



# VisIT: 대용량 데이터 분석에 대한 연구

(Analysis on large data visualization of VisIT)

허영주 (po<sup>pea</sup>@[kisti.re.kr](mailto:po<sup>pea</sup>@kisti.re.kr))

한국과학기술정보연구원  
Korea Institute of Science & Technology Information

---

---

# 목차

1. 서론 .....	1
2. 개요 .....	2
3. VisIt의 주요 기능 .....	4
가. 다양한 가시화 기능 .....	4
나. 정성적, 정량적 가시화/분석 기능 .....	8
다. 다양한 메쉬 데이터 지원 .....	10
라. 강력한 GUI .....	10
마. 테라급 데이터 가시화를 위한 병렬/분산 구조 .....	12
바. C++, Python 및 Java와의 인터페이스 .....	12
사. 동적으로 로딩되는 플러그인 .....	12
아. 멀티-플랫폼 지원 .....	13
4. VisIt Challenge .....	14
가. Index 1. ....	18
1) 플랫폼: Graph .....	18
2) 실험 개요 .....	18
나. Index 2. ....	19
1) 플랫폼: Dawn .....	19
2) 실험 개요 .....	19
다. Index 3. ....	21
1) 플랫폼: Jagure(Cray) .....	21

2) 실험 개요 .....	21
라. Index 4. ....	22
1) 플랫폼: Franklin(Cray XT4) .....	22
마. Index 5. ....	23
1) 플랫폼: Jagure(Cray) .....	23
바. Index 15. ....	24
1) 플랫폼: Eureka .....	24
5. 결론 .....	26

## 그림 차례

[그림 3-1] 3차원 볼륨 데이터의 표면을 나타내는 Contour Plot .....	5
[그림 3-2] 온도를 나타내는 Pseudocolor Plot .....	5
[그림 3-3] 분자 구조를 나타내는데 사용되는 Molecule Plot .....	5
[그림 3-4] 데이터를 원뿔의 표면으로 투사해서 보여주는 Cone Operator	6
[그림 3-5] cell 주변의 local region에 있는 field를 보여주는 onion-peel operator .....	7
[그림 3-6] scalar field의 값이 특정 영역 내에 있는 셀을 나타내는데 사용된 threshold operator .....	7
[그림 3-7] Line-Out 기능 .....	8
[그림 3-8] Pick Operation .....	9
[그림 3-9] query를 통해 얻은 정량적 정보 .....	9
[그림 3-10] VisIt의 GUI: 메인 컨트롤 패널과 가시화 윈도우 .....	10
[그림 3-11] VisIt의 GUI: 애니메이션 생성에 사용되는 키프레임 윈도우	10
[그림 3-12] VisIt의 GUI: volume plot attributes 윈도우 .....	11
[그림 4-1] JagurePF(OLCF/ORNL의 Cray XT5)에서 32,000개의 Opteron core를 사용해서 생성한 2TZ 데이터의 isosurface 이미지 .....	15
[그림 4-2] Franklin(NESRC/LBNL의 Cray XT4)에서 32,000개의 Opteron core를 사용해서 생성한 2TZ 데이터의 볼륨 렌더링 이미지 .....	15
[그림 4-3] Eureka로 가시화한 Turbulent Flow 데이터 .....	24

## 표 차례

[표 4-1] Vis Top 15 .....	16
--------------------------	----

---

## 1. 서론

VisIt은 DOE(Department of Energy) 산하의 ASCI(Advanced Simulation and Computing Initiative)에서 개발한 소프트웨어로, 테라(tera)급 시뮬레이션 결과를 가시화하고 분석할 용도로 만들어졌다. VisIt의 초기 버전은 2002년 가을에 출시됐으며, 첫 출시 당시부터 테라 규모의 데이터를 지원할 목적으로 개발됐다. 테라급 데이터를 분석하는데 필요한 복잡한 기능을 갖추고 있음에도 불구하고 VisIt은 오픈소스 프로그램으로 공개돼 있으며, 2차원, 혹은 3차원의 structured/unstructured 메쉬를 모두 지원할 뿐만 아니라 데이터의 분산/병렬 처리가 가능하다. 또, 리눅스, 윈도우, MacOS 등의 다양한 운영 체제를 지원하며, 사용자 인터페이스는 데스크탑에서, 실제 계산 처리는 원격의 컴퓨터에서 병렬로 수행하는 기능도 지원한다.

이렇듯 VisIt은 우수한 기능을 갖추고 있는 선도적인 응용 프로그램이므로 대용량 데이터의 실시간 분석을 목표로 하는 소프트웨어라면 VisIt과의 성능 비교가 필수적이라 할 수 있겠다.

본 문서에서는 VisIt에 대한 분석과 함께, VisIt의 성능 측정 및 미래 데이터 처리 가능성에 대해 타진한 VisIt Challenge의 다양한 결과에 대한 비교분석을 다룰 것이다. 그리고, 이 분석 내용을 지표로 삼아 향후 GLOVE(GLOBAL Virtual Environment for research) 개발 및 성능 향상의 방향 및 목표점을 설정해야 할 것이다.

---

## 2. 개요

VisIt은 다양한 플랫폼에서 과학 데이터를 가시화하고 분석하기 위한 툴로, 사용자와의 상호 작용을 위한 GUI와 대용량 데이터 분석을 위한 병렬 처리 기능을 제공한다. VisIt의 가장 큰 특징은 테라급의 대용량 데이터를 지원한다는 데서 찾아볼 수 있는데, 이런 대용량데이터 뿐만 아니라 킬로바이트 급의 작은 데이터도 지원 가능하다.

VisIt 애플리케이션은 거대 데이터셋의 병렬화된 후처리를 위한 인프라로 볼 수 있으며, 데이터 탐색, 비교 분석, 비주얼 디버깅, 정량적 분석 및 그래픽 처리 등의 목적으로 사용할 수 있다.

VisIt은 여러 썬드-파티 라이브러리를 사용하는데, Qt 위젯 라이브러리는 사용자 인터페이스에, Python 프로그래밍 언어는 커맨드-라인 인터프리터 구현에, 그리고 VTK(Visualization ToolKit) 라이브러리는 데이터 모델과 다양한 가시화 알고리즘에 활용하고 있다. VisIt 자체 기능의 개발에는 50man/year가 투입됐다. 이런 기능으로는 대용량 데이터 처리를 위한 병렬화 기능, 사용자 인터페이스, 커스텀 데이터 분석 루틴, (ARM(Adaptive Refinement Meshes나 mixed materials zone같은) 비표준화된 데이터 모델 지원 및 전체 기능 안정화 등을 들 수 있겠다.

VisIt의 썬드-파티 라이브러리의 코드만 해도 백만 라인이 넘으며, VisIt의 자체 코드는 1.5백만 라인에 달한다. 윈도우, Mac, AIX, IRIX, Solaris, Tru64 등을 포함하는 다양한 UNIX 운영체제, SGI의 Altix, Cray의 XT4를 포함하는 다양한 리눅스 운영체제 등을 모두 지원한다.

VisIt의 기본적인 설계 구조는 클라이언트-서버 모델로써, 서버가 병렬화돼있는 구조이다. 일반적으로 클라이언트-서버 모델은 원격지 셋팅에 있어서 효과적인 가시화를 가능케 해주며, 서버 병렬화는 어느정도 타당성 있는 상호 작용을 보장하면서 대용량 데이터를 처리할 수 있게 해준다. VisIt은 다양한 대용량 데이터를 가시화하는데 사용돼 왔는데, VisIt으로 가시화할 수 있었던 대용량 데이터의 사례를 들자면 다음과 같다.

- 2천 160억개의 데이터 포인트를 가진 structured grid

- 
- 10억개의 포인트를 가진 파티클 시뮬레이션
  - 1억~10억 개의 element를 지닌 curvilinear/unstructured/ARM 메쉬

VisIt은 데이터 플로우 네트워크 패러다임을 따른다. 이 패러다임은 서로 다른 모듈을 연결함으로써 분석을 수행케 하는 것이다. 이 모듈들은 VisIt의 5개의 주요 사용자 인터페이스 추상 모듈로부터 나오는 것으로, 각 모듈에 대한 사례는 매우 다양하다. 21개의 'plot' (데이터를 렌더링하는 방법)이 있으며, 42개의 'operator' (데이터를 조작하는 방법)이 있다. 85개의 파일 포맷 리더가 있으며, 50개가 넘는 'query' (정량적 정보를 추출하는 방법)이 있고, 거기에 100개가 넘는 'expression' (수량을 생성하는 방법)이 있다. 거기에 새로운 plot, operator 및 데이터베이스 모듈을 동적으로 구성할 수 있는 플러그인 기능도 있다. 이런 플러그인은 부분적으로는 코드로부터 생성할 수 있고, 심지어는 Qt와 Python 사용자 인터페이스의 자동 생성 기능을 포함하기도 한다.

VisIt 프로젝트는 본래 DOE의 ASC의 일환으로 LLNL(Lawrence Livermore National Laboratory)에서 시작했으나, 여러 다양한 그룹에 의해 분산 수행되는 프로젝트가 됐다. 현재 프로젝트를 주로 진행하는 곳은 VACET, ASC 및 GNEP(Global Nuclear Energy Partnership from the DOE's Office of Nuclear Energy)며, 이중 프로젝트를 주로 진행하는 곳은 VACET이라 할 수 있다. VACET(Visualization and Analytics Center for Enabling Technologies)은 DOE의 SciDAC(Scientific Discovery through Advanced Computing) 프로그램에서 파생된 조직으로, 과학 데이터 가시화 및 분석 소프트웨어 기술에 초점을 두고 있는 조직이다.

프로젝트에는 여러 기관 및 대학에 소속돼 있는 25명의 개발자가 참여하고 있다. VisIt 프로젝트는 2005년에 R&D 100 award를 수상했으며, 1년에 5만 건 이상의 다운로드 횟수를 기록하고 있다. 또, 세계 최고라 일컬을 수 있는 최상위 6개의 컴퓨터 시스템에서 데이터 분석을 위해 사용하고 있는 강력한 툴이다.



---

### 3. VisIt의 주요 기능

VisIt의 주요 기능은 다음과 같다.

#### 가. 다양한 가시화 기능

VisIt은 스칼라, 벡터 및 텐서 필드 가시화를 위한 매우 다양한 기능을 지원한다. VisIt의 가시화 기능은 다음의 2가지 카테고리로 나뉘볼 수 있다.

- Plot: Plot은 boundary, contour, curve, label, mesh, pseudocolor, scatter, streamline, subset, surface, tensor, vector 및 volume 등을 포함하며 데이터를 가시화하는데 사용된다.
- Operator: operator는 가시화에 앞서 데이터에 수행되는 동작을 나타낸다. 예를 들자면 slice, 인덱스 선택, iso-surface, onion peel, reflect, threshold, 부분 선택 등을 들 수 있다.

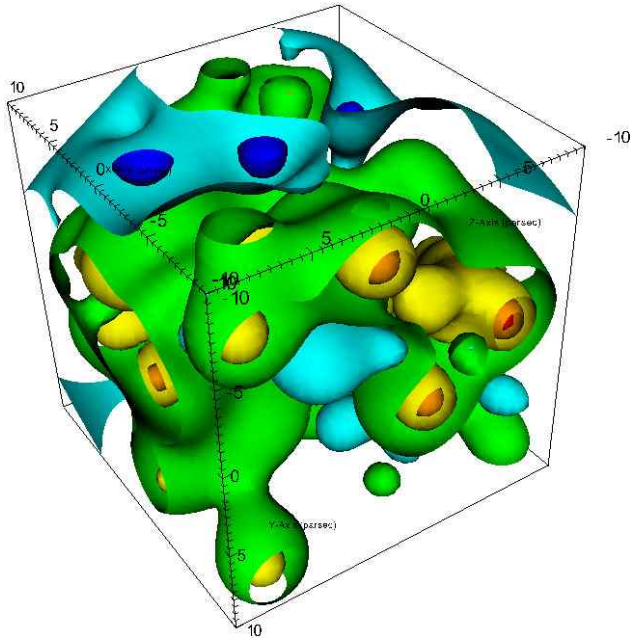
<표 3-1> VisIT이 제공하는 가시화 기능

Plots	Operators	General
Contour	Slices(cone, plane, sphere)	parallel version available
Mesh	Index Select	Animation
Vector	Iso-Surface	Movie-making
Subset	Onion Peel	Direct manipulation tools
Pseudocolor	Reflect	Pick & Query
Surface	Threshold	Line-out
Curve	Transform	Status bars
Volume	Replicate	Interruptable
Scatter	Resample	Remote data access
Spreadsheet	Revolve	Fault Tolerant
Molecule	Elevate	Scalable rendering
Parallel Axis	Project	Online help
Histogram	Displace	Read data directly from simulation memory
Label	Subset restriction	Material interface reconstruction
Truecolor	Expressions	Reads over 60 different file formats
Tensor		Scriptable

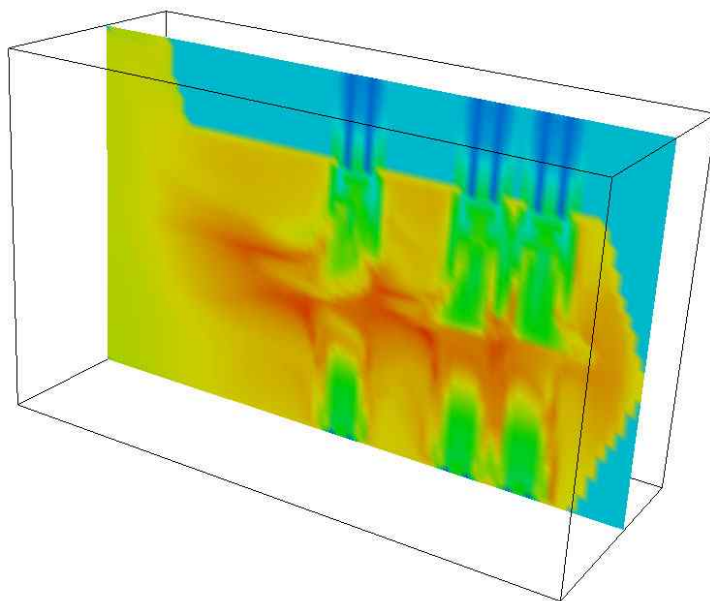
---

VisIT은 2D와 3D 데이터를 모두 다룰 수 있으며 데이터 애니메이션 기능도 제공한다. VisIT이 제공하는 기능은 <표 3-1>에서 볼 수 있다.

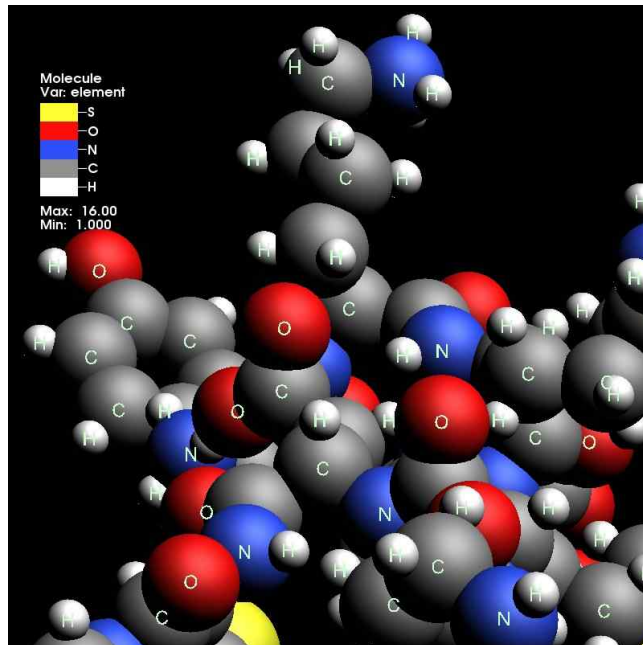
<그림 3-1> ~ <그림 3-3>은 VisIt의 다양한 Plot 기능을, <그림 3-4> ~<그림 3-6>은 Operator 기능을 나타낸다.



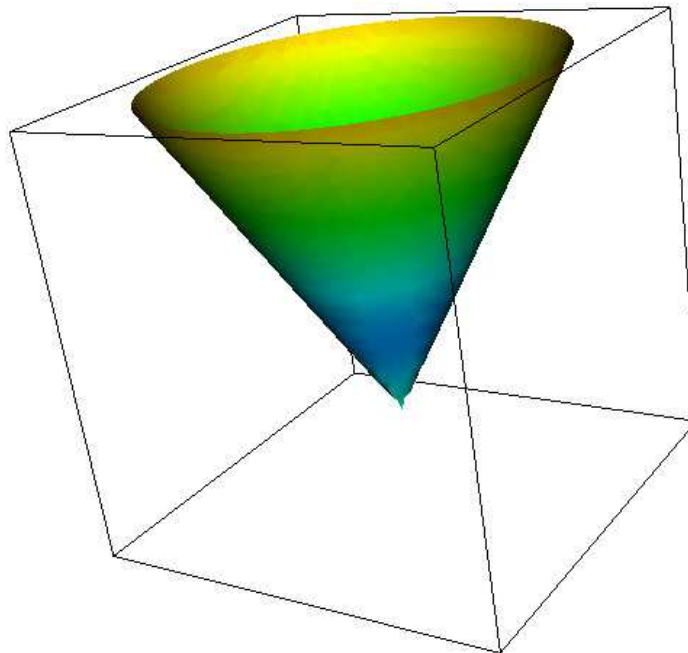
<그림 3-1> 3차원 볼륨 데이터의 표면을 나타내는 Contour Plot



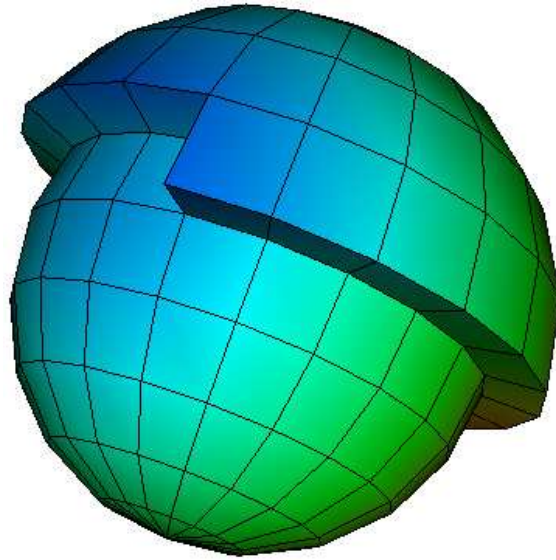
<그림 3-2> 온도를 나타내는 Pseudocolor Plot



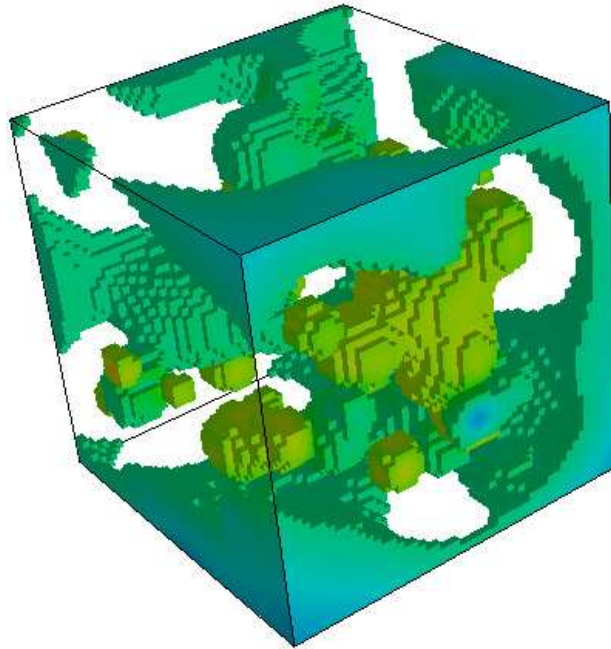
<그림 3-3> 분자 구조를 나타내는데 사용되는  
Molecule Plot



<그림 3-4> 데이터를 원뿔의 표면으로 투사해서  
보여주는 Cone Operator



<그림 3-5> cell 주변의 local region에 있는 field를 보여주는 onion-peel operator



<그림 3-6> scalar field의 값이 특정 영역 내에 있는 셀을 나타내는데 사용된 threshold operator

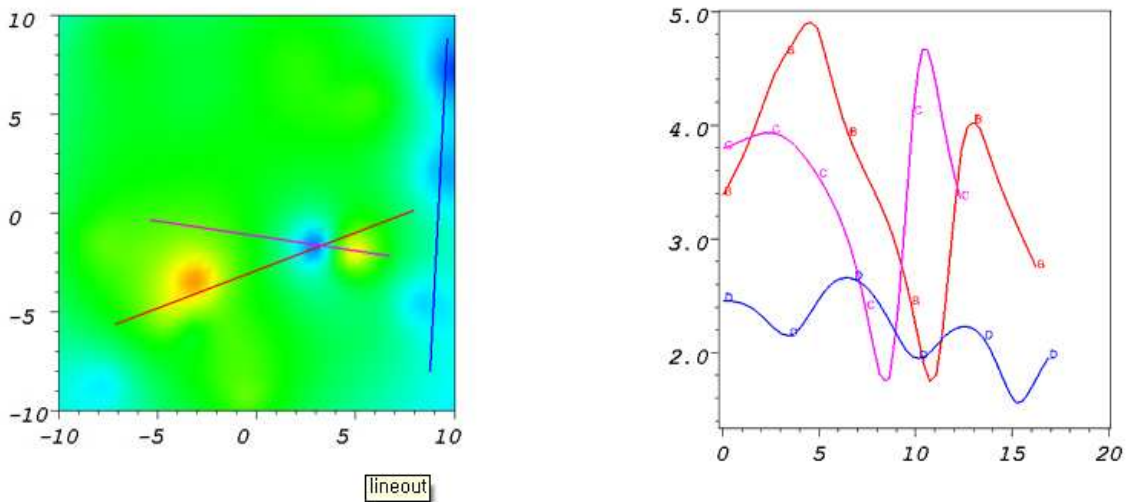
## 나. 정성적, 정량적 가시화/분석 기능

VisIt은 파생 필드를 위한 기능을 제공하는데, 이 기능은 현재 필드를 토대로 계산을 통해 생성한 새로운 필드를 지원한다는 뜻이다. 예를 들어 데이터셋에 velocity 필드가 있을 경우, velocity magnitude 필드를 생성하는 것이 가능하다.

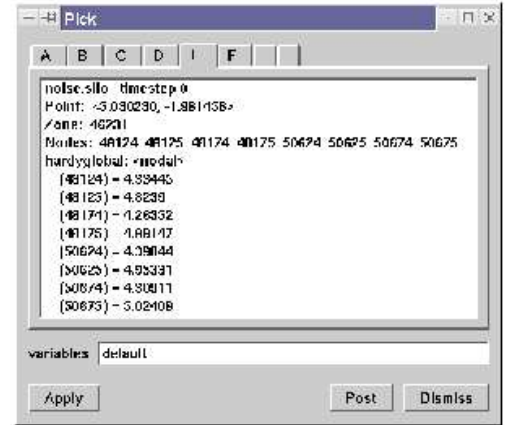
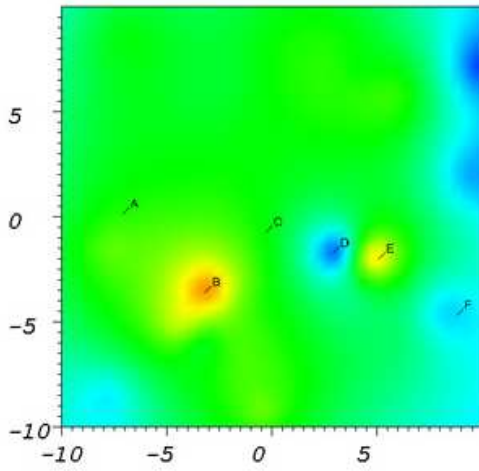
VisIT의 정량적 분석 툴은 다음과 같은 기능을 포함한다.

- Line-out: 마우스를 이용한 상호작용으로 line을 정의함으로써 고차원의 데이터셋에서 곡선을 생성하는 것이 가능하다.
- Pick & query: 이미지 클릭을 통해 데이터에 query를 던지는 것이 가능하다. query는 일종의 연산으로 데이터셋으로부터 데이터를 사용해서 계산된 값을 리턴하게 된다.

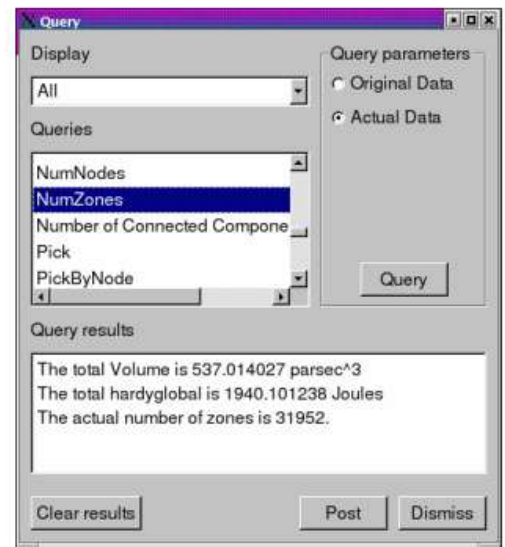
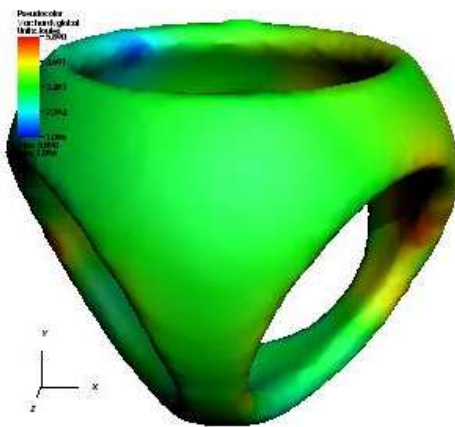
VisIT은 query 인터페이스를 제공하는데, 이 인터페이스를 통해서 볼륨이나 surface 영역에서 파생 분석을 수행하는 것이 가능하다. <그림 3-7>~ <그림 3-9>를 보면 이런 분석 툴의 예를 볼 수 있다.



<그림 3-7> Line-out 기능을 수행하면 line을 따라 거리 함수를 수행함으로써 필드 값에 대한 곡선을 생성한다. 왼쪽 이미지는 스칼라 필드와 커브 생성을 위해 사용한 라인을, 오른쪽 이미지는 결과로 생성된 커브를 보여준다.



<그림 3-8> Pick operation은 선택된 점에 대한 정보를 나타낸다. 이렇게 나타낼 수 있는 정보에는 위치, cell ID, node ID, 필드값 등이 있다. 왼쪽 이미지는 스칼라 필드에서 사용자가 선택한 지점을 나타내고, 오른쪽 이미지는 선택된 점에 대한 정보를 나타낸다.



<그림 3-9> query를 통해 얻은 정량적 정보.

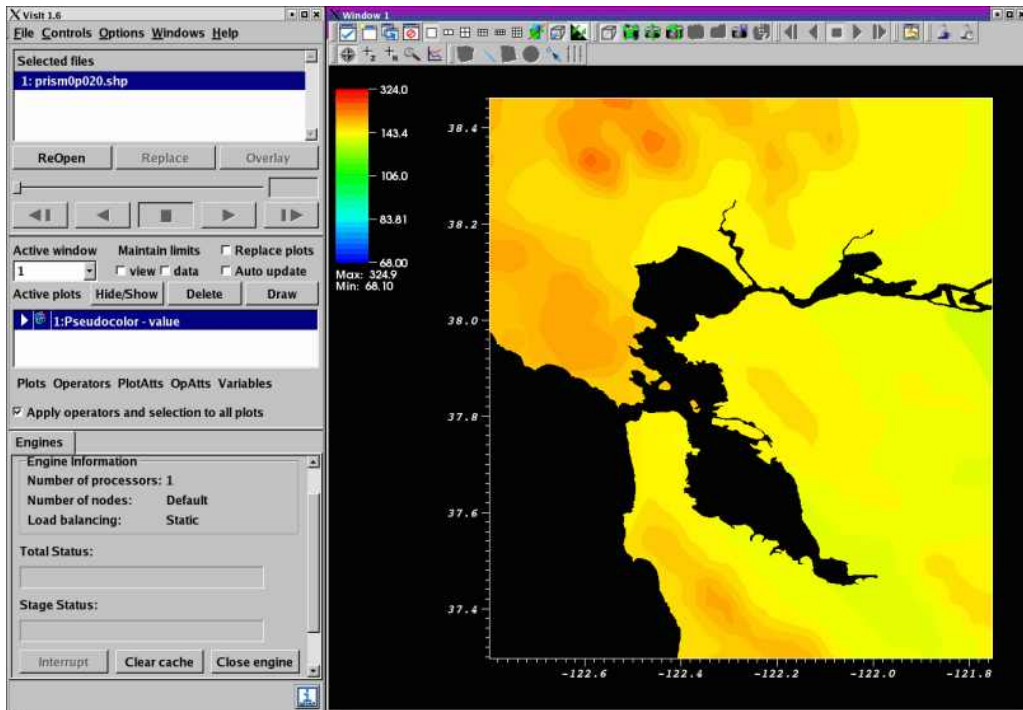


## 다. 다양한 메쉬 데이터 지원

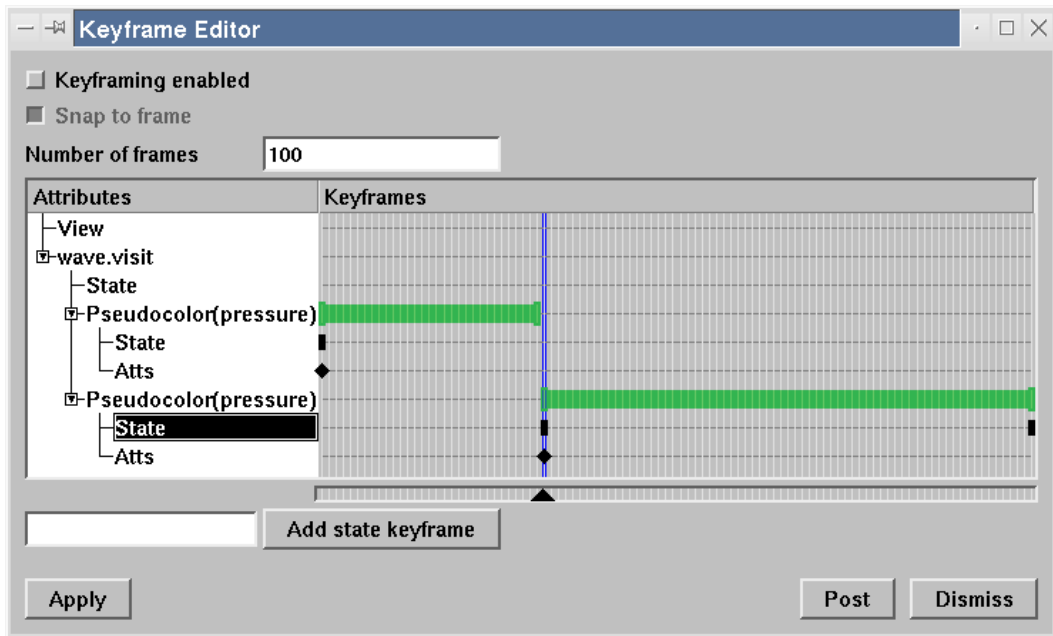
VisIT은 2차원 혹은 3차원 데이터, 포인트, rectilinear, curvilinear 및 unstructured mesh 등, 다양한 메쉬 타입을 모두 지원한다. 또, structured AMR 메쉬 및 CSG 메쉬도 지원한다.

## 라. 강력한 GUI

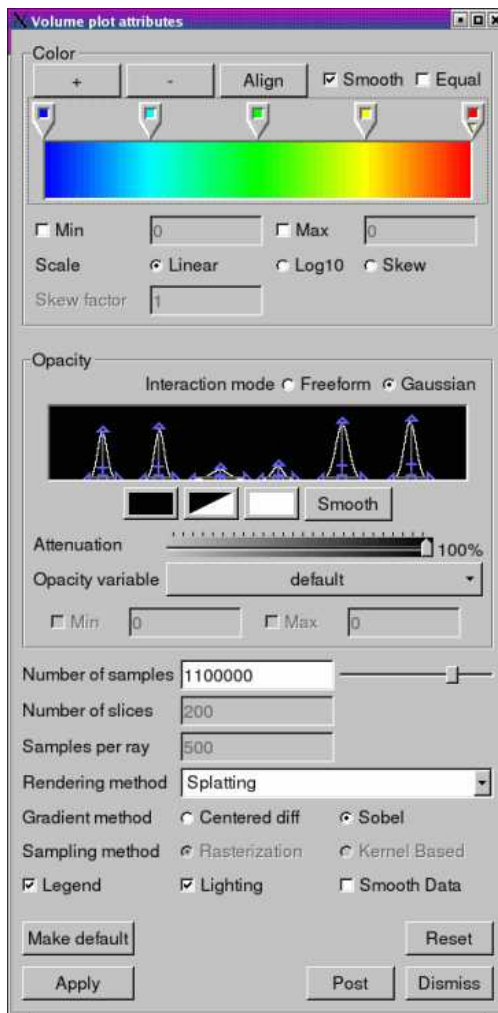
VisIt의 GUI는 매우 강력해서 초보자도 손쉽게 데이터 가시화 작업을 수행할 수 있으며, 이전에 사용해본 경험이 있는 사람들은 한 단계 높은 수준의 기능을 사용할 수 있다. VisIt은 타임스텝이 여러 단계인 데이터셋으로부터 자동으로 애니메이션을 생성할 수도 있으며, 키프레임 애니메이션 기능도 제공한다. 또, 마우스를 이용해서 손쉽게 오브젝트에 pan, zoom, 회전 등의 기능을 수행할 수 있으며 사용자와의 상호작용을 통해 plane이나 sphere같은 오브젝트를 배치할 수도 있다.



<그림 3-10> VisIt의 GUI: 메인 컨트롤 패널과 가시화 윈도우. 이 컨트롤 패널은 파일 선택 영역, active plots 영역 및 어트리뷰트 윈도우를 위한 노트패드 영역으로 구성된다.



<그림 3-11> VisIt의 GUI: 애니메이션 생성에 사용되는 키프레임 윈도우.



<그림 3-12> VisIt의 GUI: volume plot attributes 윈도우. 볼륨 데이터의 transfer function을 설정하는데 사용된다.



---

## 마. 테라급 데이터 가시화를 위한 병렬/분산 구조

VisIt은 대용량 데이터를 실시간으로 처리하기 위한 방법으로 병렬/분산 구조를 채택하고 있다. VisIt의 렌더링 및 데이터 처리 기능은 뷰어 부분과 엔진 부분으로 나눌 수 있으며, 이 구성요소들은 모두 여러 대의 머신에 분산 배치할 수 있다.

뷰어 부분은 렌더링을 수행하는 구성요소로, 일반적으로 로컬 데스크탑이나 가시화 서버에서 실행된다. 이런 이유로 최신 그래픽 카드의 설치에 필수적이라 할 수 있겠다.

엔진 부분은 데이터 처리와 입/출력 부분을 담당하며, 일반적으로 데이터가 위치하고 있는 원격지 머신에서 실행된다. 엔진 부분이 존재함으로써 원시 데이터를 전송할 필요가 없으며, 하이-엔드 컴퓨팅 장치와 입출력 리소스를 사용할 수 있다. 엔진 부분은 단일 프로세서를 이용해서 작업을 순차적으로 실행할 수도 있고, 수천 개의 프로세서를 이용해서 작업을 병렬로 처리할 수도 있다.

## 바. C++, Python 및 Java와의 인터페이스

VisIt은 C++, Python 및 Java 인터페이스를 지원한다. C++과 Java 인터페이스를 이용하면 VisIt의 새로운 사용자 인터페이스를 제공하는 것이 가능하며, 기존의 C++이나 Java 애플리케이션을 이용해서 가시화 기능을 추가로 지원할 수도 있다. Python 스크립트 인터페이스를 이용하면 사용자들이 스크립트 언어를 이용해서 배치 프로세스를 실행하는 것이 가능하다. 이 기능을 이용하면 복잡한 애니메이션 작업을 수행하는 것이 가능하다. 또, Python을 이용하는 시뮬레이션 시스템을 백-플레인으로 활용함으로써 가시화 성능을 시뮬레이션 시스템으로 손쉽게 통합할 수도 있다.

## 사. 동적으로 로딩되는 플러그인

VisIt은 동적으로 로딩되는 플러그인을 통해 확장성을 확보한다. VisIt의 모든 plot, operator 및 데이터베이스 리더들은 모두 플러그인으로 구현되어 있으며, 실행 시간에 플러그인 디렉토리로부터 로딩된다. 플러그인을 새로 추가할 때는 플러그인 디렉토리에 플러그인을 추가 설치하기만 하면 된다.

---

VisIt에는 플러그인 생성용 툴이 제공되며, 이 툴을 이용하면 매우 간단하게 플러그인을 새로 생성할 수 있다. 사용자가 플러그인의 속성에 대해 기술하기만 하면 이 툴은 플러그인을 구현하는데 필요한 대부분의 코드를 자동으로 생성한다. 예를 들어 operator의 경우, 플러그인 생성용 툴은 GUI 애플리뷰트 윈도우에 필요한 코드와 C++, Python 및 Java 인터페이스, 그리고 VisIt과의 인터페이스에 필요한 코드를 모두 생성한다. 사용자는 그저 실제 operation을 수행하는 C++ 코드를 작성하기만 하면 된다.

## 아. 멀티-플랫폼 지원

VisIt은 UNIX(Irix, Tru64, AIX, Linux, Solaris), MacOSX (10.3, 10.4) 및 윈도우즈 플랫폼을 모두 지원한다.

---

## 4. VisIT Challenge

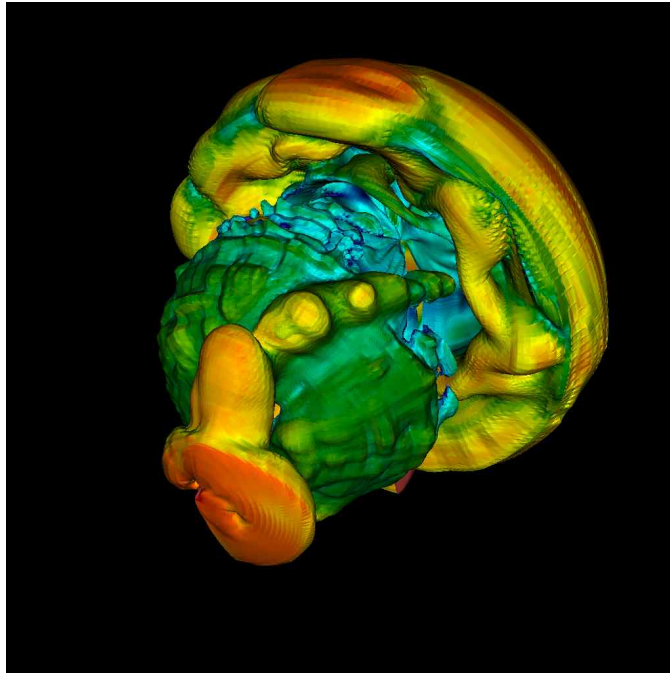
VisIT challenge는 DOE(Department Of Energy)에서 수행한 실험으로, 현재 과학자들이 생성하는 데이터보다 훨씬 더 큰 대용량의 데이터를 미래 수준으로 잡고, 병렬 컴퓨팅 환경에서 처리가 가능할지에 대한 가능성을 타진한 실험이다.

실험 팀은 8,000에서 32,000개의 코어를 사용, VisIT으로 500 billion에서 2trillion zone(grid points)에 이르는 데이터를 가시화했다. 실험에 참가한 기관은 Lawrence Berkeley National Laboratory(Berkeley Lab), Lawrence Livermore National Laboratory(LLNL), 그리고 Oak Ridge National Laboratory(ORNL)이다. 가시화와 실험은 주로 VisIT을 개발한 VACET(Visualization and Analytics Center for Enabling Technology)에서 담당했다.

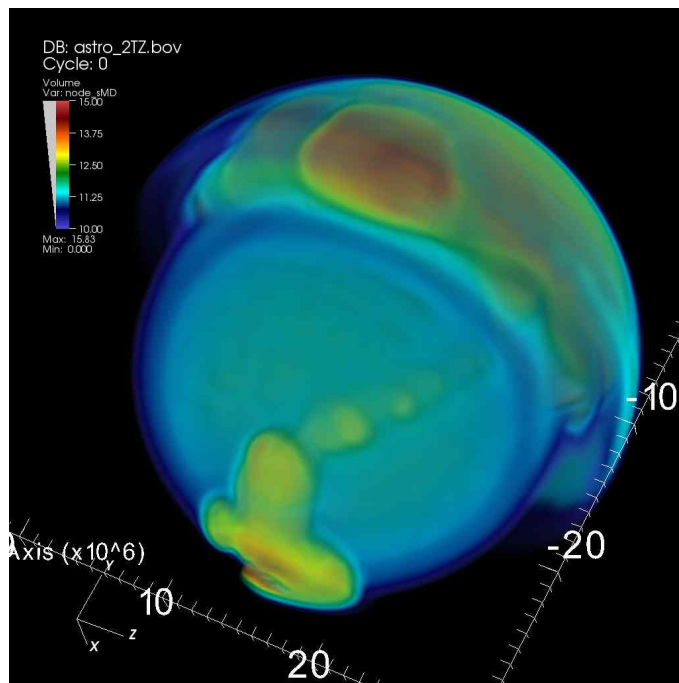
이 실험을 위해 VACET 팀은 천체 물리 시뮬레이션을 통해 데이터를 확보했으며, 샘플링 작업을 통해 5천억 셀(2TB/scalar)에서 2조 셀(8TB/scalar)에 이르는 데이터를 생성해 냈다.

실험은 시스템에 따라 8,000~32,000개의 코어에서 VisIT을 병렬로 실행하는 방식으로 수행됐다. 병렬로 데이터를 로딩하고, 애플리케이션은 isosurface를 생성하고 볼륨 렌더링을 수행하는, 가시화에서 가장 일반적인 이 2가지 작업을 통해 이미지를 생성하는 작업을 수행했다. 이 실험을 통해 실험팀은 잠재적으로 병목 현상을 일으킬 수 있는 시점을 알아내고, VisIT의 다음 버전을 개발하기에 앞서 그 부분을 최적화하는데 필요한 데이터를 수집했다.

1조 셀(4TB/scalar)의 데이터를 가시화하는 데는 16,000개의 코어를 사용했으며, Jaguar와 Franklin 머신에서는 32,000개의 코어를 사용해서 2조 셀의 데이터를 가시화하는 실험을 수행했다. 컴퓨터마다 처리 속도가 조금씩 차이가 나긴 했는데, 대체로 병목이 발생한 부분은 I/O 부분으로, 16,000개의 코어에서 I/O에 2분 이상이 걸렸다. 반면 contouring에는 대략 10초 정도의 시간이, 그리고 렌더링에는 1초~10초 정도의 시간이 소요됐다.



<그림 4-1> JaguarPF(OLCF/ORNL의 Cray XT5)에서 32,000개의 Opteron core를 사용해서 생성한 2TZ 데이터의 isosurface 이미지



<그림 4-2> Franklin(NESRC/LBNL의 Cray XT4)에서 32,000개의 Opteron core를 사용해서 생성한 2TZ 데이터의 볼륨 렌더링 이미지

VisIt Challenge의 결과는 대부분 VisIt Top 15의 순위 내에 들었으며, VisIt Top 15의 순위는 <표 4-1>과 같다.

<표 4-1> VisIt Top 15

Index	Date	Number of zones per timestep	Number of timesteps	Total Size Zones & Bytes	Mesh + Vars	Platform	Visioneers (Visualization-Engineers)
1	2009.10	20,001 <sup>3</sup> (8Tz)	1		rect+1	12000 cpus of Graph (Linux)	Cyrus Harrison
2	2009.06	15,871 <sup>3</sup> (4Tz)	1		rect+1	65536 cpus of Dawn (Blue Gene/P)	Brad Whitlock
3	2009.05	12,596 <sup>3</sup> (2Tz)	1		rect+1	32000 cpus of Jaguar (Cray)	David Pugmire / Sean Ahern
4	2009.06	12,596 <sup>3</sup> (2Tz)	1		rect+1	32000 cpus of Franklin (Cray)	Mark Howison / Prabhathat / Hank Childs
5	2009.04	10,000 <sup>3</sup> (1Tz)	1		rect+1	16000 cpus of Jaguar (Cray)	David Pugmire / Sean Ahern
6	2009.05	10,000 <sup>3</sup> (1Tz)	1		rect+1	16000 cpus of Ranger (Sun Linux)	Hank Childs
7	2009.05	10,000 <sup>3</sup> (1Tz)	1		rect+1	16000 cpus of Juno (Linux)	Hank Childs
8	2009.06	10,000 <sup>3</sup> (1Tz)	1		rect+1	16000 cpus of Franklin (Cray)	Mark Howison / Prabhathat / Hank Childs
9	2009.05	8000 <sup>3</sup> (0.5Tz)	1		rect+1	8000 cpus of Purple (AIX)	Hank Childs
10	2009.03	4000 <sup>3</sup> (64Bz)			rect+1	256 cpus of Ranger (Sun Linux)	Hank Childs / Cyrus Harrison
11	2007.07	3000 <sup>3</sup> (27Bz)			rect+3		Hank Childs
12	2003.05	1152 <sup>3</sup> (1.52Bz)	400+	608Bz	rect+1		Hank Childs / Daniel Laney/ Mark Miller

---

13	2003.05	$720^2 \times 1620$ (0.85Bz)	1000+	850 Bz	rect+1	125 cpus of MCR	Hank Childs / Daniel Laney / Mark Miller
14	2008.11/12	$3600 \times 308^2$ (0.34Bz)	1		rect+4	128 cpus of Franklin (Cray)	Gunther H. Weber / Cameron Geddes
15	2009.03	23.6Mz	3600+	85Bz	point+3		Hank Childs / Jeremy Meredith

Index 1~5까지의 각 실험 결과에 대한 상세 설명은 다음과 같다. 데이터에 대한 별도의 설명이 없는 실험은 모두 VisIt Challenge의 테스트 데이터를 이용한 것이다(<그림 4-1>, <그림 4-2> 참조).

---

## 가. Index 1.

- Date: 2009.10.
- Number of zones / timestep:  $20,001^3(8Tz)$
- Mesh + Vars: rect + 1
- Total Size: approximately 32TB/scalar
- Platform: 12000 CPUs of Graph
- Visoneer: Cyrus Harrison

### 1) 플랫폼: Graph

- 가시화 및 데이터 분석 전용 리소스
- 전체 노드 수: 576
- CPU: AMD Opteron
  - 노드당 CPU 개수: 24 (quad-socket, 6-core Istanbul)
  - 전체 CPU 개수: 13,824
  - CPU 스피드: 2.0GHz
- 메모리
  - 노드당 메모리: 64GB(로그인 노드), 128GB(계산 노드)
  - 전체 메모리: 72,960GB
  - 메모리 어드레싱: 64-bit
- 네트워크: IB 4x
- OS: CHAOS 4.3

### 2) 실험 개요

VisIT Challenge의 일환으로 수행된 실험으로, 12,000개의 코어로 8 Tz의 데이터를 처리.

---

## 나. Index 2.

- Date: 2009.06.
- Number of zones / timestep:  $15,871^3(4Tz)$
- Mesh + Vars: rect + 1
- Total Size: approximately 16TB/scalar
- Platform: 65536 CPUs of Dawn
- Visoneer: Brad Whitlock

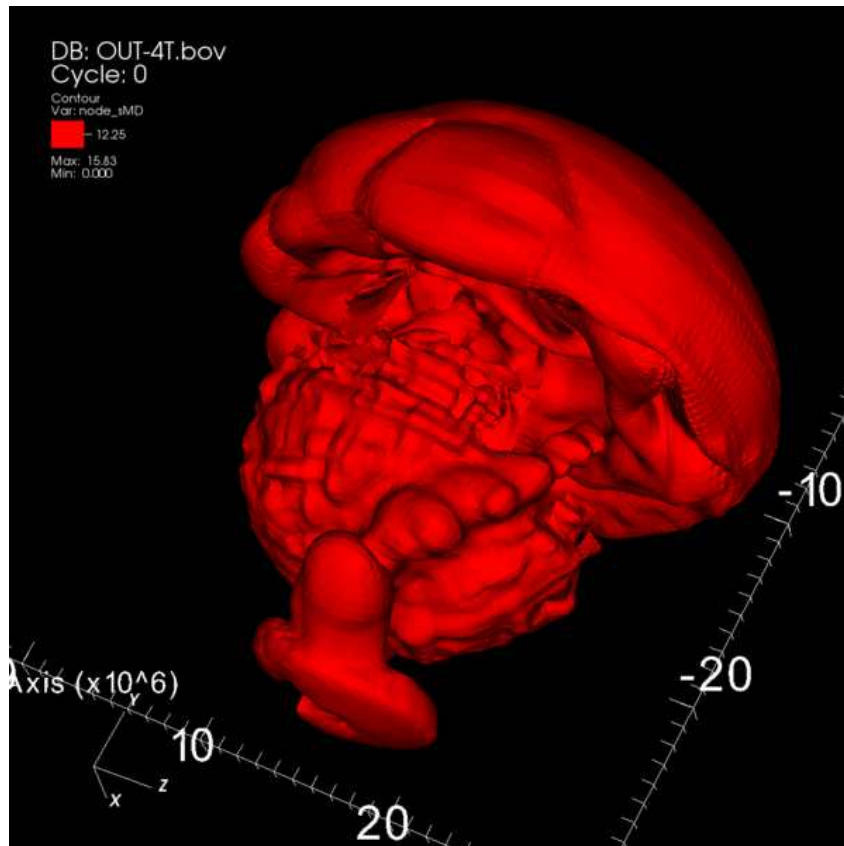
### 1) 플랫폼: Dawn

- LLNL의 Sequoia의 초기 도입분. Sequoia는 2012년에 설치 완료 예정
- 전체 노드 수: 37,166
- CPU
  - 노드당 CPU 개수: 4
  - 전체 CPU 개수: 147,864
  - CPU 스피드: 850MHz
- 메모리
  - 노드당 메모리: 4GB
  - 전체 메모리: 147.5TB
  - 메모리 어드레싱: 32-bit(compute/IO nodes) / 64-bit (front-end nodes)
- 전체 디스크 공간: 2.3 PB
- 네트워크: BlueGene torus and tree
- OS: SuSE SLES 10(login nodes) / Compute Node Krenel(Compute nodes)

### 2) 실험 개요

VisIT Challenge의 일환으로 수행된 실험으로, 65,536개의 코어로 4 Tz의 데이터를 처리, contouring을 수행.





<그림 4-3> Dawn(LLNL의 BlueGene/P)에서 65,536개의 core를  
사용해서 생성한 4TZ 데이터의 contouring plot

---

## 다. Index 3.

- Date: 2009.05.
- Number of zones / timestep:  $12,596^3(2Tz)$
- Mesh + Vars: rect + 1
- Total Size: approximately 8TB/scalar
- Platform: 32000 CPUs of Jaguar (Cray)
- Visoneer: David Pugmire / Sean Ahern

### 1) 플랫폼: Jagure(Cray)

- ORNL Leadership Computing Facility(OLCF)의 주 시스템으로, XT5 부분과 XT4 부분으로 나뉨
- XT5
  - 전체 노드 수: 18,688 노드
  - CPU: dual hex-core AMD Opeteron 2435 (Istanbul), 2.6GHz
  - 노드당 메모리: 16GB
  - peak performance: 2.3 petaflops/s
- XT4
  - 전체 노드 수: 7,832 노드
  - CPU: quad-core AMD Opteron 1354 (Budapest), 2.1GHz
  - 노드당 메모리: 8GB
  - peak performance: 263 teraflops/s
  - 전체 메모리: 147.5TB

---

## 라. Index 4.

- Date: 2009.06.
- Number of zones / timestep:  $12,596^3(2Tz)$
- Mesh + Vars: rect + 1
- Total Size: approximately 8TB/scalar
- Platform: 32000 CPUs of Franklin (Cray)
- Visoneer: Mark Howison / Prabhat / Hank Childs

### 1) 플랫폼: Franklin (Cray XT4)

- NERSC(National Energy Research Scientific COmputer Center)의 MPP(Massively Parallel Processing) 시스템
- 전체 노드 수: 9,572
- CPU: quad-core AMD Opteron (Budapest)
  - 노드당 CPU 개수: 1
  - 전체 CPU 개수: 38,288
  - CPU 스피드: 2.3GHz
- peak performance: 9.2 GFlops/sec per core
- 메모리
  - 노드당 메모리: 8GB
  - 전체 메모리: 78TB
- OS: SuSE Linux (service nodes) / Compute Node Linux(CNL, Compute nodes)

---

## 마. Index 5.

- Date: 2009.04.
- Number of zones / timestep:  $10,000^3(1Tz)$
- Mesh + Vars: rect + 1
- Total Size: approximately 8TB/scalar
- Platform: 16000 CPUs of Jagure (Cray)
- Visoneer: David Pubmire / Sean Ahern

### 1) 플랫폼: Jagure (Cray)

- ORNL Leadership Computing Facility(OLCF)의 주 시스템으로, XT5 부분과 XT4 부분으로 나뉨
- XT5
  - 전체 노드 수: 18,688 노드
  - CPU: dual hex-core AMD Opeteron 2435 (Istanbul), 2.6GHz
  - 노드당 메모리: 16GB
  - peak performance: 2.3 petaflops/s
- XT4
  - 전체 노드 수: 7,832 노드
  - CPU: quad-core AMD Opteron 1354 (Budapest), 2.1GHz
  - 노드당 메모리: 8GB
  - peak performance: 263 teraflops/s
  - 전체 메모리: 147.5TB

---

## 바. Index 15: Time-Varying Data 예제

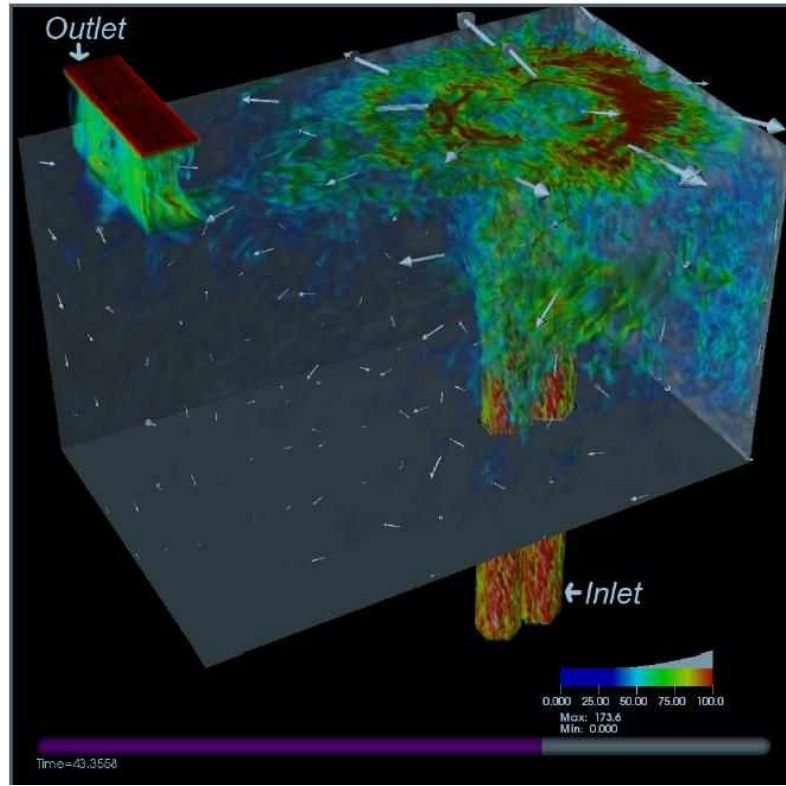
- Date: 2009.03.
- Number of zones / timestep: 23.6Mz
- Mesh + Vars: point + 3
- TimeSteps: 3600+
- Total Size: 85Bz
- Platform: Eureka
- Visoneer: Hank Childs / Jeremy Meredith

### 1) 플랫폼: Eureka

- Argonne National Lab의 가시화/데이터 분석 전용 머신
- 100 dual quad-core servers
- 200 Quadro FX5600 GPUs
- 111 teraflops
- 3.2+ TB RAM

### 2) 개요

“Turbulent Flow of Coolant in an Advanced Nuclear Reactor”로, 시뮬레이션은 ANL의 Intrepid로, 가시화는 ANL의 Eureka로 수행했다. 2009년에 DOE의 OASCR(Office of Advanced Scientific Computing Research) Award를 수상했다.



<그림 4-4> Eureka로 가시화한 Turbulent Flow 데이터

---

## 5. 결론

VisIt은 테라급 이상의 대용량 데이터를 분석, 가시화하는 소프트웨어로, 대용량 데이터를 지원하기 위해 병렬/분산 구조를 기반으로 한다. VisIt은 매우 우수한 기능을 갖춘 선도적인 가시화 응용 프로그램으로 대용량 데이터를 실시간으로 분석하는데 있어서 독보적인 위치를 차지하고 있다. VisIt을 개발한 DOE(Department Of Energy)에서는 VisIt의 성능을 측정하고 미래 데이터의 처리 가능성을 타진하기 위해 VisIt Challenge라는 실험을 수행했는데, 이 실험은 현재 과학자들이 생성하는 데이터보다 훨씬 더 큰 용량의 데이터를 미래 수준으로 잡고 병렬 컴퓨팅 환경에서 처리가 가능할지에 대한 가능성을 타진한 것이다. 이 실험에서 실험팀은 8,000~32,000개의 코어에서 VisIt을 병렬로 수행했으며, 대체로 I/O 부분에서 2분 이상, contouring에는 대략 10초, 그리고 렌더링에서는 1초~10초 정도의 시간이 소요되는 우수한 결과를 확인할 수 있었다(1조 셀의 데이터 기준).

VisIt은 2000년부터 50man/year 정도의 인력을 투입, 개발된 거대 소프트웨어 프로젝트로, 세계 최고 수준의 고성능 컴퓨터 대부분이 사용하고 있는 우수한 가시화 소프트웨어다. 특히 GLOVE 개발의 지표로 삼을만한 선도 사례로 들 수 있겠으며, 향후 GLOVE가 지향하는 방향과 기준을 설정하는 데 있어 중요한 지표가 될 것이다.