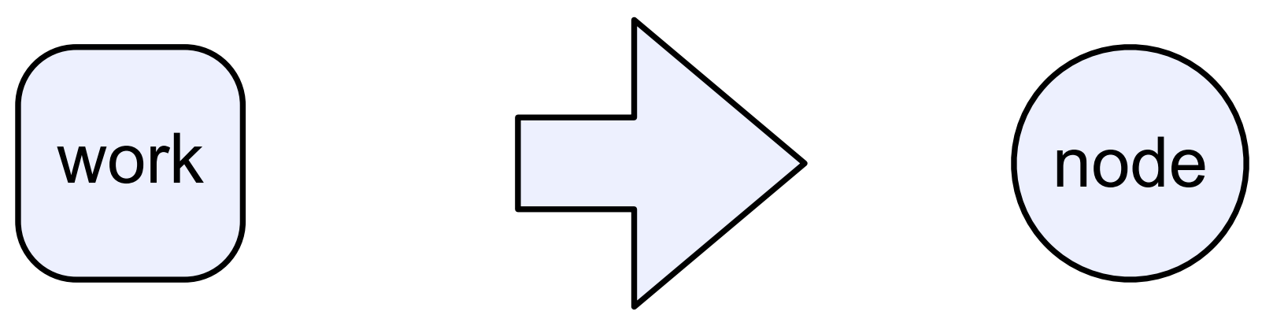
XCPU^3

Workload Distribution & Aggregation

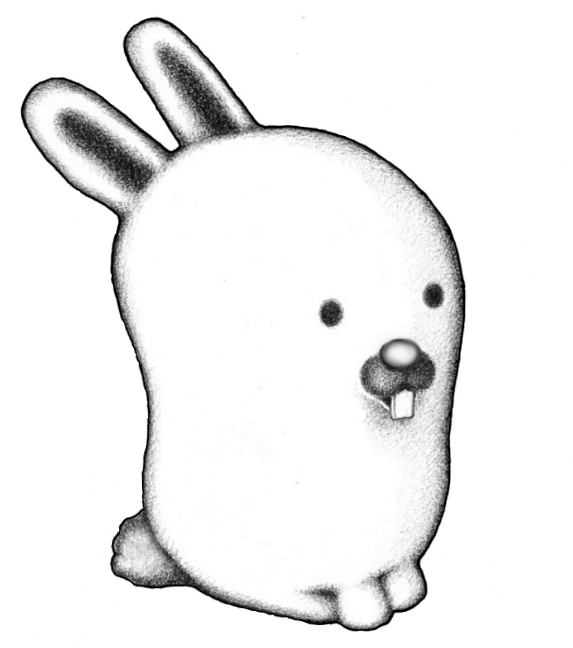
Pravin Shinde & Eric Van Hensbergen

Problem



- Workload distribution hasn't evolved much from when we were batch scheduling tasks to single machines
- Today's Cluster Based Schedulers:
  - Not interactive.
  - Not resilient to failure.
  - Difficult for existing tasks to dynamically grow or shrink resources allocated to it.
  - Difficult to deploy & administer.
  - Based on middleware instead of integrated with underlying operating system.
  - In many cases tightly bound to the underlying runtime or language.
  - Unlikely to function at exascale.

Related Work



System V UNIX

Provided synthetic file system access to process information which was later extended to a hierarchy in Linux procfs.

Plan 9 from Bell Labs

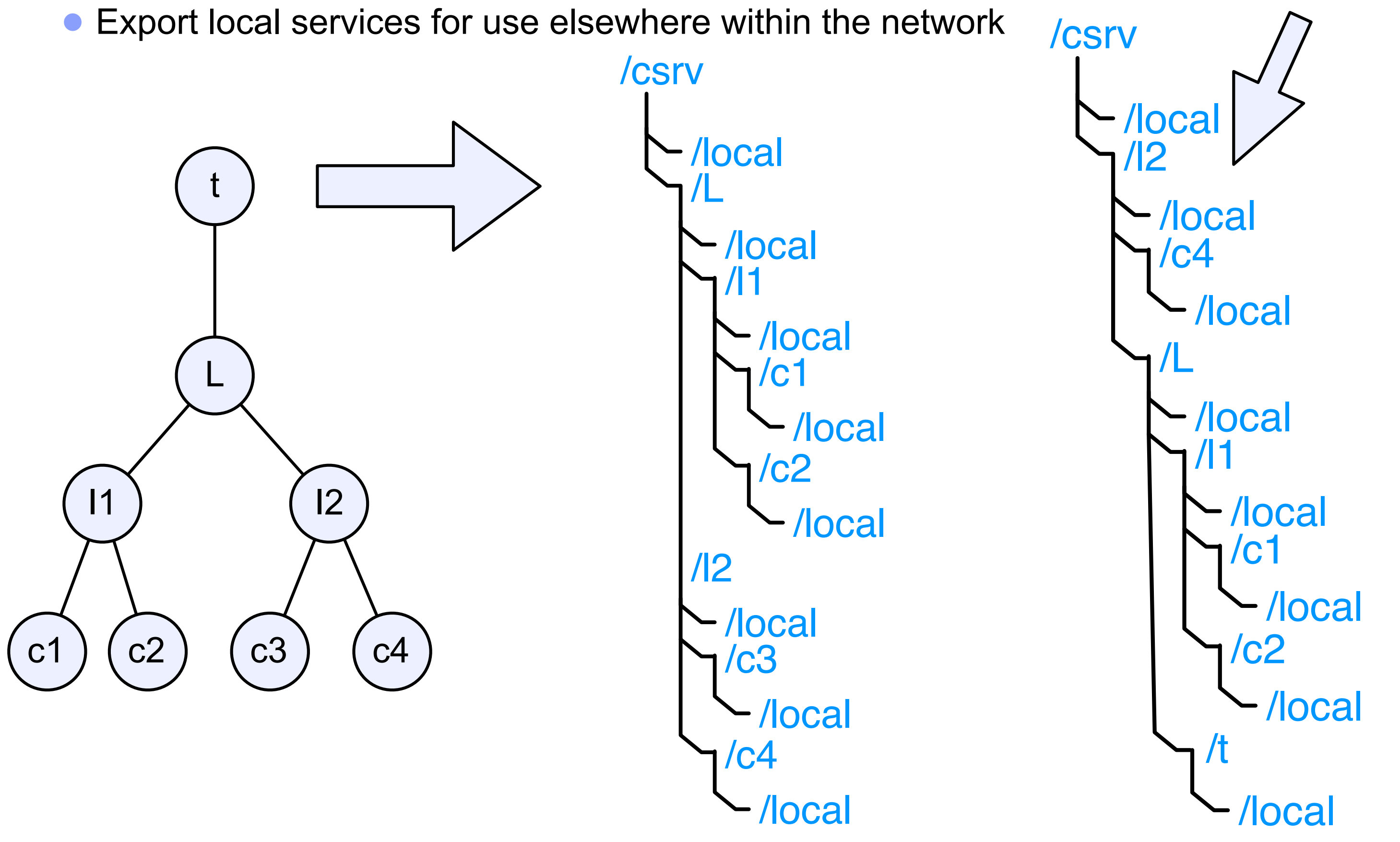
Extended basic procfs concepts by also enabling control and debug interfaces. The nature of the Plan 9 distributed namespace also made these process interfaces available over the network.

XCPU (LANL)

Built an application-layer provided file system for UNIX systems using the Plan 9 model. XCPU extended previous work by allowing process creation to occur via the file system and allowed for execution and coordination of groups of processes on remote systems.

Our Approach

- Establish hierarchical namespace of cluster services
- Automount remote servers based on reference (ie. cd /csrv/criswell)
- Export local services for use elsewhere within the network



Environment Syntax

- key=value
- OBJTYPE=386
- SYSTYPE=Linux
- etc.

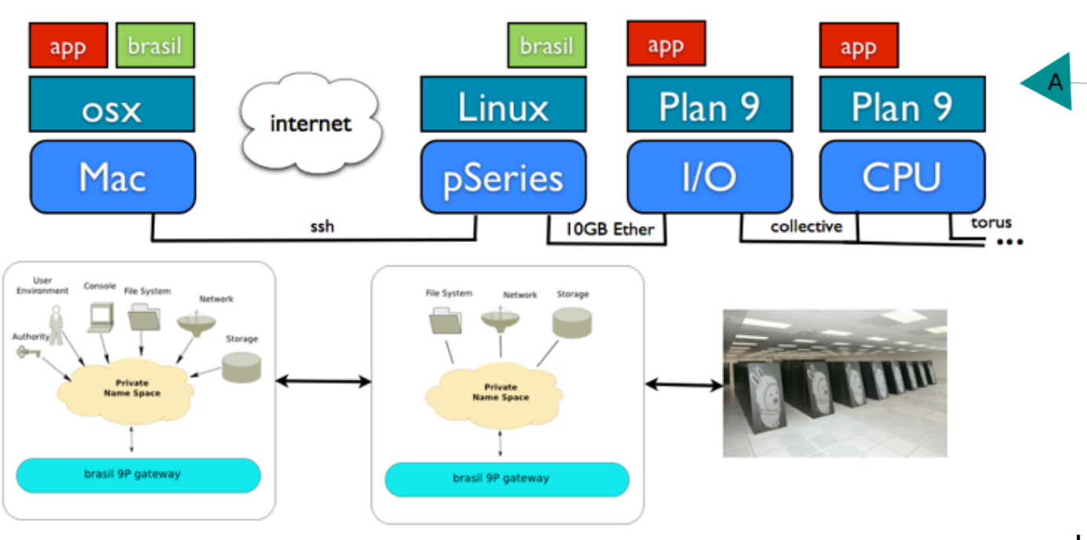
Name Space File Syntax

- `mount` [–`abcC`] *servename* `old` [*spec*]: Mount servename on old.
- `bind` [–`abcC`] new old: Bind *new* on *old*.
- `import` [–`abc`] host `[`remotepath] mountpoint: Import *remotepath* from machine *server* and attach it to *mountpoint*.
- `cd` dir: Change the working directory to *dir*.
- unmount [*new*] *old:* Unmount *new* from *old*, or everything mounted on *old* if *new* is missing.
- clear: Clear the name space with `rfork(RFCNAMEG)`.
- `.` path: Execute the namespace file *path*. Note that *path* must be present in the name space being built.

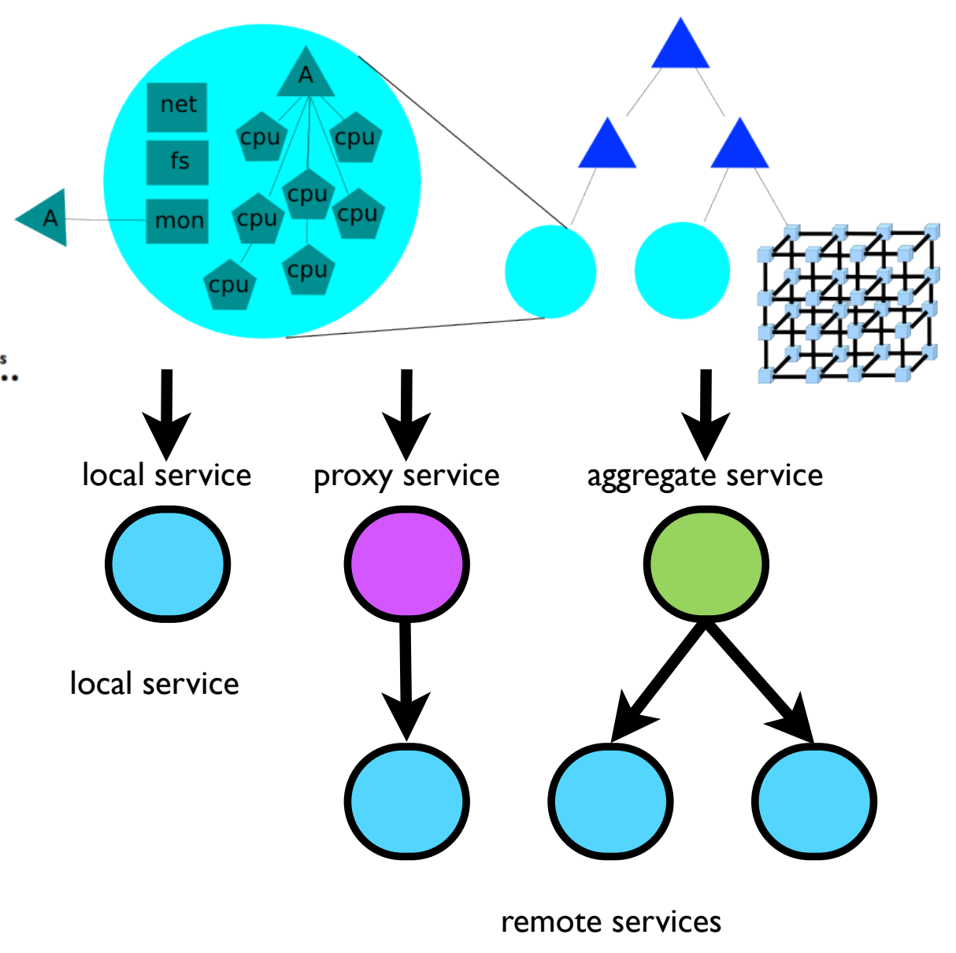
```
/local
  arch        - architecture & platform (ie. Linux i386)
  env         - default environment variables for host
  ns          - default name space for host
  fs          - access to host file system
  net         - access to host network (i.e. Plan 9 devip)
  status      - load average, running jobs, available memory
  clone       - open to establish new session
  /0
  /1          - session subdirectories
  /n
    ctl       - reservation and task control
    env       - environment variables for task
    ns        - name space for task
    args      - task arguments
    wait      - blocks until all threads complete
    status    - current task status (reserved, running, etc.)
    stdin     - aggregate standard input for task
    stdout    - aggregate standard output for task
    stdio     - combined standard I/O for task
    /0        - component thread session subdirectories
    /n
      ctl     - thread control
      env     - environment variables for thread
      ns      - name space for thread
      args    - thread arguments
      wait    - blocks until thread completes
      status  - current thread status (reserved, running, etc.)
      stdin   - standard input for thread
      stdout  - standard output for thread
      stdio   - standard I/O for thread
```

Control File Syntax

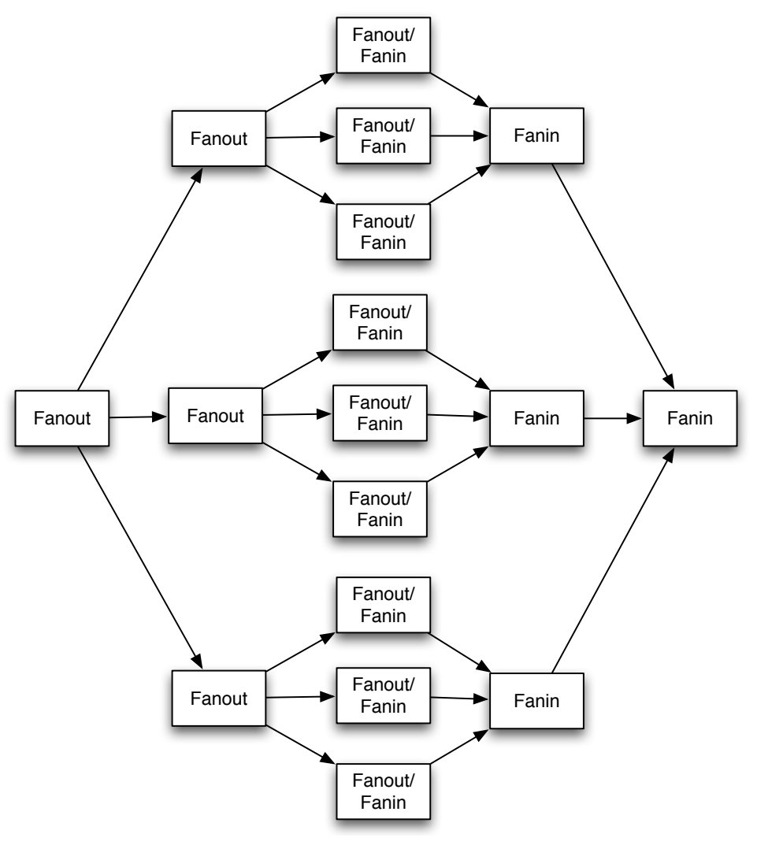
- reserve [n] [os] [arch] - reserve a (number of) resources with os and arch specification
- dir [wdir] - set the working directory for the task
- exec commands args ... - spawn a host process to run the command with arguments as given
- kill - kill the host command immediately
- killonclose - set the device to kill the host command when the ctl file is closed
- nice [n] - set the scheduling priority of the host command
- splice [path] - splice standard output to [path] (on executing host)



Desktop Extension



Aggregation Via Dynamic Namespace and Distributed Service Model



PUSH Pipeline Model

Scaling

Reliability

This project is supported in part by the U.S. Department of Energy under Award Number DE-FG02- 08ER25851

AUSTIN RESEARCH LAB
http://www.research.ibm.com/austin

For More Information: http://www.research.ibm.com/hare