Remote Execution to Save Local Energy

Compute locally on battery power

vs

Transmit to wired compute server +
Idle in low power mode + Receive results

- Why and when it works?
- How? The mechanisms required.

Effectiveness?
(Rudenko et al)

- Identical, dedicated laptops
- Wireless WaveLAN send 3W, receive 1.48 W, sleep .18W, 2MB/s

- Li ion battery power, measured by APM metric
- No interference
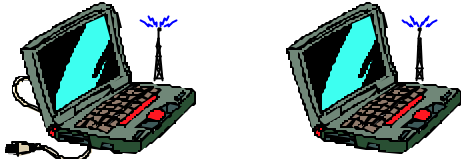- Display and disk timeouts of 1 min

Compilation

- Both have replicas of source code to be compiled
- Sends back stripped executables

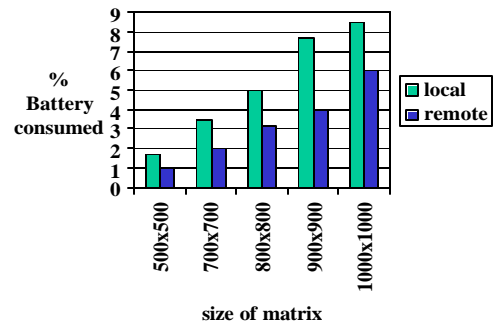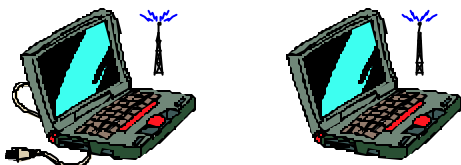- Send modifications only

Compilation Results

% Battery consumed

KB of changed code

local
remote
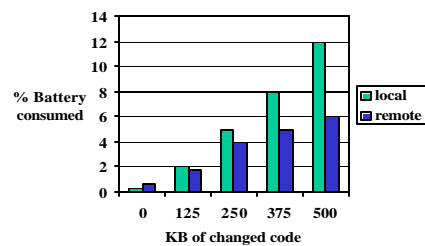
Gaussian Solution of System of Equations

- Little I/O, little VM paging
- Solution vector returned

- Entire matrix shipped

Gaussian Results

% Battery consumed vs size of matrix

- local
- remote

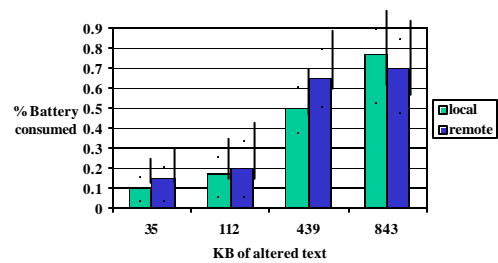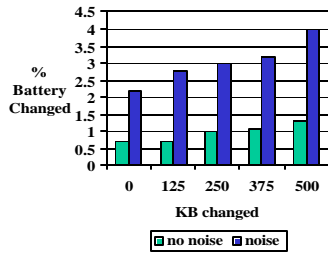Categories: 500x500, 700x700, 800x800, 900x900, 1000x1000

Latex

- Both have replicas of source to be formatted
- Sends back formatted document

- Send altered text only
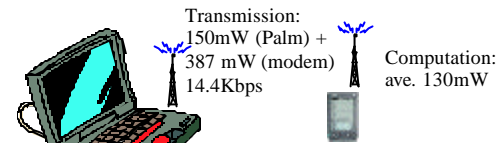- Application with heavy I/O

Latex Results

% Battery consumed vs KB of altered text

- local
- remote

Categories: 35, 112, 439, 843

With Noise

A second pair of machines with saturated socket noise resulting in significant backoff and retransmit

CRA-W Summer Project '99
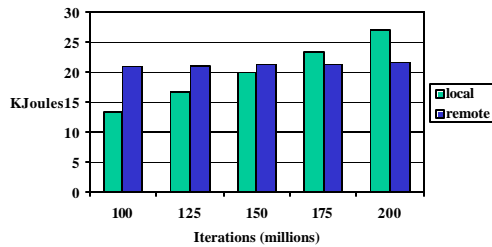

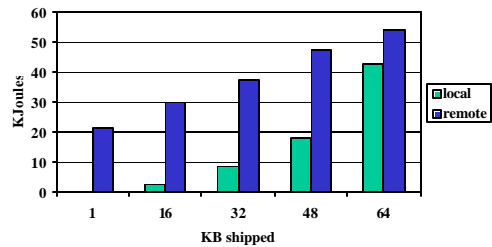Transmission: 150mW (Palm) + 387 mW (modem) 14.4Kbps

Computation: ave. 130mW

- D'Agents server
- Fast server machine (but interpretation of Tcl)
- Tcl agents "parked" on the Palm; PalmOS C program of similar functionality
- PilotLauncher - to ship code and data

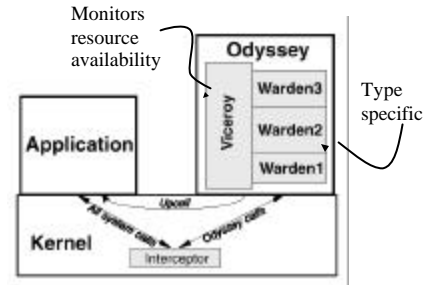## Results (constant message size)
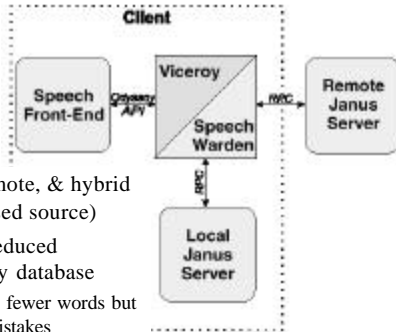

Results (data shipped)

3

Energy Aware Adaptation
(Flinn and Satya - SOSP99)

- Odyssey - system for adaptation
- Fidelity - the degree to which delivered data matches the reference copy at the server
  - type-specific notion
- Question: Can lowering fidelity be used as a mechanism for energy-aware adaptation? Enough savings possible?

Odyssey Architecture

Monitors resource availability

Odyssey

Viceroy

Warden3

Warden2

Warden1

Type specific

Application

Kernel

Interceptor

Odyssey Speech Recognizer

Client

Speech Front-End

Viceroy

Speech Warden

Remote Janus Server

Local Janus Server

- Local, remote, & hybrid (compressed source)
- fidelity: reduced vocabulary database
  - tradeoff: fewer words but fewer mistakes

Energy of Speech Recognition

Reduced fidelity
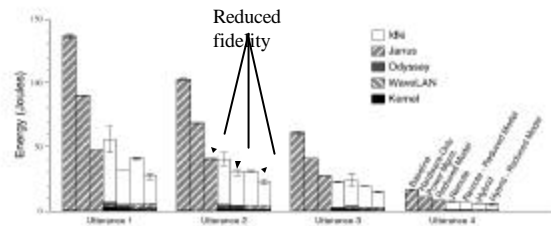
- Baseline - local, full fidelity, no HW power mgt.
- HW-only - display off, disk spun-down, network off
- Remote - most of the energy spent in idle mode on portable

Conclusions

- Significant variation in effectiveness of fidelity reduction across data objects
- Significant variation in effectiveness of fidelity reduction across applications
- Reducing fidelity can enhance effectiveness of HW power management (by lowering utilization of HW)

How?

- Mixed results as to the effectiveness of remote execution for energy savings.
  How can "right" choices be made?
- What aspects could benefit from more energy efficient *implementation* of the support systems themselves?
  - D'agents system used interpretation on the server vs. direction execution on portable.
  - Implementation of "Idle"
- What are the basic building blocks?

Infrastructure Requirements
(Rudenko et al)

- Ability to *do* remote execution, shipping arguments and results
  - Energy efficient listening on portable
- Replication mechanisms and synchronization for data and code.
- Providing consistent execution environment
- Decision strategy for choosing between local and remote (UI or automagically?)