Research Strategy Workshop
MPI for Software Systems

Jayadev Misra

Department of Computer Science
University of Texas at Austin

Sept. 23, 2005

The Big Picture

- Managerial
- Sociological
- Technical

Managerial Issues

- Who to hire.
- How to evaluate performance.
- How to motivate: Incentive structure.
- How much top-down control.
- . . .

Sociological

- Does it benefit the Society, Germany?
- Relevance of the research for current practices.
- How to do liaison with industry/ application builders
  technology transfer in both directions
- Education: Influence research at universities. Introduce co-supervision of students, internship programs.
- . . .

Technical: Research driven by application domain

- Embedded and Real-time systems
  Automotive, Avionics, Industrial Process Control
- Graphics and Visualization
- Global Warming
- Collaborative Computing
- Business Process Workflow.
- . . .

Liaison with industry may identify intellectually stimulating problems.

Technical: Research driven by Tool Building

- Evaluate designs
- Simulate designs
- Spot potential disasters. . . .
- . . .

Technical: Research driven by Grand Challenges

Courtesy Tony Hoare

Typical Grand Challenges

Prove Fermat's last theorem	(accomplished)
Put a man on the moon	(accomplished)
Cure cancer within ten years	(failed in 1970s)
Map the Human Genome	(accomplished)
Map the Human Proteome	(too difficult now)
Find the Higgs boson	(in progress)
Find Gravity waves	(in progress)
Unify the four forces of Physics	(in progress)
Hilbert's program for math foundations	(abandoned 1930s)

Typical Grand Challenges In Computing Science

Prove that P is not equal to NP	(outstanding)
The Turing test	(outstanding)
The verifying compiler	(abandoned in 1970s)
A championship chess program	(completed 1997)
A GO program at professional standard	(too hard)
Machine translation of English to Russian	(failed in 1960s)

A Typical Grand Challenge Project

It offers fundamental and radical advance In basic Science or Engineering.

- Is a fifteen-year project,
- With world-wide participation,
- And clear evaluation of success or failure.

A Grand Challenge Project needs

- Maturity of the state of the art,
- General support from the international scientific community,
- Long-term commitment from the teams who engage in it,
- Understanding from funding agencies.

Criteria

- Fundamental: How does it work? Why does it work?
- Historical: formulated long ago.
- Astonishing: Currently beyond our reach.
- Idealistic: does not duplicate commercially motivated evolution of existing products.

A Grand Challenge is not ...

- A solution to all problems
- A science-fiction scenario
- A list of open questions
- A roadmap of strategic directions
- A call for proposals
- A specially funded programme of research
- A plan for a commercial product
- A promise of competitive advantage

Two Grand Problems in Software Engineering

- Robustness
- Security

Some Personal Grand Challenges

- Design Theory for Middleware
- Safe Design/ Integration Principles
  - Checked automatically for safety
  - UML designs have inspired practitioners, not academics

Orchestrate Services: In collaboration with William Cook

- Contact two airlines simultaneously for price quotes.
- Buy ticket from either airline if its quote is at most $300.
- Buy the cheapest ticket if both quotes are above $300.
- Buy any ticket if the other airline does not provide a timely quote.
- Notify client if neither airline provides a timely quote.

Example: workflow

- An office assistant contacts a potential visitor.
- The visitor responds, sends the date of her visit.
- The assistant books an airline ticket and
  contacts two hotels for reservation.
- After hearing from the airline and any of the hotels:
  he tells the visitor about the airline and the hotel.
- The visitor sends a confirmation which the assistant notes.

Example: workflow, contd.

After receiving the confirmation, the assistant

- confirms hotel and airline reservations.
- reserves a room for the lecture.
- announces the lecture by posting it at a web-site.
- requests a technician to check the equipment in the room.

Wide-area Computing

Acquire data from remote services.

Calculate with these data.

Invoke yet other remote services with the results.

Additionally

Invoke alternate services for failure tolerance.

Repeatedly poll a service.

Ask a service to notify the user when it acquires the appropriate data.

Download an application and invoke it locally.

Have a service call another service on behalf of the user.

The Nature of Distributed Applications

Three major components in distributed applications:

Persistent storage management

databases managed by the airlines and the hotels.

Specification of sequential computational logic

does ticket price exceed $300?

Methods for orchestrating the computations

contact the visitor for a second time only after hearing from the airline and one of the hotels.

We look at only the third problem.

Site

Compose basic computing elements called Sites. A site is a

- function: Compress MPEG file
- method of an object: LogOn procedure at a bank
- monitor procedure: read from a buffer
- web service: CNN, get a stock quote
- transaction: check account balance
- distributed transaction: move money from one bank to another
- Humans: Send email, expect report

Orc

An Orc expression is

1. Simple: just a site call, or
2. composition of two Orc expressions

Evaluation of Orc expression:

calls some sites,

publishes some values

Simple Orc Expression

$CNN(d)$

calls site CNN,

publishes the value, if any, returned by the site.

Composition Operators

do f and g in parallel	$f \mid g$	Symmetric composition
for all x from f do g	$f >x> g$	Sequencing
for some x from f do g	$g \text{ where } x{:}\in f$	Asymmetric composition

Orc

- Structured Distributed programming
- No special constructs for:
  thread, process, channel, communication, synchronization
- or for: time-out, priority, interrupt, arbitration, failure
- Strong theoretical foundation
- Modularity: Encouraging results on large applications

Program Structuring: Running an Auction

- Advertise the item and a minimum bid price v: call $Adv(v)$
- Get bids: $Bids(v)$ returns a stream of increasing bids, all above v.
- Post successive bids at a web site by calling $PostNext$

$$Auction_1(v) \;\underline{\Delta}$$
$$Adv(v) \gg Bids(v) \;>u>\; PostNext(u) \gg 0$$

Program $Bids$

Get the next bid exceeding v.
Assume that bidders put their bids on channel c.

$$
\begin{array}{l}
nextBid(v) \ \underline{\Delta} \\
\qquad c.get \\
\quad > x > \\
\qquad (\quad if(x{>}v) \ \gg \ let(x) \\
\qquad\ \ \mid if(x{\leq}v) \ \gg \ nextBid(v) \\
\qquad)
\end{array}
$$

Output successively increasing bids, all above v.

$$Bids(v) \ \underline{\Delta} \ nextBid(v) \ > u > (let(u) \mid Bids(u))$$

A Terminating Auction

- Terminate if no higher bid arrives for an hour (h time units).
- Post the winning bid by calling $PostFinal$.
- Return the value of the winning bid.

$Tbids(v)$ returns pairs (x, b): $b \Rightarrow x > v$, $\neg b \Rightarrow x = v$

$$
\begin{array}{l}
Auction_2(v) \ \underline{\Delta} \\
\qquad\qquad Adv(v) \\
\qquad\qquad Tbids(v) \\
\quad \gg \\
\quad >(x,b)> \\
\qquad\qquad (\quad if(b) \quad \gg PostNext(x) \quad \gg 0 \\
\qquad\qquad \ \mid \quad if(\neg b) \quad \gg PostFinal(x) \quad \gg let(x) \\
\qquad\qquad)
\end{array}
$$

$Tbids$

$Tbids(v)$ returns a stream of pairs (x, b):
x is a bid, $x \geq v$, and b is boolean.

$b \quad \Rightarrow x$ exceeds the previous bid
$\neg b \quad \Rightarrow x$ equals the previous bid,
i.e., no higher bid has been received in an hour.

$$Tbids(v) \ \underline{\Delta}$$
$$let(x, b) \mid if(b) \ \gg \ Tbids(x)$$
where
$$(x, b) :\in \ nextBid(v) \ >u> \ let(u, \textit{true})$$
$$\mid \ Rtimer(h) \ \gg \ let(v, \textit{false})$$