**Adaptive Distributed Virtual Computing Environment**

*Salim Hariri*

High Performance Distributed Computing Laboratory
Department of Electrical and Computer Engineering
The University of Arizona
Www.ece.arizona.edu/~hpdc/projects/advice

---

**Project Objectives**

- Provide a General Software Development Environment to design and construct large distributed and parallel programs and execute them on a network of heterogeneous computers.
- Environment to develop, run and visualize large distributed and parallel programs over LANs/WANs running on heterogeneous computing resources connected by wired and wireless networks.
- Provide web-based problem solving environment to develop large-scale parallel and distributed applications for C3I applications.

---

**ADViCE Environment**

---
Provided Transparencies by ADViCE

- Access Transparency: The users can access ADViCE from anywhere & anytime.
- Mobile Transparency: The users can use ADViCE while they are moving without interruption of ADViCE services.
- Configuration Transparency: The applications or users do not need to make any adjustment to reflect the changes in the resources allocated to them.
- Fault-Tolerance Transparency: The execution of an application can tolerate failures in the resources allocated to run that application.
- Performance Transparency: The resources allocated to run a given application might change dynamically to improve its performance.
Adaptive Distributed Virtual Computing Environment (ADViCE)

Visualization and Editing Service

- AES (Application Editing Service): Provides the user interface to develop, run and control the parallel / distributed application.
- AVS (Application Visualization Service): Provides visualization of resource information, selected machine information, and execution time.

Application Editing Service

- Provides the user interface to develop, run and control the parallel / distributed application.

Application life cycle control interface

User Authentication Interface

Connect to ADViCE Server

Connect to Application Flow Graph

Edit Application Flow Graph

Interpret Application Flow Graph

Application Life-cycle control interface

Connect
Edit
Run
Stop

Interpret

Adaptive Distributed Virtual Computing Environment (ADViCE) Application Editing Service

Adaptive Distributed Virtual Computing Environment (ADViCE) User Authentication

AES Flow Graph

Application Execution Flow Graph
Adaptive Distributed Virtual Computing Environment (ADViCE)

Application Visualization Service

- Provides visualization of resource information, selected machine information, and execution time.

Visualization of Resource Information
Visualization of Selected Machine Information
Visualization of Execution Time

Resource Information
Selected Machine Information
Execution Time

Adaptive Distributed Virtual Computing Environment (ADViCE)
Adaptive Distributed Virtual Computing Environment (ADViCE)

Application Visualization Service

Execution Time

Control and Management Server (CMS)

- **ARS (Application Resource Service)**: Provides optimization of the allocation of resources to given application tasks.
- **AMS (Application Management Service)**: Utilizes standard management functions to control and manage the execution of applications.
- **ACS (Application Control Service)**: Provides applications with the required services to set up, run, control and manage their execution.
- **ADS (Application Data Service)**: Provides services to establish high-speed communication data paths between the given application tasks.

Application Resource Service

- **Definition**: Assign and reassign tasks to resources to optimize performance, fault tolerance and security.

Scheduler

- Configuration service
- Prediction service
- Mapping service
- Performance
- Fault Tolerance
- Security

Objectives

- Performance
- Fault Tolerance
- Security

APT
Adaptive Distributed Virtual Computing Environment (ADViCE)

**Application Resource Service**

- Application Flow Graph
- Application Resource Service
- Performance
- Fault Tolerance

**Database Server**

- Registered ADViCE Resources

**Resource Constraint Table**

- Resource Status Information

**ADViCE Implementation**

- Implementation Approach
- Experimentation with ADViCE
  - Evaluation of Parallel Algorithm and Configuration
  - Problem Solving Environment
  - C3I Application (Treat Analysis, Tracker,..)

**Using ADViCE as a Parallel Evaluation Tool**

- One can use the ADViCE to experiment and evaluate the performance of different configurations when the number of computers, network type, and problem sizes are changed.
- Compare the time and effort required to perform such tasks with or without using the ADViCE.
Comparing the times required to develop, compile, execute, and visualize a Matrix Multiplication task using p4 and ADViCE for a 1024 x 1024 problem size with 4 nodes.

<table>
<thead>
<tr>
<th></th>
<th>p4</th>
<th>ADViCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Development Time</td>
<td>862 min (431 lines)</td>
<td>2.10 min</td>
</tr>
<tr>
<td>Compilation Time</td>
<td>7.010 sec</td>
<td>0 sec</td>
</tr>
<tr>
<td>Runtime Setup</td>
<td>0.680 sec</td>
<td>0.015 sec</td>
</tr>
<tr>
<td>Task Execution Time</td>
<td>0.194 sec</td>
<td>0.136 sec</td>
</tr>
<tr>
<td>Visualization and Evaluation Time</td>
<td>1.890 sec</td>
<td>0.095 sec</td>
</tr>
</tbody>
</table>

Using ADViCE as a Problem Solving Environment

- Demonstrate how the ADViCE can enable a novice programmer to develop large-scale parallel and distributed applications running on geographically distributed heterogeneous resources.
- ADViCE provides a web-based interface to enable users to develop, configure, execute, and visualize a distributed application in a few minutes.

ADViCE as Problem Solving Env.

- non-ADViCE: The user or team developers need to develop techniques to interact and communicate the modules running on different computers, and they need to develop or integrate techniques to run and manage the execution of the distributed application, as well as collect and visualize the required performance results.
Task Performance Evaluation

- The Linear Equation Solver application has been selected as a running example (sequential algorithm is around 2000 lines of code).
- Consists of an LU decomposition (LU) task, two Matrix Inversion (INV) tasks, and Matrix-vector Multiplication (MULT_V) tasks.
- The environment for this experiment is:
  - 1024 x 1024 matrix
  - 4 nodes of SUN Sparc
  - ATM network.

ADViCE as Problem Solving Env.

Performance Results

- Comparing the timing of several software phases for a Linear Equation Solver application using p4 and ADViCE.

<table>
<thead>
<tr>
<th></th>
<th>p4</th>
<th>ADViCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>838 min</td>
<td>1314 min</td>
</tr>
<tr>
<td>Compilation</td>
<td>6.450 sec</td>
<td>8.100 sec</td>
</tr>
<tr>
<td>Runtime Setup</td>
<td>1.200 sec</td>
<td>1.580 sec</td>
</tr>
<tr>
<td>Task Execution</td>
<td>0.386 sec</td>
<td>0.556 sec</td>
</tr>
<tr>
<td>Total Execution</td>
<td>1.691 sec</td>
<td>1.451 sec</td>
</tr>
<tr>
<td>Visualization</td>
<td>3200 sec</td>
<td>0.140 sec</td>
</tr>
</tbody>
</table>

Development Time (419 lines) (657 lines) (431 lines)
Compilation Time 0.000 sec 0.043 sec
Runtime Setup 1.200 sec 1.580 sec
Task Execution Time 0.386 sec 0.556 sec
Total Execution Time 1.691 sec 1.451 sec
Visualization and Evaluation Time 3200 sec 0.140 sec
Adaptive Distributed Virtual Computing Environment (ADViCE)

Interactions for ADViCE

- Change Detection
- Verification and Analysis
- Adaptation

- Ads
- AMS
- Local ACS
- Local ACS

- Max retransmissions
- Max delay
- Network Link
- Voting
- Task Down
- Machine Crash
- Join/Leave ADViCE components
- Weak signal from users

- Failed experiments
- Long delay experience
- Task Down
- Weak Link
- ADViCE components failure
- Weak signal from users

- Setup new connection
- Rescheduling
- Restart Task
- Reconfiguration
- Migrate components
- Data Compress or Reroute
- Change ADViCE Server URL

- Max. retransmission
- Max. delay
- Time-out
- Answer-Back
- Voting
- Threshold
- Check Requirement Source URL

- Change Detection
- Verification
- Adaptation

Example of Adaptive VDCE (Case: Process failure)

- Detection Phase
- Adaptation Phase
- Analysis Phase

Adaptation of Distributed Virtual Computing Environment (ADViCE)

Example of Fault Tolerance Process Failure