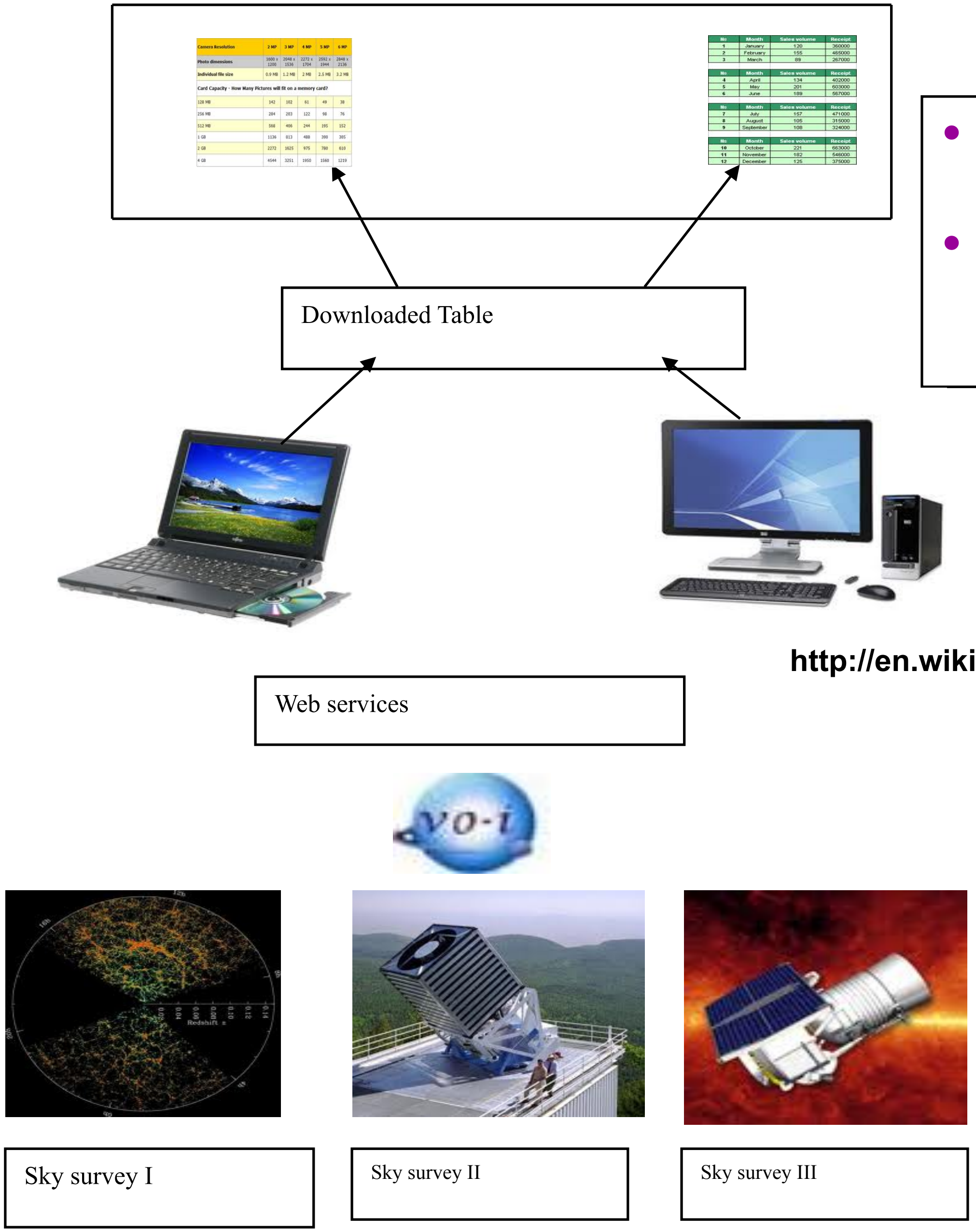
Distributed data mining in accessing the data from VO's



Camera Resolution	2 MP	3 MP	4 MP	5 MP	6 MP
Photo dimensions	1600 x 1200	2048 x 1536	2272 x 1704	2592 x 1944	2848 x 2136
Individual file size	0.9 MB	1.2 MB	2 MB	2.5 MB	3.2 MB
Card Capacity - How Many Pictures will fit on a memory card?					
128 MB	142	102	61	49	38
256 MB	284	203	122	98	76
512 MB	568	406	244	195	152
1 GB	1136	813	488	390	305
2 GB	2272	1625	975	780	610
4 GB	4544	3251	1950	1560	1219

№	Month	Sales volume	Receipt
1	January	120	360000
2	February	155	465000
3	March	89	267000
№	Month	Sales volume	Receipt
4	April	134	402000
5	May	201	603000
6	June	189	567000
№	Month	Sales volume	Receipt
7	July	157	471000
8	August	105	315000
9	September	108	324000
№	Month	Sales volume	Receipt
10	October	221	663000
11	November	182	546000
12	December	125	375000

Downloaded Table

Web services

Sky survey I

Sky survey II

Sky survey III

- All the columns of the downloaded table may not be relevant
- Some of the columns may be redundant .

http://en.wikipedia.org/wiki/Fundamental_plane_(elliptical_galaxies)

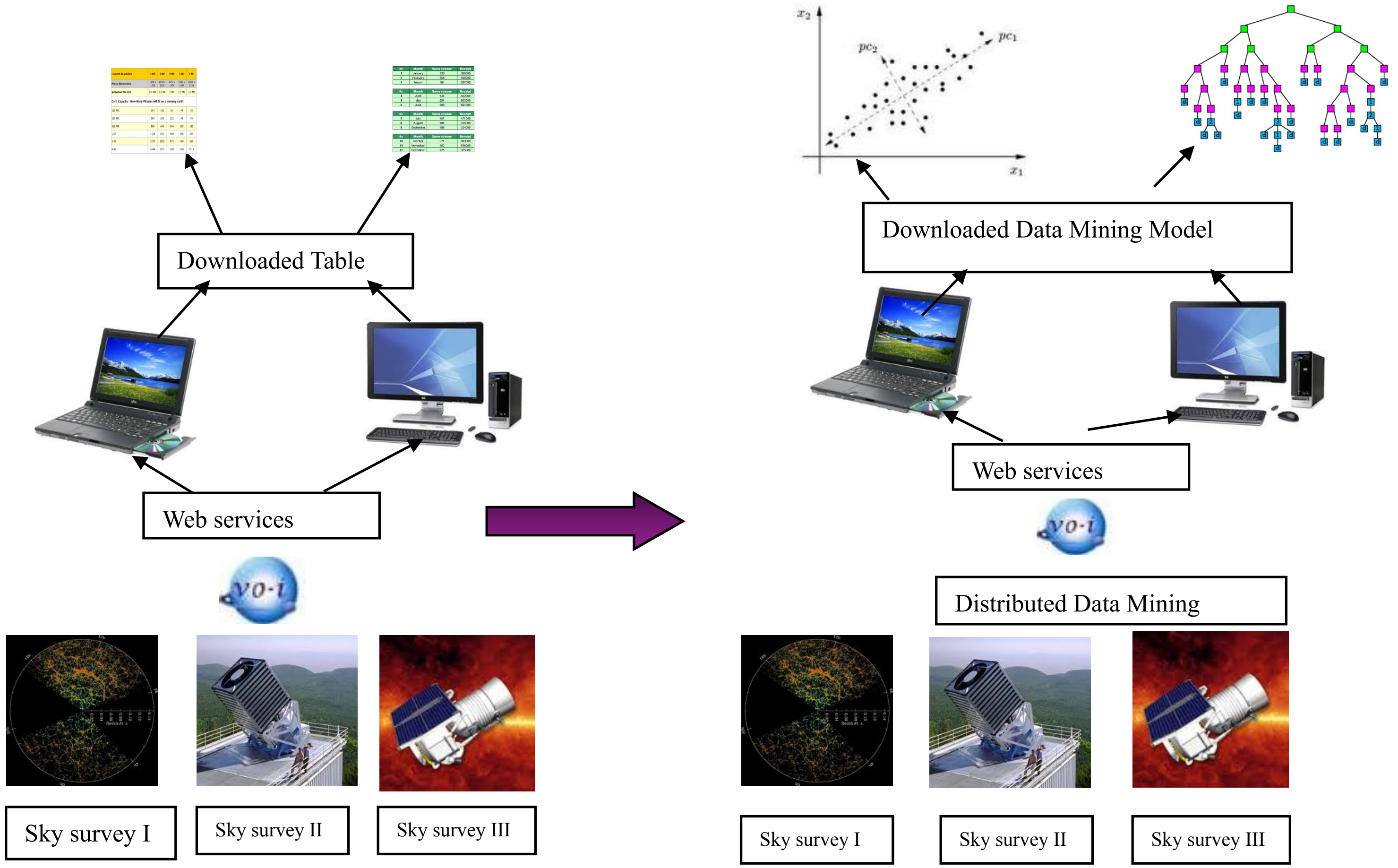
For Eg:

The fundamental plane is a relationship between the effective radius, average surface brightness and central velocity dispersion of normal elliptical galaxies.

What can be done ??

- We can embed the service of filtering the data by applying the data mining algorithms and provide the data mining model instead of raw tables.

- Which needs DDM (Distributed Data Mining) to be carried out without having to down-load large tables to the user.

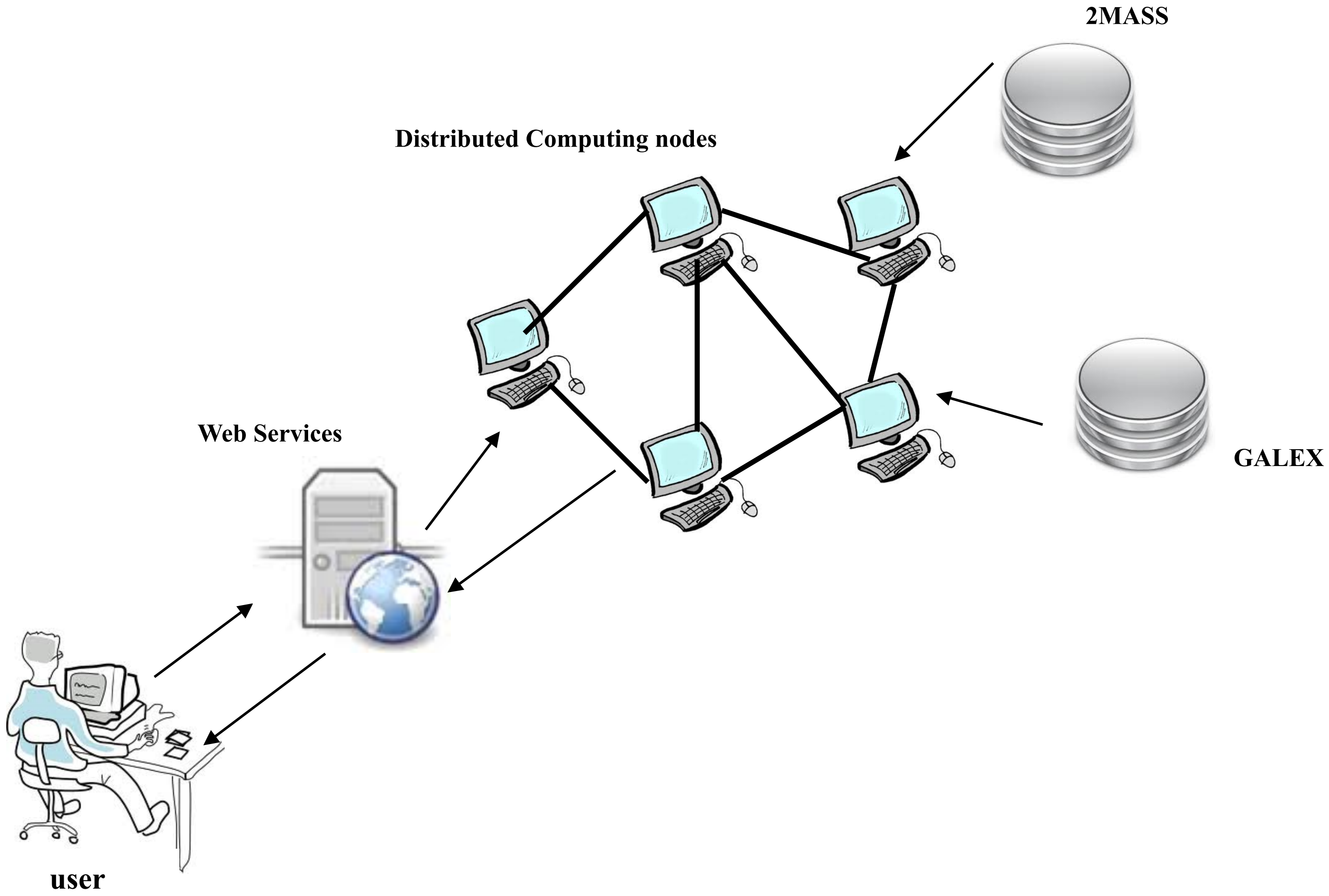


a. Users are getting only raw data

b. Users can get data mining model rather than raw data

rather than raw data

Fig 1 . Distributed Data Mining Data Flow that can be embedded in VO-I


2MASS
Distributed Computing nodes
Web Services
GALEX
user

DDM (Distributed Data Mining)

- DDM strive to analyze the data in a distributed manner without down-loading all the data to a single site.
- DDM is possible in horizontal or vertical partitions .
- In case of horizontal the data is divided among rows, but the number of columns are same at all sites.
- Where as in vertical partition the data is divided among columns ,but the number of rows are same at all sites.
- We considered vertical partition for our implementation .

As an initial step….

- **Reducing the dimension of large high-dimensional data sets will make the analysis efficient .**
- **Reduction of dimensionality using principal component analysis.**
- **PCA can be computed from eigen vectors of covariance matrix.**
- **In our implementation covariance matrix is calculated in a distributed manner.**

Distributed Principal Component Analysis

Problem:

Data are distributed (vertically partitioned) amongst t nodes .

$[X]_{n \times m} = (X_0\, X_1\, X_2\, X_3\, X_4 \ldots\ldots\ldots X_{t-1})$

where Xj resides at node Sj ,
a n × mj matrix , ∑j=1 to t mj = m

Aim:

Compute PCA of X without moving X ($X_0\, X_1\, X_2\, X_3\, X_4 \ldots\ldots\ldots X_{t-1}$) data matrix to a central location such that to avoid the communication and computation bottleneck.

Demonstration with 3 nodes

For example the status of the data is as follows

- node 0 ------x y columns
- node 1 ------z w columns
- node 2 ------l column

•The data need not be centralized like….

$$\begin{bmatrix} x_1 & y_1 & z_1 & w_1 & l_1 \\ x_2 & y_2 & z_2 & w_2 & l_2 \\ x_3 & y_3 & z_3 & w_3 & l_3 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ x_m & y_m & z_m & w_m & l_m \end{bmatrix}$$

To calculate covariance matrix

$$\begin{bmatrix} xx & xy & xz & xw & xl \\ yx & yy & yz & yw & yl \\ zx & zy & zz & zw & zl \\ wx & wy & wz & ww & wl \\ lx & ly & lz & lw & ll \end{bmatrix}$$

The communication b/w 3 nodes

1.sends data to n0

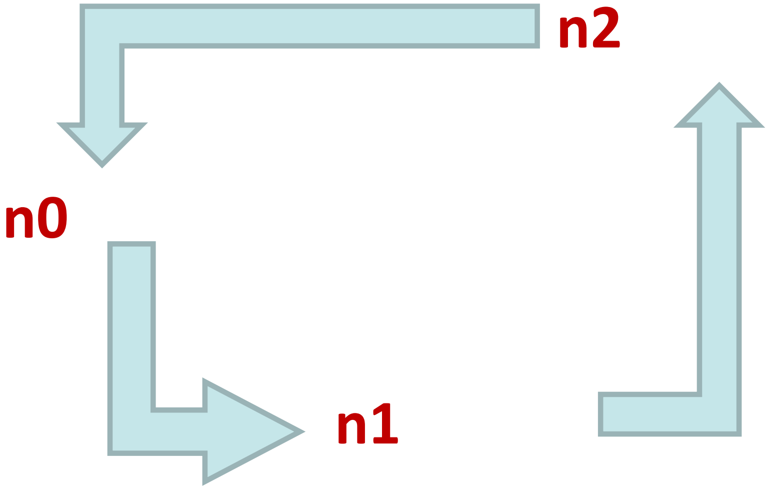
2.Calculates Cov_{n2n2}

3.Calculates Cov_{n1n2}

1.sends data to n2

2.Calculates Cov_{n0n0}

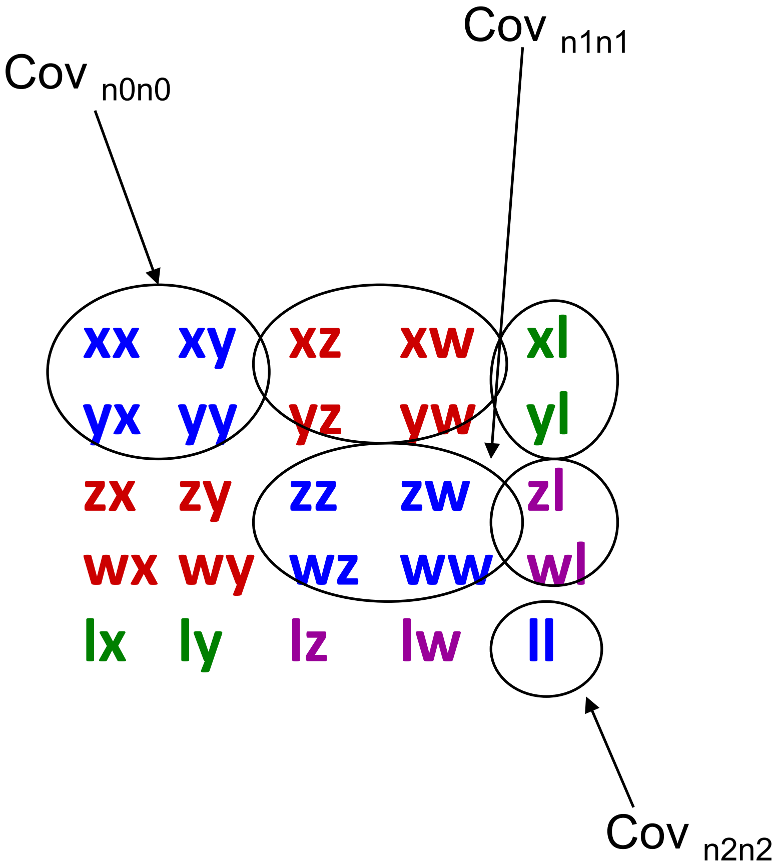
3.Calculates the remaining components of Cov_{n0n2}



1.sends data to n2

2.Calculates Cov_{n1n1}

3.Calculates Cov_{n0n1}



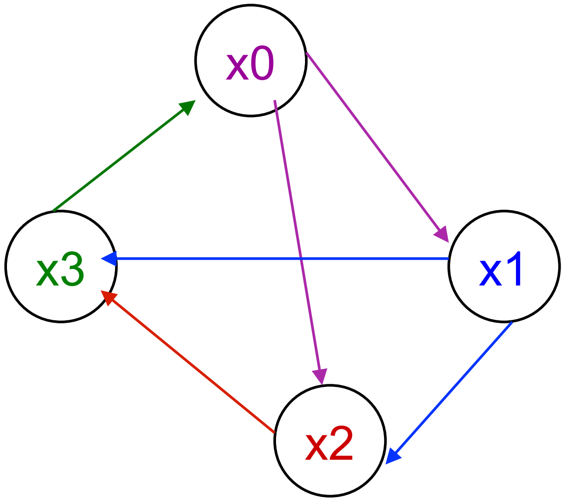
xx	xy	xz	xw	xl
yx	yy	yz	yw	yl
zx	zy	zz	zw	zl
wx	wy	wz	ww	wl
lx	ly	lz	lw	ll

Cov_{n0n0}

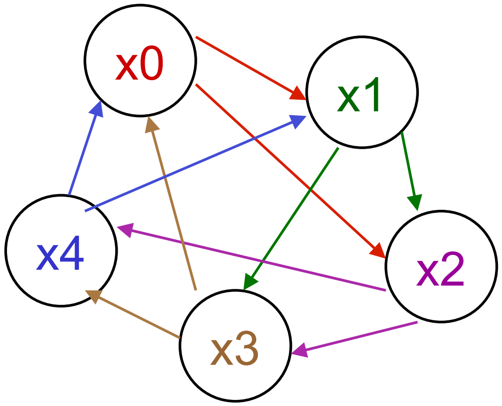
Cov_{n1n1}

Cov_{n2n2}

Generalization with n nodes



If the total number of nodes is even
i.e. t = 2r ,where r>=1
i)send Xj ,where j=0 to r-1 to its r successive nodes
ii)send Xj ,where j=r to 2r-1 to its r-1 successive nodes
iii)Compute Cv(Xj,k) parallel y at Sk



if the total number of sites/nodes is odd
i.e. t= 2r+1 ,where r>=1
i)send Xj,where j=0 to 2r to its r successive nodes
ii)Compute Cv(Xj,k) parallel y at Sk

FeedBack !!