 사용한 것보다 더 수명이 줄어든다는 점을 참고해야 한다.

7. 결론

지금까지 Picasso의 전반적인 사용법에 대해 설명했다. Picasso는 여러 가지 주변기기가 연결된 중규모 클러스터인 만큼 다양한 기기의 사용법을 숙지하는 것이 바람직하다.

Picasso를 구성하는 물리적인 기계장치는 사용법에서는 큰 변화가 없겠지만 소프트웨어 설치 등 운영정책에 따라서 실제 사용법이 바뀔 수 있는 여지도 있으므로 사용자는 웹 페이지(http://sv.ksc.re.kr/svwiki/VisualizationSystemUserGuide)를 통해서 최신 내용을 항시 확인해야 한다.