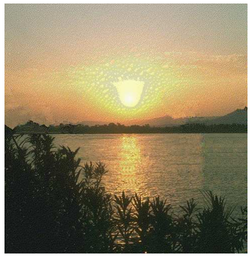
The Nile Project: Replicated CORBA Objects for Scalability and Fault-Tolerance

Michael Ogg, Aleta Ricciardi, Fabio Previato

Department of Electrical & Computer Engineering
University of Texas at Austin



UT Austin, Cornell, UC San Diego, U Florida

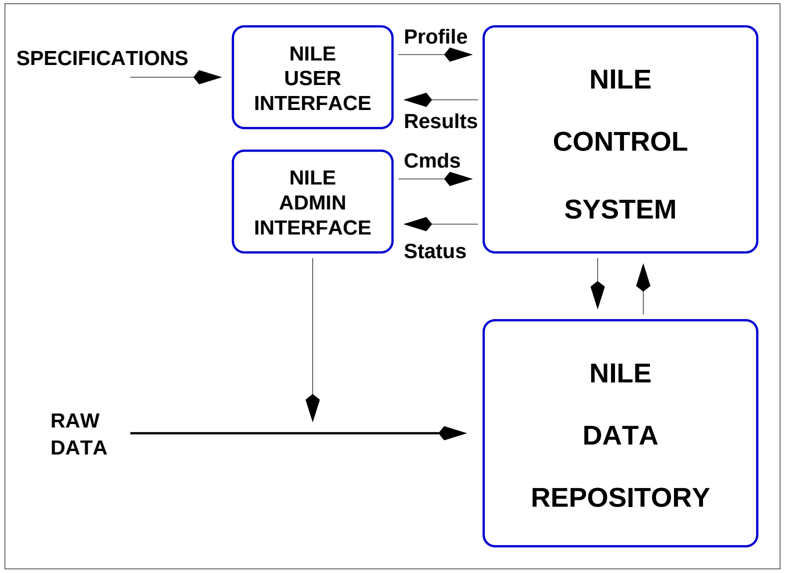
`http://www.nile.utexas.edu/`

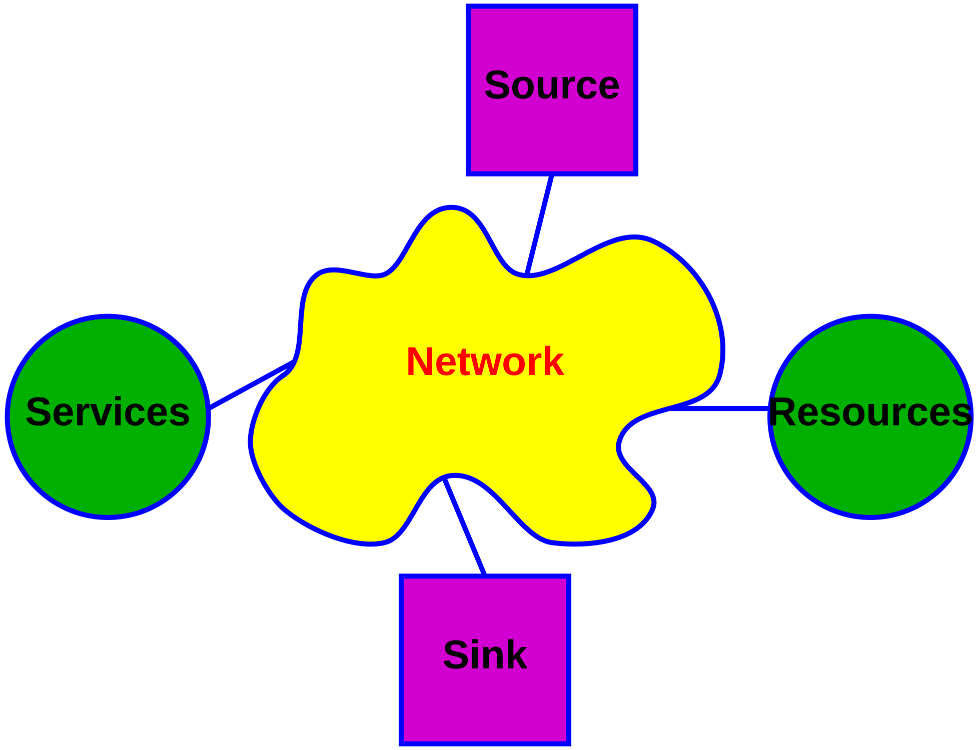
The Nile System

- A scalable, fault-tolerant distributed meta operating system for scientific computing

- Similar to TINA architecture. Nile is representative with well-understood application

- Cluster computing: arbitrarily many, large, distributed clusters.
    - 10s of sites spread across the US
    - 100s of processors/site
    - 100s of TBytes data (at a few sites)





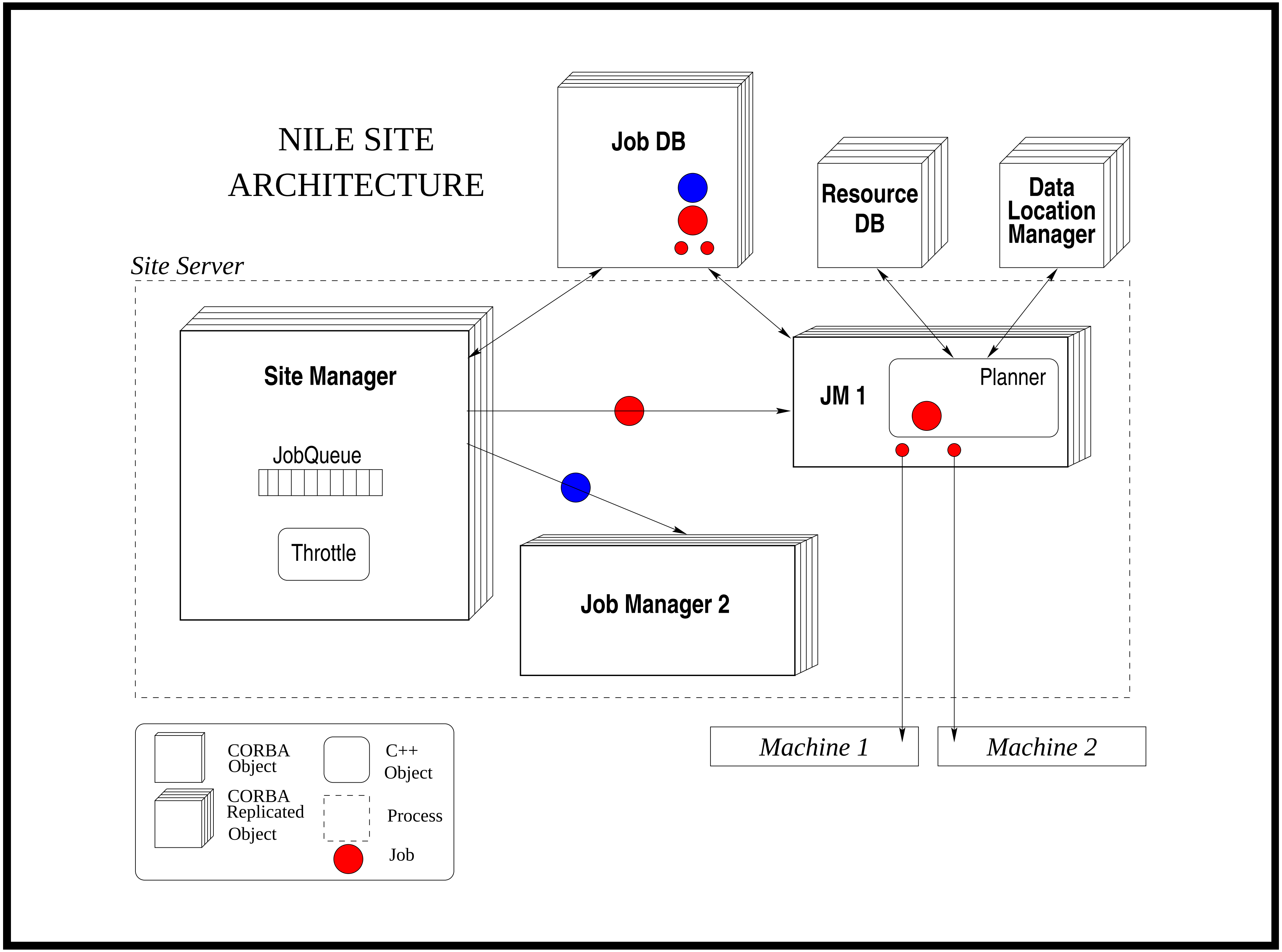
Nile: Source=data; Sink=CPU;

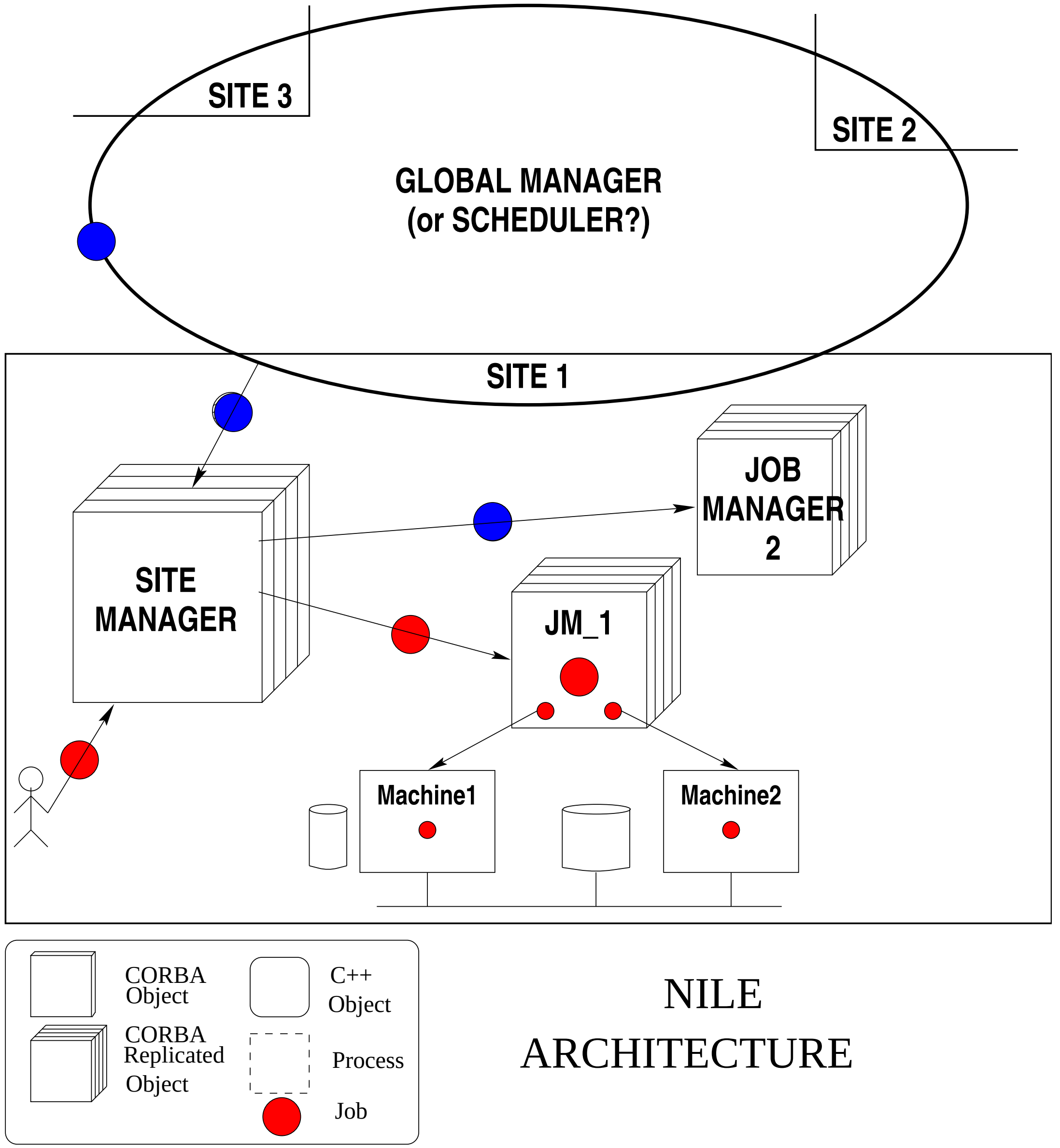
Phone System: Source=called party; Sink=calling party;

Software Architecture

Long-lived, Scalable, Open, Fault-Tolerant, Wide-Area Operating System

- CORBA: longevity, standards, interfaces, maintenance, openness
- Fault-tolerant: as system size increases, failures *will* occur
- Electra: fault-tolerant ORB
  - v2.0-compliant ORB that supports replicated objects
  - built on Isis/Horus f-t transport system (will use others too)
    f-t groups & multicasts, virtual synchrony
- Wide-Area scheduling of remote resources


NILE SITE
ARCHITECTURE
Job DB
Resource
DB
Data
Location
Manager
Site Server
Site Manager
JobQueue
Throttle
JM 1
Planner
Job Manager 2
Machine 1
Machine 2
CORBA
Object
C++
Object
CORBA
Replicated
Object
Process
Job



NILE
ARCHITECTURE

Java, CORBA, IIOP

- CORBA v2.0 allows ORB inter-operability via GIOP and IIOP
- Not *quite* working yet in Electra (any day now)
- So – we have written "poor man's IIOP" for Java:
  - Java IDL compiler to generate classes
  - serialize messages to ASCII string in Java client
  - use sockets to send to C++ proxy
  - proxy translates, then makes Electra method calls
- For IIOP: message contains target IP address, so have to "know" service is fault-tolerant
- more generally, there are replicated object subtleties, c.f. `http://lsewww.epfl.ch/ersads/`

Hardware

Commodity Hardware: Affordable, Modular, Scalable

Development work done first on Sparc (SunOS), now almost everything on PCs with Linux.

- PPro for scalability, cost, and flexibility
  - Existing farms of Digital AXP at Cornell, Florida, Minnesota
  - Plans for PPro farms at UT, Cornell, ...
- 100 Mbps Ethernet (switches and hubs)
- At any farm, disk for ≥ 100 GB data
- ATM for remote connections?

Networking and Data

Goal is transparent, on-demand access to all data. Options:

- Move computations to data (heterogeneity issues solved, insufficient CPU near data)
- Move data to computations (insufficient networking)
- Co-locate all resources (not viable)
- Replicate resources (infeasible)

Nile Status: March 1997

Feature	Status	Ready
NCS IDLs/Objects	done	
NCS heterogeneity	done	
scalability	testing	
Java client (proxy)	done	
Java client (IIOP)	waiting	Spring '97
Data Repository (1)	done (legacy)	
Data Repository (2)	specifications	late '97
WAN design	hooks exist	Summer '97 on
WAN scheduling	algorithms	Summer '97 on

Nile Demonstration

- Demonstration has Nile running on network of 2 Linux laptops
- `SiteManager` objects are replicated on `dhuez` and `forclaz`
- `SubJobs` run on `dhuez` and `forclaz`
- Data served from `forclaz`
- Web browser runs on `forclaz`