



***(usp BigData Oriented Architecture)***

Big Data Software Appliance

Simple, High-Speed Big Data Processing using Shell Scripting  
and uspTukubai

Universal Shell Programming Laboratory, Ltd.

January 2013




# usp BOA

(BigData Oriented Architecture) :  
 A **Software Appliance** dedicated  
 to Big Data Processing based on  
 the Unicage Development  
 Method



- OSS Based
- Scale Out Architecture
- High-Speed Performance
- Inexpensive Hardware
- Maintenance and Support
- Stability

Hadoop 	<ul style="list-style-type: none"> <li>▪ OSS Based</li> <li>▪ Inexpensive H/W</li> <li>▪ Scale Out Architecture</li> </ul>
Appliance Devices	<ul style="list-style-type: none"> <li>▪ High-Speed Performance</li> <li>▪ Maintenance and Support</li> <li>▪ Stability</li> </ul>

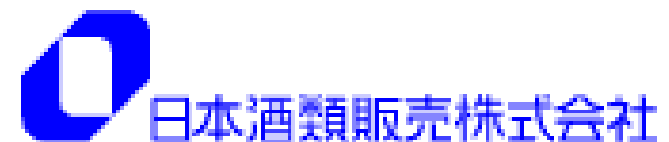
**Unicage Development Method**  
 A development method using shell scripts and text files to cheaply, quickly and flexibly design and implement core enterprise data systems.

# Customers of the Unicage Development Method



Many enterprises have adopted the Unicage Development Method for the following reasons:

1. Fast learning curve for programmers (1 month training + 3 months OTJ)
2. Data Structure is transparent (Structured Text Files)
3. Programs are very short (95% shorter than traditional DB middleware)
4. Runs under latest UNIX/Linux OS to ensure security



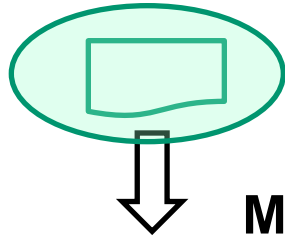
# Big Data Processing Examples



<p>1 Batch Processing (Leading Credit Card Company)</p>	<p><b>Processing of daily transaction details on 60,000,000 credit card accounts</b></p> <p><b>OLD:</b> COBOL Program running on Large Server (15hrs. 29mins)</p> <p>↓</p> <p><b>NEW:</b> UNICAGE Program running on 5 PCs (1hr. 56mins)</p>
<p>2 Complex ETL (Leading Investment Bank)</p>	<p><b>Data Creation for DB Loading of 30,000,000 daily transaction records</b></p> <p><b>OLD:</b> JAVA + PostgreSQL (90 minutes)</p> <p>↓</p> <p><b>NEW:</b> Unicage Program running on 1 PC (91.58 seconds)</p>
<p>3 Complex ETL (Large Electric Utility)</p>	<p><b>Preprocessing of 10GB of Smart Meter data</b></p> <p><b>OLD:</b> JAVA on HPUX Itanium 1.6GHz/2Core (15 hours)</p> <p>↓</p> <p><b>NEW:</b> Unicage Program running on 1 PC (FreeBSD 9.1) (4 mins 16 secs)</p>
<p>4 Large Data Search (Biggest Search Engine in Korea)</p>	<p><b>50.3 Billion Log Records from 5 years (19.2TB)</b></p> <p>10 Types of SQL Searches translated to Unicage</p> <p>Search Time: 0.227 sec - 4.763 sec</p>



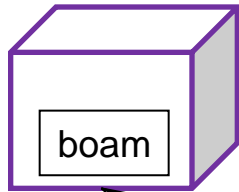
Feature	Description
OSS Base	Runs under any UNIX-based OS (Linux/FreeBSD)
Scale Out	Can be scaled out similar to Hadoop Can increase processing nodes dynamically based on load
High-Speed Processing	High-speed commands (usp Tukubai) and distributed processing Can be used for front-end processing for legacy DB apps and Hadoop
Inexpensive Hardware	Runs on entry-class servers
Maintenance Support	Can be supported by anyone savvy in OS kernel technology
Stability	Only uses the features of UNIX that have been stable for 40 years



**Demo 1: Processing 1 billion records**

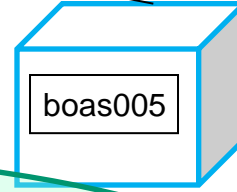
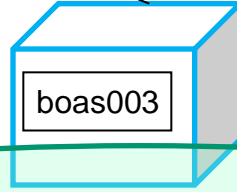
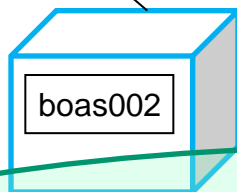
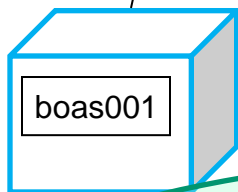
## Master Server

CPU: Xeon E5-2687W 3.10GHz x2  
Memory: 128GB HDD: SATA 1TB



**Shell Script is written only on this server**

10Gbps Network



Optimum data structure is calculated

## Slave Servers

5 PCs

Store massive data

Perform actual processing

CPU: Core i7-3960X  
3.30GHz  
Memory: 32GB  
HDD: SATA 1TB

**Demo 2: Processing 10 billion records (BigData)**

Can scale out easily as the amount of data grows...

Process	Description	Speed
1. Select (para-grep)	Select all records starting with the text "123" from among 1 billion records using 10 parallel processes	3 secs.
2. Sort (clust-qsort)	Sort 1 billion random records in ascending order using 40 parallel processes on 5 slave servers	97 secs.
3. Sum (clust-sm2)	Sum key fields in 1 billion random records using 40 parallel processes on 5 slave servers	35 secs.
4. Mathematical Operations (clust-awk)	Perform mathematical calculations between fields on 1 billion records using 40 parallel processes on 5 slave servers	22 secs.
(clust-lcalc)	Perform precision floating-point operations	67 secs.
5. Join (clust-join1)	Perform a join operation on 1 billion records using 40 parallel processes on 5 slave servers. The master server is relatively small.	37 secs.
6. Complicated Operations (clust-shell)	Distribute 1 billion records by key block units and perform several calculations (key sumup, average, round, literal) using 40 parallel processes on 5 slave servers	17 secs.

Process	Description	Speed
1. Big Data Select (apli-select)	Perform a matching select on 10,000 transactions (join and exclude) from among 10 billion records distributed across the slave servers	4.5 secs.
2. Big Data Update (Add & Change, Delete, Sum) apli-update apli-delete apli-sumup	Update (add & change), delete and sum 10,000 transactions from among 10 billion records distributed across the slave servers	5.5 secs.
3. Big Data Search (apli-search)	Search account holder data based on Rank, Gender, Geographical Region, Age Group, Length of Membership and Minimum Average Score from among 10 billion records distributed across the slave servers	1.2 secs.



## Shell Script Example using uspBOA

When large amounts of data processing are required in the shell script, we use the commands in the “clust” and “para” families.

```
#!/bin/ush -x
#Example script for summing a large data set

clust-join1 slavefile1 key=1 master URE | # Data JOIN (Bigdata)
para-self 10 2/NF-1 | # SELECT (Bigdata)
clust-sm2 slavefile2 1 2 3 4 | # SUM
sm4 1 1 2 2 3 4 | # Intermediate sum
self 1 2 4 3 5 | # SELECT
sm5 1 3 4 4 | # TOTAL
map num=1 | # Transform
sed 's/A/Sales/g' | # Text replace
sed 's/B/Profit/g' |
keta 4 6@NF-1 | # Format columns
comma 4 5 | # Insert commas
cat header -> result # Output with header
exit 0
```

“slavefile” contains the names of the slave servers and the number of parallel processes

# uspBOA Fundamentals

## Unicage Development Method Free Seminar

Date: February 21 (Thurs) 4:30 PM to 6:00 PM (Reception opens 4PM)

Place: USP Labs Seminar Room

Fee: No Charge

(We will stop accepting reservations once the seminar is full so please reserve early)

Reservations →

<http://www.usp-lab.com/seminar1.html>

uspBOA is perfect for the following kinds of customers:

- Those who want to begin Big Data processing
- Those who want to use Big Data processing but the available tools are too expensive
- Those who have implemented Hadoop but are having difficulty utilizing it fully

## Contact Us:

Email: sales@usp-lab.com  
Telephone: +81 3 3432 1174  
Contact: Mr. Kobayashi or Mr. Osakabe

Universal Shell Programming Laboratory, Ltd.  
SS Building 3<sup>rd</sup> Floor, 3-4-2 Nishi Shinbashi, Minato-ku, Tokyo, JAPAN  
<http://www.usp-lab.com>