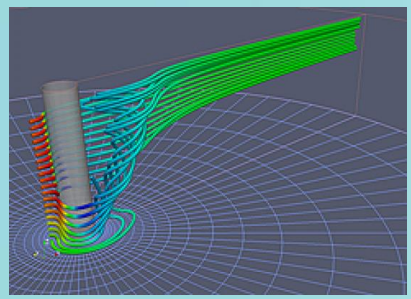
Casillas Ruben

Andrzej Pownuk

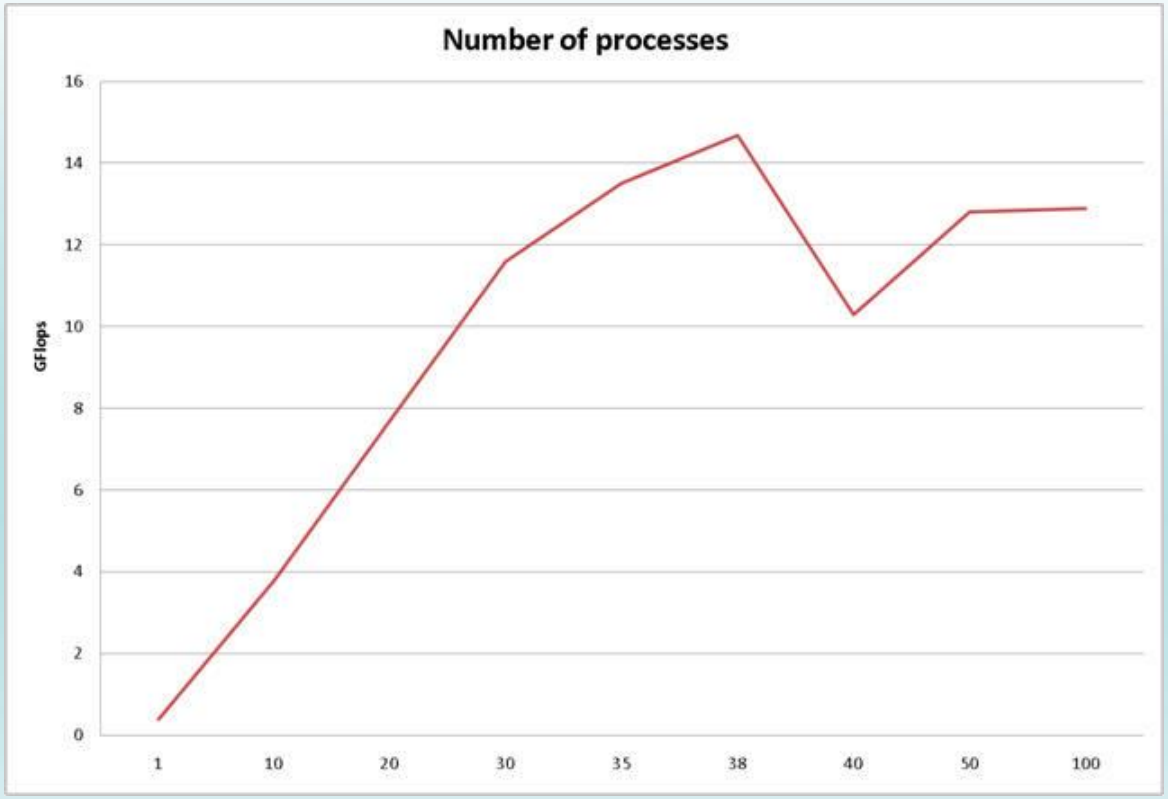


Parallel Web Computing

The University of Texas at El Paso

Parallel computing

- Parallel computing is a form of computation in which many calculations are carried out simultaneously.



Tools

- Compute clusters (MPI library).
- Shared Memory SMP Systems.
- SOA – Service Oriented Architecture (used in the Windows HPC Server 2008).
- Client-server computing.
- PVM (Parallel Virtual Machine)

Advantages of web computing

- Web applications are naturally cross-platform.

- Using web applications it is possible to use computational power of desktop computers with any operating system which are able run web applications (Windows, Mac OS X, Linux, Free BSD, Solaris, HP Unix, etc.).
- It is possible to use a combination of server-side applications and the client side applications.
- It is not necessary to install web applications.

Comparison with traditional approach

- Biggest Computer at UTEP.
  - Sacagawea: a 64-node Penguin Computing Linux cluster.

- Roadrunner - supercomputer built by IBM at the Los Alamos National Laboratory in New Mexico. 12,960 IBM PowerXCell 8i and 6,480 AMD Opteron dual-core processors.

- How many computers are connected to the internet?
  A rough estimate shows 605.6 million computers are connected to the Internet World wide.

SETI@home

- SETI@home is a scientific experiment that uses Internet-connected computers in the Search for Extraterrestrial Intelligence (SETI). You can participate by running a free program that downloads and analyzes radio telescope data.

- http://setiathome.berkeley.edu/

Condor Project http://www.cs.wisc.edu/condor/

- The goal of the Condor® Project is to develop, implement, deploy, and evaluate mechanisms and policies that support High Throughput Computing (HTC) on large collections of distributively owned computing resources. Guided by both the technological and sociological challenges of such a computing environment, the Condor Team has been building software tools that enable scientists and engineers to increase their computing throughput.

Parallel method for calculating definite integrals

$$\int_{\Omega} f(x)\,dx = \sum_{i} \int_{\Omega_i} f(x)\,dx$$

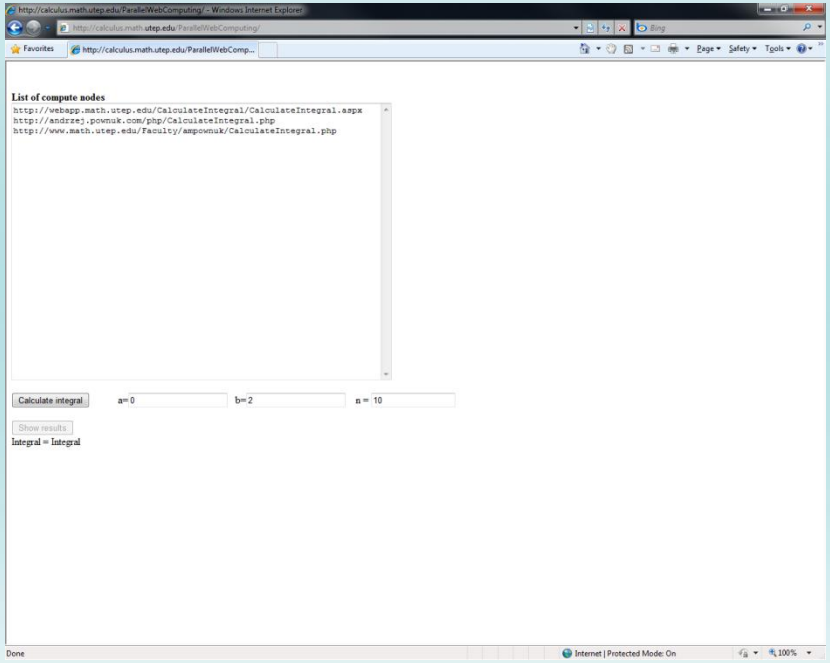
$$\Omega = \bigcup_{i} \Omega_i$$

Each process calculate one integral $\int_{\Omega_i} f(x)\,dx$

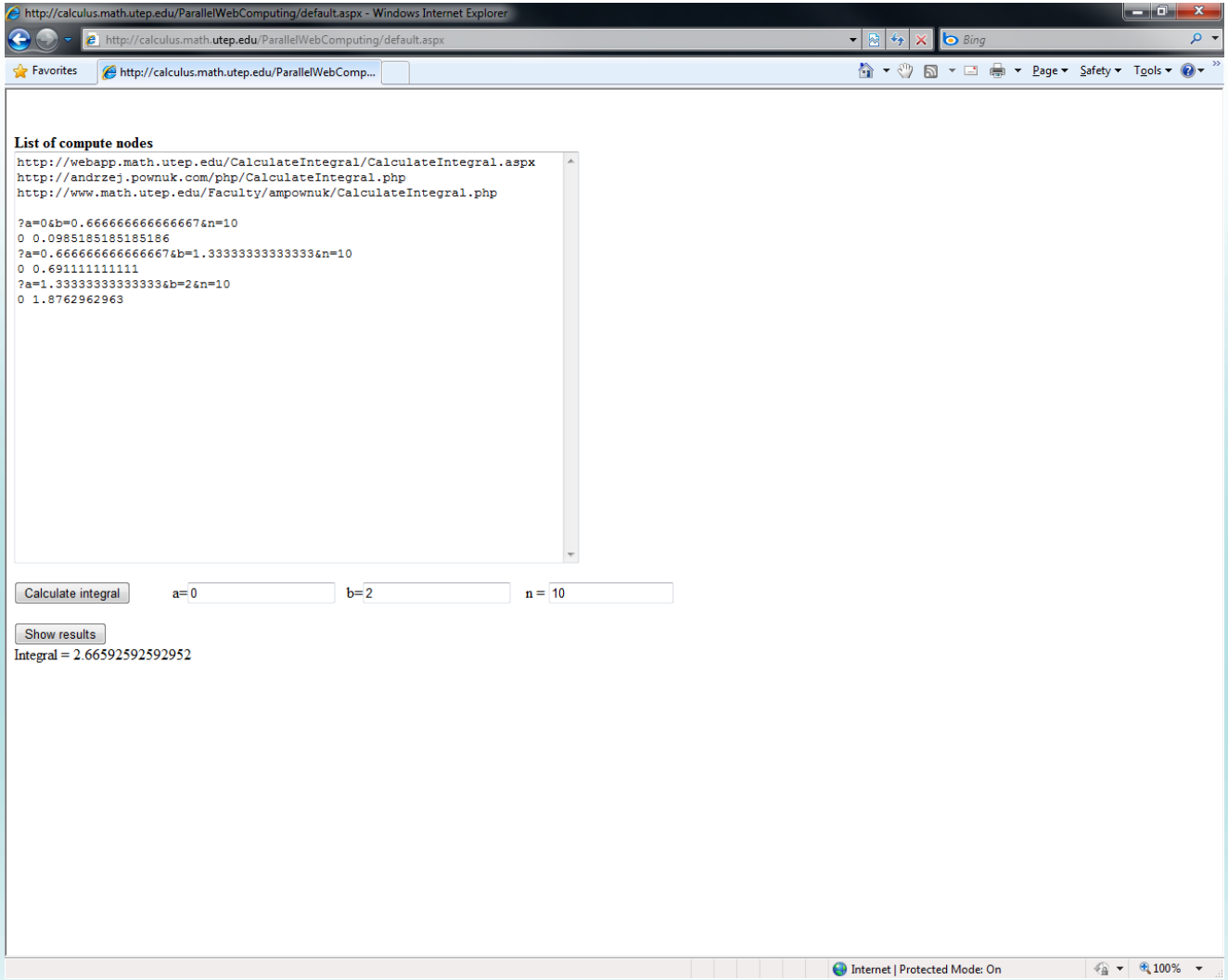
Example 1 – Server side computing

- Calculation of the integral by using server side scripts.

- http://calculus.math.utep.edu/ParallelWebComputing/



Numerical results

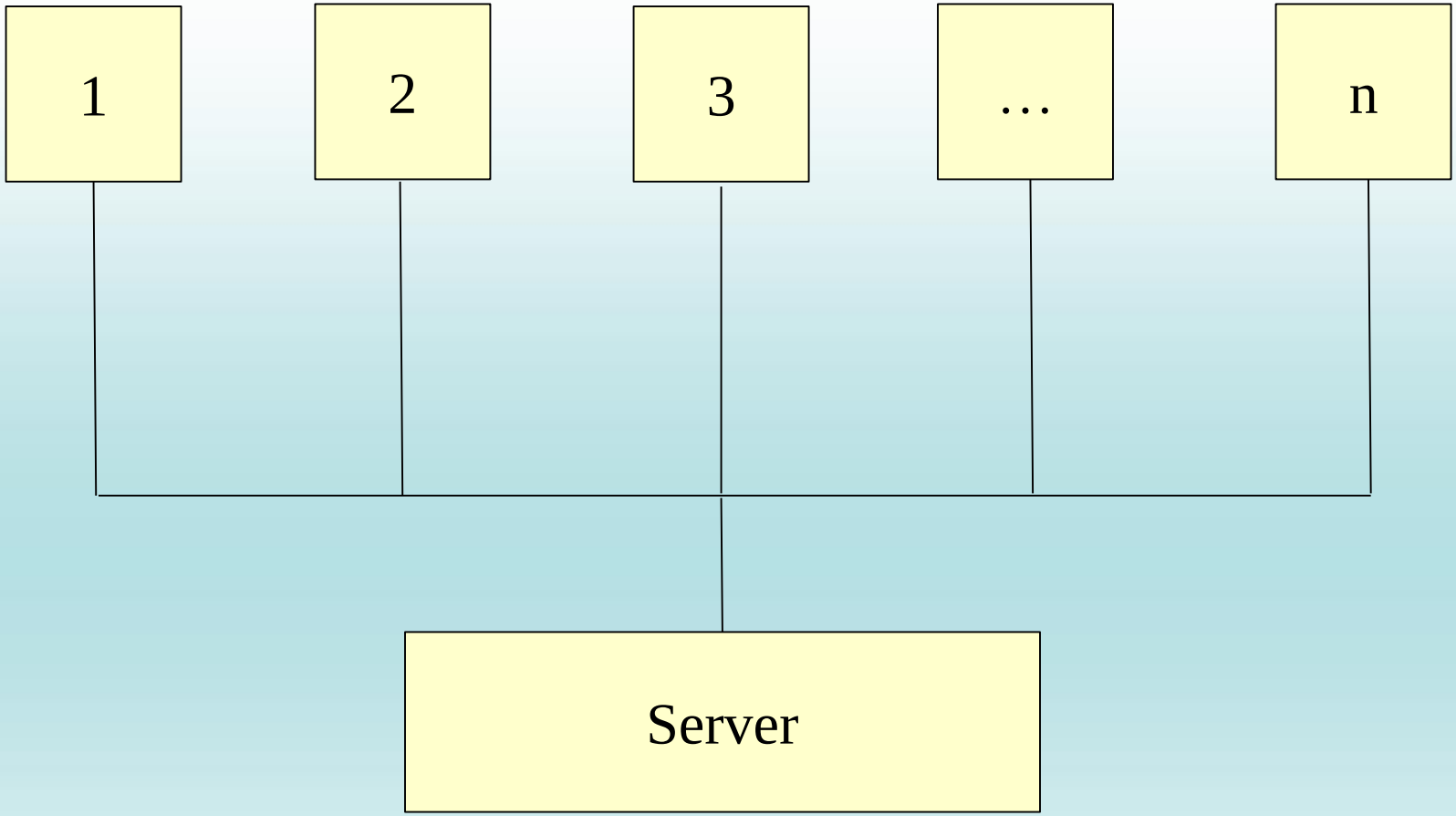


The method can be implemented on different systems

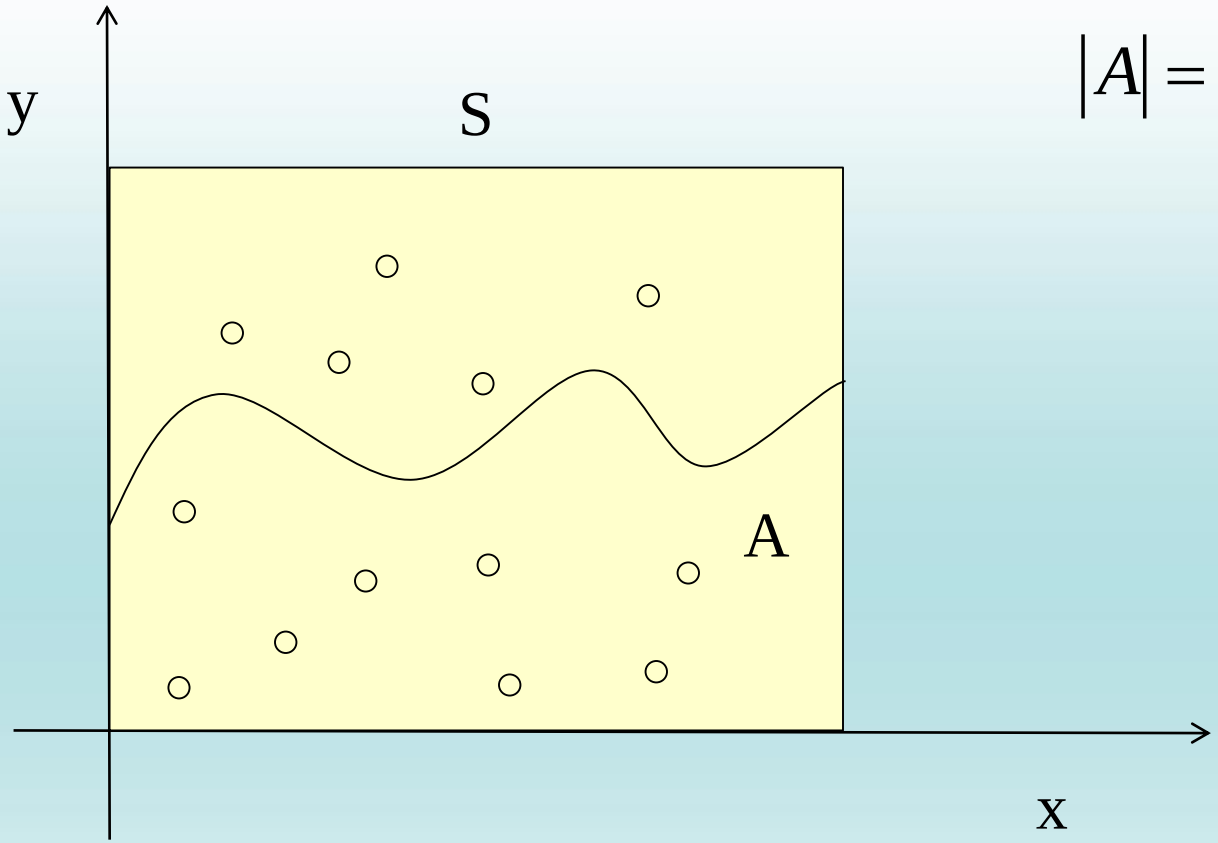
- Examples
  - PHP of Linux
  - ASP.NET on Windows
  - CGI (C,C++)
  - Java, Java scripts, etc.

- Disadvantages
  - Server side computing require installation of the program on the server.
  - It is not possible to use some servers for computational intensive problems.

Parallel computing using java applets (client side computing)



Calculation of the integral by using Monte Carlo simulations



$$|A| = \frac{n_A}{N}|S|$$

Parallel Monte Carlo simulations

$$\frac{n_A}{N} = \frac{n_A^{(1)} + n_A^{(2)} + \ldots + n_A^{(n)}}{N^{(1)} + N^{(2)} + \ldots + N^{(n)}} = \frac{\sum_i n_A^{(i)}}{\sum_i N^{(i)}}$$

Traditional approach (MPI library)

```
int m=10000;
for(int i = 0;i<m;i++)
{
        double x = (double)rand()/(RAND_MAX + 1.0);
        double y = (double)rand()/(RAND_MAX + 1.0);
        if (y>x*x) { mycount++; }
};
MPI_Reduce(&mycount, &count, 1, MPI_INT, MPI_SUM, 0,
MPI_COMM_WORLD);
if(myid==0) {
        double area = (double)count/((double)(numprocs*m));
        std::cout<<"area = "<<area;
}
```

Traditional approach (OpenMP)

```
#pragma omp parallel for private(sum) reduction(+: mycount)
for(ii = 0; ii < n; ii++)
{
        double x = (double)rand()/(RAND_MAX + 1.0);
        double y = (double)rand()/(RAND_MAX + 1.0);

        if (y>x*x) { mycount++; }

}
```

Communications between java applets and the server

- HTTP Request
- Query string
  - http://server?na=100&N=1000

Conclusions

- Using web applications it is possible to create simple distributed applications.
- Web applications are very easy to deploy.
- Web applications are naturally cross-platform.